ICINCO 2009

6TH INTERNATIONAL CONFERENCE ON INFORMATICS IN CONTROL, AUTOMATION AND ROBOTICS

Proceedings

Volume 2 - Robotics and Automation

MILAN - ITALY · JULY 2 - 5, 2009

ORGANIZED BY
INSTICC

IN COOPERATION WITH
aaai

CO-SPONSORED BY
IFAC

INSTICC PRESS

ICINCO 2009

Proceedings of the
6th International Conference on
Informatics in Control, Automation and Robotics

Volume 2
Robotics and Automation

Milan, Italy

July 2 - 5, 2009

Organized by
INSTICC – Institute for Systems and Technologies of Information, Control and Communication

Co-Sponsored by
IFAC – International Federation of Automatic Control

In Cooperation with
AAAI – Association for the Advancement of Artificial Intelligence

Edited by Joaquim Filipe, Juan Andrade Cetto and Jean-Louis Ferrier

Printed in Portugal

ISBN: 978-989-674-000-9

Depósito Legal: 294961/09

http://www.icinco.org

secretariat@icinco.org

BRIEF CONTENTS

INVITED SPEAKERS

Daniel S. Yeung

University of Technology

China

Maria Pia Fanti

Polytechnic of Bari

Italy

Janan Zaytoon

University of Reims Champagne Ardennes

France

Alessandro Giua

University of Cagliari

Italy

Peter S. Sapaty

Institute of Mathematical Machines and Systems, National Academy of Sciences

Ukraine

ORGANIZING AND STEERING COMMITTEES

CONFERENCE CHAIR

Joaquim Filipe, Polytechnic Institute of Setúbal / INSTICC, Portugal

PROGRAM CHAIR

Juan Andrade Cetto, Institut de Robòtica i Informàtica Industrial CSIC-UPC, Spain

Jean-Louis Ferrier, University of Angers, France

PROCEEDINGS PRODUCTION

Sérgio Brissos, INSTICC, Portugal

Helder Coelhas, INSTICC, Portugal

Vera Coelho, INSTICC, Portugal

Andreia Costa, INSTICC, Portugal

Bruno Encarnação, INSTICC, Portugal

Bárbara Lima, INSTICC, Portugal

Raquel Martins, INSTICC, Portugal

Carla Mota, INSTICC, Portugal

Vitor Pedrosa, INSTICC, Portugal

José Varela, INSTICC, Portugal

CD-ROM PRODUCTION

Elton Mendes, INSTICC, Portugal

Pedro Varela, INSTICC, Portugal

GRAPHICS PRODUCTION AND WEBDESIGNER

Marina Carvalho, INSTICC, Portugal

SECRETARIAT AND WEBMASTER

Marina Carvalho, INSTICC, Portugal

PROGRAM COMMITTEE

Arvin Agah, The University of Kansas, U.S.A.

Alessandro Chiuso, Universita di Padova, Italy

Hyo-Sung Ahn, Gwangju Institute of Science and Technology (GIST), Korea, Republic of

Frank Allgower, University of Stuttgart, Germany

Francesco Amigoni, Politecnico di Milano, Italy

Plamen Angelov, Lancaster University, U.K.

Peter Arato, Budapest University of Technology and Economics, Hungary

Helder Araújo, University of Coimbra, Portugal

Marco Antonio Arteaga, Universidad Nacional Autonoma de Mexico, Mexico

Vijanth Sagayan Asirvadam, Universiti Technologi PETRONAS, Malaysia

T. Asokan, Indian Institute of Technology Madras, India

Robert Babuska, Delft University of Technology, The Netherlands

Ruth Bars, Budapest University of Technology and Economics, Hungary

Adil Baykasoglu, University of Gaziantep, Turkey

Maren Bennewitz, University of Freiburg, Germany

Karsten Berns, University of Kaiserslautern, Germany

Arijit Bhattacharya, Dublin City University, Ireland

Sergio Bittanti, Politecnico di Milano, Italy

Stjepan Bogdan, University of Zagreb, Faculty of EE&C, Croatia

Jean-louis Boimond, ISTIA - LISA, France

Djamel Bouchaffra, Oakland University, U.S.A.

Bernard Brogliato, INRIA, France

Edmund Burke, University of Nottingham, U.K.

Clifford Burrows, Innovative Manufacturing Research Centre, U.K.

Dídac Busquets, Universitat de Girona, Spain

Giuseppe Carbone, LARM - Laboratorio di Robotica e Meccatronica, Italy

J. L. Martins de Carvalho, Instituto de Sistemas e Robótica - Porto, Portugal

Alessandro Casavola, University of Calabria, Italy

Riccardo Cassinis, University of Brescia, Italy

Chien Chern Cheah, Nanyang Technological University, Singapore

Tongwen Chen, University of Alberta, Canada

Wen-Hua Chen, Loughborough University, U.K.

Graziano Chesi, University of Hong Kong, China

Carlos Coello Coello, Cinvestav-IPN, Mexico

Yechiel Crispin, Embry-riddle Aeronautical University, U.S.A.

Michael A. Demetriou, Worcester Polytechnic Institute, U.S.A.

Guilherme DeSouza, University of Missouri, U.S.A.

Jorge Dias, Institute of Systems and Robotics, Portugal

Gamini Dissanayake, University of Technology, Sydney, Australia

Denis Dochain, Université Catholique de Louvain, Belgium

Tony Dodd, The University of Sheffield, U.K.

Alexandre Dolgui, Ecole des Mines de Saint Etienne, France

Prabu Dorairaj, Wipro Technologies, India

Marco Dorigo, Université Libre de Bruxelles, Belgium

Venky Dubey, Bournemouth University, U.K.

Petr Ekel, Pontifical Catholic University of Minas Gerais, Brazil

Andries Engelbrecht, University of Pretoria, South Africa

Sebastian Engell, Univeristy of Dortmund, Germany

PROGRAM COMMITTEE (CONT.)

Simon G. Fabri, University of Malta, Malta

Sergej Fatikow, University of Oldenburg, Germany

Jean-marc Faure, Ecole Normale Superieure de Cachan, France

Paolo Fiorini, Università degli Studi di Verona, Italy

Juan J. Flores, University of Michoacan, Mexico

Georg Frey, German Researc Center for Artificial Intelligence - DFKI, Germany

Manel Frigola, Technical University of Catalonia (UPC), Spain

John Qiang Gan, University of Essex, U.K.

Nicholas Gans, National Reasearch Council and Air Force Research Laboratory, U.S.A.

Leonardo Garrido, Monterrey Institute of Technology, Mexico

Andrea Garulli, Universita' di Siena, Italy

Lazea Gheorghe, Technical University of Cluj-Napoca, Romania

Paulo Gil, Universidade Nova de Lisboa, Portugal

Alessandro Giua, University of Cagliari, Italy

Luis Gomes, Universidade Nova de Lisboa, Portugal

Dongbing Gu, University of Essex, U.K.

Guoxiang Gu, Louisiana State University, U.S.A.

Jason Gu, Dalhousie University, Canada

Wail Gueaieb, University of Ottawa, Canada

José J. Guerrero, Universidad de Zaragoza, Spain

Thomas Gustafsson, Luleå University of Technology, Sweden

Maki K. Habib, Saga University, Japan

Hani Hagras, University of Essex, U.K.

Wolfgang Halang, Fernuniversitaet, Germany

Riad Hammoud, Delphi Corporation, U.S.A.

Uwe D. Hanebeck, Institute for Anthropomatics, Germany

Robert Harrison, The University of Sheffield, U.K.

Dominik Henrich, University of Bayreuth, Germany

Francisco Herrera, University of Granada, Spain

Victor Hinostroza, University of Ciudad Juarez, Mexico

Wladyslaw Homenda, Warsaw University of Technology, Poland

Guoqiang Hu, Kansas State University, U.S.A.

Marc Van Hulle, K. U. Leuven, Belgium

Fumiya Iida, Robot Locomotion Group, U.S.A.

Atsushi Imiya, IMIT Chiba University, Japan

Hisao Ishibuchi, Osaka Prefecture University, Japan

Thira Jearsiripongkul, Thammasat University, Thailand

Dimitrios Karras, Chalkis Institute of Technology, Greece

Dusko Katic, Mihailo Pupin Institute, Serbia

Graham Kendall, University of Nottingham, U.K.

Tamas Keviczky, Delft University of Technology, The Netherlands

Jonghwa Kim, University of Augsburg, Germany

Won-jong Kim, Texas A&M University, U.S.A.

Waree Kongprawechnon, Thammasat University, Thailand

Israel Koren, University of Massachusetts, U.S.A.

George L. Kovács, Hungarian Academy of Sciences, Hungary

H. K. Lam, King's College London, U.K.

Kemal Leblebicio, Middle East Technical University, Turkey

Graham Leedham, University of New England, Australia

Kauko Leiviskä, University of Oulu, Finland

Kang Li, Queen's University Belfast, U.K.

Tsai-Yen Li, National Chengchi University, Taiwan

PROGRAM COMMITTEE (CONT.)

Yangmin Li, University of Macau, China

Huei-Yung Lin, National Chung Cheng University, Taiwan

Zongli Lin, University of Virginia, U.S.A.

Jing-Sin Liu, Institute of Information Science, Academis Sinica, Taiwan

Jose Tenreiro Machado, Institute of Engineering of Porto, Portugal

Frederic Maire, Queensland University of Technology, Australia

Om Malik, University of Calgary, Canada

Jacek Mandziuk, Warsaw University of Technology, Poland

Hervé Marchand, INRIA, France

Gerard Mckee, The University of Reading, U.K.

Seán McLoone, National University of Ireland (NUI) Maynooth, Ireland

Carlo Menon, Simon Fraser University, Canada

Sanya Mitaim, Thammasat University, Thailand

Pieter Mosterman, The MathWorks, U.S.A.

Rafael Muñoz-salinas, University of Cordoba, Spain

Kenneth Muske, Villanova University, U.S.A.

Andreas Nearchou, University of Patras, Greece

Luciana P. Nedel, Universidade Federal do Rio Grande do Sul (UFRGS), Brazil

Sergiu Nedevschi, Technical University of Cluj-Napoca, Romania

Anton Nijholt, University of Twente,
The Netherlands

Hendrik Nijmeijer, Eindhoven University of Technology, The Netherlands

Juan A. Nolazco-Flores, ITESM, Campus Monterrey, Mexico

Urbano Nunes, University of Coimbra, Portugal

José Valente de Oliveira, Universidade do Algarve, Portugal

Romeo Ortega, LSS/CNRS/Supélec, France

Manuel Ortigueira, Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, Portugal

Selahattin Ozcelik, Texas A&M University-Kingsville, U.S.A.

Christos Panayiotou, University of Cyprus, Cyprus

Stefano Panzieri, Università degli Studi "Roma Tre", Italy

Igor Paromtchik, INRIA, France

D. T. Pham, Cardiff University, U.K.

Marie-Noëlle Pons, CNRS, France

Raul Marin Prades, Jaume I University, Spain

Jerzy Respondek, Silesian University of Technology, Poland

A. Fernando Ribeiro, Universidade do Minho, Portugal

Robert Richardson, University of Leeds, U.K.

Rodney Roberts, Florida State University, U.S.A.

Juha Röning, University of Oulu, Finland

António Ruano, CSI, Portugal

Fariba Sadri, Imperial College London, U.K.

Carlos Sagüés, University of Zaragoza, Spain

Mehmet Sahinkaya, University of Bath, U.K.

Antonio Sala, Universidad Politecnica de Valencia, Spain

Abdel-badeeh Salem, Ain Shams University, Egypt

Mitsuji Sampei, Tokyo Institute of Technology, Japan

Medha Sarkar, Middle Tennessee State University, U.S.A.

Jurek Sasiadek, Carleton University, Canada

Daniel Sbarbaro, Universidad de Concepcion, Chile

Carla Seatzu, University of Cagliari, Italy

João Sequeira, Instituto Superior Técnico / Institute for Systems and Robotics, Portugal

PROGRAM COMMITTEE (CONT.)

Michael Short, University of Leicester, U.K.

Silvio Simani, University of Ferrara, Italy

Dan Simon, Cleveland State University, U.S.A.

Adam Slowik, Koszalin University of Technology, Poland

Michael Small, Hong Kong Polytechnic University, Hong Kong

Burkhard Stadlmann, University of Applied Sciences Wels, Austria

Tarasiewicz Stanislaw, Université Laval, Canada

Karl Stol, University of Auckland, New Zealand

Olaf Stursberg, Technische Universitaet Muenchen, Germany

Chun-Yi Su, Concordia University, Canada

Cornel Sultan, Virginia Tech, U.S.A.

Ryszard Tadeusiewicz, AGH University of Science and Technology, Poland

Choon Yik Tang, University of Oklahoma, U.S.A.

Daniel Thalmann, VR Lab EPFL, Switzerland

N. G. Tsagarakis, Istituto Italiano di Tecnologia, Italy

Antonios Tsourdos, Cranfield University (Cranfield Defence and Security), U.K.

Nikos Tsourveloudis, Technical University of Crete, Greece

Anthony Tzes, University of Patras, Greece

Dariusz Ucinski, University of Zielona Gora, Poland

Nicolas Kemper Valverde, Universidad Nacional Autónoma de México, Mexico

Eloisa Vargiu, University of Cagliari, Italy

Laurent Vercouter, Ecole Nationale Superieure des Mines de Saint-Etienne, France

Bernardo Wagner, University of Hannover, Germany

Axel Walthelm, sepp.med GmbH, Germany

Dianhui Wang, La Trobe University, Australia

Qing-Guo Wang, National University of Singapore, Singapore

Zidong Wang, Brunel University, U.K.

James Whidborne, Cranfield University, U.K.

Dirk Wollherr, Technische Universität München, Germany

Marek Zaremba, Université du Québec (UQO), Canada

Janan Zaytoon, University of Reims Champagne Ardennes, France

Qin Zhang, University of Illinois at Urbana-Champaign, U.S.A.

AUXILIARY REVIEWERS

Andrea Baccara, Università degli Studi di Cagliari, Italy

Rui Cortesao, University of Coimbra, Portugal

Matteo de Felice, ENEA, Italy

Andrea Gasparri, University of Roma Tre, Italy

Zhi Han, The MathWorks, U.S.A.

Vitor Jorge, Universidade Federal do Rio Grande do Sul, Brazil

Enrico di Lello, Universita' degli Studi "Roma Tre", Italy

Andrea Monastero, University of Verona, Italy

Federico di Palma, University of Verona, Italy

Maura Pasquotti, Universita' degli Studi di Verona, Italy

Katalin Popovici, The MathWorks, France

Monica Reggiani, University of Padua, Italy

Maurizio di Rocco, Università degli Studi Roma Tre, Italy

P. Lopes dos Santos Santos, Instituto de Sistemas e Robótica - Porto, Portugal

Thomas Tometzki, Process Dynamics and Operations Group, Germany

Ali Emre Turgut, IRIDIA, Belgium

Maja Varga, Faculty of Electrical Engineering and Computing, Croatia

Kai Wurm, Institute of Computer Science, Germany

SELECTED PAPERS BOOK

A number of selected papers presented at ICINCO 2009 will be published by Springer-Verlag in a LNEE Series book. This selection will be done by the Conference chair and Program Co-chairs, among the papers actually presented at the conference, based on a rigorous review by the ICINCO 2009 Program Committee members.

FOREWORD

This book contains the proceedings of the 6th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2009) which was organized by the Institute for Systems and Technologies of Information, Control and Communication (INSTICC) and held in Milan. ICINCO 2009 was co-sponsored by the International Federation for Automatic Control (IFAC) and held in cooperation with the Association for the Advancement of Artificial Intelligence (AAAI).

The ICINCO Conference Series has now consolidated as a major forum to debate technical and scientific advances presented by researchers and developers both from academia and industry, working in areas related to Control, Automation and Robotics that benefit from Information Technology.

In the Conference Program we have included oral presentations (full papers and short papers) and posters, organized in three simultaneous tracks: “Intelligent Control Systems and Optimization”, “Robotics and Automation” and “Systems Modeling, Signal Processing and Control”. We have included in the program five plenary keynote lectures, given by internationally recognized researchers, namely - Daniel S. Yeung (University of Technology, China), Maria P. Fanti (Polytechnic of Bari, Italy), Janan Zaytoon (University of Reims Champagne Ardennes, France), Alessandro Giua (Universita’ di Cagliari, Italy) and Peter S. Sapaty (Institute of Mathematical Machines and Systems, National Academy of Sciences Ukraine).

The meeting is complemented with three satellite workshops, focusing on specialized aspects of Informatics in Control, Automation and Robotics; namely, the International Workshop on Artificial Neural Networks and Intelligent Information Processing (ANNIIP), the International Workshop on Intelligent Vehicle Controls & Intelligent Transportation Systems (IVC & ITS) and the International Workshop on Networked Embedded and Control System Technologies: European and Russian R&D Cooperation (NESTER).

ICINCO received 365 paper submissions, not including those of workshops, from more than 55 countries, in all continents. To evaluate each submission, a double blind paper review was performed by the Program Committee. Finally, only 178 papers are published in these proceedings and presented at the conference. Of these, 129 papers were selected for oral presentation (34 full papers and 95 short papers) and 49 papers were selected for poster presentation. The full paper acceptance ratio was 9%, and the oral acceptance ratio (including full papers and short papers) was 35%. As in previous editions of the Conference, based on the reviewer’s evaluations and the presentations, a short list of authors will be invited to submit extended versions of their papers for a book that will be published by Springer with the best papers of ICINCO 2009.

Conferences are also meeting places where collaboration projects can emerge from social contacts amongst the participants. Therefore, in order to promote the development of research and professional networks the Conference includes in its social program a Conference and Workshops Social Event & Banquet in the evening of July 4 (Saturday).

We would like to express our thanks to all participants. First of all to the authors, whose quality work is the essence of this Conference. Next, to all the members of the Program Committee and auxiliary reviewers, who helped us with their expertise and valuable time. We would also like to deeply thank the invited speakers for their excellent contribution in sharing their knowledge and vision. Finally, a word of appreciation for the hard work of the secretariat; organizing a conference of this level is a task that can only be achieved by the collaborative effort of a dedicated and highly capable team.

Commitment to high quality standards is a major aspect of ICINCO that we will strive to maintain and reinforce next year, including the quality of the keynote lectures, of the workshops, of the papers, of the organization and other aspects of the conference. We look forward to seeing more results of R&D work in Informatics, Control, Automation and Robotics at ICINCO 2010.

Joaquim Filipe
Polytechnic Institute of Setúbal / INSTICC, Portugal

Juan Andrade Cetto
Institut de Robòtica i Informàtica Industrial CSIC-UPC, Spain

Jean-Louis Ferrier
University of Angers, France

CONTENTS

INVITED SPEAKERS

KEYNOTE SPEAKERS

ROBOTICS AND AUTOMATION

FULL PAPERS

SHORT PAPERS

POSTERS

INVITED SPEAKERS

KEYNOTE SPEAKERS

SENSITIVITY BASED GENERALIZATION ERROR FOR SUPERVISED LEARNING PROBLEM WITH APPLICATIONS IN MODEL SELECTION AND FEATURE SELECTION

Daniel S. Yeung
University of Technology
China

Abstract: Generalization error model provides a theoretical support for a classifier's performance in terms of prediction accuracy. However, existing models give very loose error bounds. This explains why classification systems generally rely on experimental validation for their claims on prediction accuracy. In this talk we will revisit this problem and explore the idea of developing a new generalization error model based on the assumption that only prediction accuracy on unseen points in a neighbourhood of a training point will be considered, since it will be unreasonable to require a classifier to accurately predict unseen points "far away" from training samples. The new error model makes use of the concept of sensitivity measure for an ensemble of multiplayer feedforward neural networks (Multilayer Perceptrons or Radial Basis Function Neural Networks). Two important applications will be demonstrated, model selection and feature reduction for RBFNN classifiers. A number of experimental results using datasets such as the UCI, the 99 KDD Cup, and text categorization, will be presented.

BRIEF BIOGRAPHY

Daniel S. Yeung (Ph.D., M.Sc., M.B.A., M.S., M.A., B.A.) is the President of the IEEE Systems, Man and Cybernetics (SMC) Society, a Fellow of the IEEE and an IEEE Distinguished Lecturer. He received the Ph.D. degree in applied mathematics from Case Western Reserve University. In the past, he has worked as an Assistant Professor of Mathematics and Computer Science at Rochester Institute of Technology, as a Research Scientist in the General Electric Corporate Research Center, and as a System Integration Engineer at TRW, all in the United States. He was the chairman of the department of Computing, The Hong Kong Polytechnic University, Hong Kong, and a Chair Professor from 1999 to 2006. Currently he is a Chair Professor in the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China. His current research interests include neural-network sensitivity analysis, data mining, Chinese computing, and fuzzy systems. He was the Chairman of IEEE Hong Kong Computer Chapter (91and 92), an associate editor for both IEEE Transactions on Neural Networks and IEEE Transactions on SMC (Part B), and for the International Journal on Wavelet and Multiresolution Processing. He has served as a member of the Board of Governors, Vice President for Technical Activities, and Vice President for Long Range Planning and Finance for the IEEE SMC Society. He co-founded and served as a General Co-Chair since 2002 for the International Conference on Machine Learning and Cybernetics held annually in China. He also serves as a General Co-Chair (Technical Program) of the 2006 International Conference on Pattern Recognition. He is also the founding Chairman of the IEEE SMC Hong Kong Chapter.

His past teaching and academic administrative positions include a Chair Professor and Head at Department of Computing, The Hong Kong Polytechnic University, the Head of the Management Information Unit at the Hong Kong Polytechnic University, Associate Head/Principal Lecturer at the Department of Computer Science, City Polytechnic of Hong Kong, a tenured Assistant Professor at the School of Computer Science and Technology and an Assistant Professor at the Department of Mathematics, both at Rochester Institute of Technology, Rochester, New York.

He also held industrial and business positions as a Technical Specialist/Application Software Group Leader at the Computer Consoles, Inc., Rochester, New York, an Information Resource Sub-manager/Staff Engineer at the Military and Avionics Division, TRW Inc., San Diego, California, and an Information Scientist of the Information System Operation Lab, General Electric Corporate Research and Development Centre, Schenectady, New York.

IMPACT OF THE ICT ON THE MANAGEMENT AND PERFORMANCE OF INTELLIGENT TRANSPORTATION SYSTEMS

Maria Pia Fanti
Department of Electrical and Electronic Engineering, Polytechnic of Bari, Via Re David 200, Bari, Italy
fanti@deemail.poliba.it

Keywords: Intelligent Transportation Systems, Urban Traffic Control, Intermodal Transportation Networks, Information and Communication Technologies.

Abstract: Intelligent Transportation Systems (ITS) modelling, planning, and control are research streams that, in the last years, have received a significant attention by the researcher and practitioner communities due not only to their economic impact, but also to the complexity of decisional, organizational, and management problems. Indeed, the increasing complexity of these systems and the availability of the modern ICT (Information and Communication Technologies) for the interaction among the different decision makers and for the acquisition of information by the decision makers, require both the development of suitable models and the solution of new decision problems. This presentation is aimed at showing the new attractive researches and projects in the field of ITS operational control and management in Europe. In particular, it points out the key solution of using effectively and efficiently the latest developments of ICT for ITS operational management.

1 INTRODUCTION

The term Intelligent Transportation Systems (ITS) is used to refer to technologies, infrastructure, and services, as well as the planning, operation, and control methods to be used for the transportation of person and freight. In particular, Information and Communication Technologies (ICT) are considered to be the key tools to improve efficiency and safety in transportation systems. Indeed, the advent of ICT has a tremendous impact on the planning and operations of freight transportation and on traffic management systems. ITS technologies increase the flow of available data, improve the timeliness and quality of information and offer the possibility to control and coordinate operations and traffic in real-time. Significant research efforts are required to adequately model the various planning and management problems under ITS and real-time information, and to develop efficient solution methods.

In recent years, the European Union has sponsored several projects targeting advancements of different transportation systems. On the other hand, ITS topics are considered relevant and attractive research areas.

In Section 2 the paper recalls the most important European Projects in the fields of ITS and intelligent freight transportation. Moreover, Sections 3 and 4 present the research advances in two crucial sectors of ITS: the management of Urban Traffic Networks (UTN) and of Intermodal Transportation Networks (ITN), respectively.

2 EUROPEAN PROJECTS IN THE FIELD OF ITS

A basic project on ITS is CESAR I & II (Co-operative European System for Advanced Information Redistribution) that proposes an Internet communication platform that aims to integrate services and data for unaccompanied traffic and the rolling motorway traffic management. Moreover, in the field of railway system management, CroBIT (Cross Border Information Technology) is a new system that provides the railways with a tool to track consignments and integrates freight railways along a transport corridor providing total shipment visibility. A maritime navigation information structure in

European waters is established by MarNIS (Maritime Navigation Information Systems) that is an integrated project aiming to develop tools that can be used to exchange maritime navigation information and to improve safety, security and efficiency of maritime traffic.

In addition, several projects focus specifically on efficient freight transportation. For instance, Freightwise aims to establish a framework for efficient co-modal freight transport on the Norwegian ARKTRANS system. One of the main objectives in Freightwise is establishing a framework for efficient co-modal freight transport and simplifying the interaction among stakeholders during planning, execution and completion of transport operations. Moreover, the project e-Freight is a continuation of Freightwise to promote efficient and simplified solutions in support of cooperation, interoperability and consistency in the European Transport System. E-Freight is to support the Freight Logistics Action Plan, which focuses on quality and efficiency for the movement of goods, as well as on ensuring that freight-related information travels easily among modes. Furthermore, in the Seventh Framework Program (FP7-ICT Objective 6.1), the SMARTFREIGHT project wants to make urban freight transport more efficient, environmentally friendly and safe by answering to challenges related to traffic management and the relative coordination. Indeed, freight distribution management in city centres is usually operated by several commercial companies and there is no coordination of these activities in a way that would benefit the city. The main aim of SMARTFREIGHT is therefore to specify, implement and evaluate ICT solutions that integrate urban traffic management systems with the management of freight and logistics in urban areas. Finally, EURIDICE (European Inter-Disciplinary Research on Intelligent Cargo for Efficient, Safe and Environment-friendly Logistics) is a project sponsored by the European Commission under the 7th Framework Program seeking to develop an advanced European logistics system around the concept of ‘intelligent cargo’. The goal is networking cargo objects like packages, vehicles and containers to provide information services whenever required along the transport chain. The project aims to build an information service platform centred on the individual cargo item and its interaction with the surrounding environment and the user.

3 URBAN TRAFFIC MANAGEMENT

Traffic congestion of urban roads undermines mobility in major cities. Traditionally, the congestion problem on surface streets was dealt by adding more lanes and new links to the existing Urban Traffic Networks (UTN). Since such a solution can no longer be considered for limited availability of space in urban centres, greater emphasis is nowadays placed on traffic management through the implementation and operation of ICT. In particular, traffic signal control on surface street networks plays a central role in traffic management. Despite the large research efforts on the topic, the problem of urban intersection congestion remains an open issue (Lo, 2001, Papageorgiou, 1999). Most of the currently implemented traffic control systems may be grouped into two principal classes (Papageorgiou et al., 2003, Patel and Ranganathan, 2001): i) fixed time strategies, that are derived off-line by use of optimization codes based on historical traffic data; ii) vehicle actuated strategies, that perform an on-line optimization and synchronization of the signal timing plans and make use of real time measurements. While the fixed time strategies do not use information on the actual traffic situation, the second actuated control class can be viewed as a traffic responsive network signal policy employing signal timing plans that respond automatically to traffic conditions. In a real time control strategy, detectors located on the intersection approaches monitor traffic conditions and feed information on the actual system state to the real time controller, which selects the duration of the phases in the signal timing plan in order to optimize an objective function. Although the corresponding optimal control problem may readily be formulated, its real time solution and realization in a control loop has to face several difficulties such as the size and the combinatorial nature of the optimization problem, the measurements of traffic conditions and the presence of unpredictable disturbances. The first and most notable of vehicle actuated techniques is the British SCOOT (Hunt et al., 1982), that decides an incremental change of splits, offsets and cycle times based on real time measurements. However, although SCOOT exhibits a centralized hardware architecture, the strategy is functionally decentralized with regard to splits setting. A formulation of the traffic signal network optimization strategy is presented in (Lo, 2001) and (Wey, 2000). However, the resulting procedures lead to complex mixed integer linear programming problems that are computationally intensive and the formulation for real networks requires heuristics for solutions. Furthermore, Diakaki *et al.* (2002) propose a traffic responsive urban control strategy based on a feedback approach involving the application of a systematic and powerful control design method. Despite the simplicity and the efficiency of the proposed control strategy, such a modelling approach can not directly consider the effects of offset for consecutive junctions and the time-variance of the turning rates and the saturation flows.

An improvement on urban traffic actuated control strategies is provided in (Dotoli et al., 2006) where the green splits for a fixed cycle time are determined in real time, in order to minimize the number of vehicles in queue in the considered signalized area. The paper gives a contribution in facing the *apparently insurmountable difficulties* (Papageorgiou et al., 2003) in the real time solution and realization of the control loop governing an urban intersection by traffic lights. To this aim, the paper pursues simplicity in the modelling and in the optimization procedure by presenting a macroscopic model to describe the urban traffic network. Describing the system by a discrete time model with the sampling time equal to the cycle, the timing plan is obtained on the basis of the real traffic knowledge and the traffic measurements in a prefixed set of cycles. The traffic urban control strategy is performed by solving a mathematical programming problem that minimizes the number of vehicles in the considered urban area. The minimization of the objective function is subject to linear constraints derived from the intersection topology, the fixed cycle duration and the minimum and maximum duration of the phases commonly adopted in practice. The optimization problem is solved by a standard optimization software on a personal computer, so that practical applications are possible in a real time control framework.

4 INTERMODAL TRANSPORTATION NETWORKS

Intermodal Transportation Networks (ITN) are logistics systems integrating different transportation services, designed to move goods from origin to destination in a timely manner and using multiple modes of transportation (rail, ocean vessel, truck etc.). In the related literature several papers analyze ITN operations and planning issues as container fleet management, container terminal operations and scheduling. With the development of the new ICT tools, these operative and planning issues can be dealt with in a different way. In fact, these new technologies can effectively impact on the planning and operation of ITS. In particular, ICT solutions can increase the data flow and the information quality while allowing real-time data exchange in transportation systems (Crainic and Kim, 2007, Ramstedt and Woxenius, 2006). As mentioned in (Giannopoulos, 2004), numerous new applications of ICT to the transportation field are in various stages of development, but in the information transfer area the new systems seem to be too unimodal. In the application of ICT solutions to multimodal chains, an important and largely unexplored research field is the assessment of the impact of new technologies before their implementation, by a cost-benefit analysis (Zografos and Regan, 2004, Crainic and Kim, 2007). This research field offers numerous research opportunities: for instance, a not well explored case is that of coordinating independent stakeholders in the presence of uncertainties and lack of information on the stakeholders operations and their propagation within the intermodal chain.

An efficient ITN needs to synchronize the logistics operations. Therefore, information exchange among stakeholders is essential and ICT solutions are key tools to achieve efficiency. Nevertheless, the increasing complexity of these systems and the availability of the modern ICT for the interaction among the different decision makers and for the acquisition of information by the decision makers, require both the development of suitable models and the solution of new decision problems. Moreover, ITN and their decision making process are complex systems characterized by a high degree of interaction, concurrency and synchronization. Hence, ITS can be modeled as Discrete Event Systems (DES), whose dynamics depends on the interaction of discrete events, such as demands, departures and arrivals of means of transport at terminals and acquisitions and releases of resources by vehicles. DES models are widely used to describe decision making and operational processes. In the domain of ITN, the potentialities of these models are not fully explored and exploited. In particular at the operational level, we recall the models in the Petri net (Peterson, 1981) frameworks (Danielis et al., 2009, Di Febbraro et al., 2006, Fischer et al., 2000) and the simulation models (Boschian et al., 2009, Parola and Sciomachen, 2005).

In this presentation we mention two papers (Boschian et al., 2009) and (Danielis et al., 2009) that point out the role and the impact of the ICT applications in the field of the ITN management and control. In particular, paper (Danielis et al., 2009) focuses on the ICT solutions that allow sharing information among stakeholders on the basis of user friendly technologies. To this aim the authors single out some performance indices to evaluate activities, resources (utilization) and output (throughput, lead time) by integrating information flows allowed by the use of ICT tools. A case study is analyzed considering an ITN constituted by a port and a truck terminal of an Italian town including the road-ship transshipment process. The system is modeled and simulated in a timed Petri net framework considering different dynamic conditions characterized by a diverse level of information shared between terminals and operators. The simulation results show that ICT have a huge potential for efficient real time management and operation of ITN, as well as an effective impact on the infrastructures, reducing both the utilization of the system resources as well as the cost performance indices.

An application of the ICT tools to the real-time transport monitoring in order to trace and safely handle moving goods is presented in (Boschian et al., 2009). In particular, the authors analyze and simulate a real case study involving an ITN system and the transport and the customs clearance of goods that arrive to the port and the intermodal terminal. The case study is analyzed in the frame of the EURIDICE Integrated Project. The structure and the dynamics of the ITN model is described by the Unified Modeling Language formalism (Miles and Hamilton, 2006) and is implemented by a discrete-event simulation in Arena environment. The task is to provide services for the efficient utilization of infrastructures, both singularly and across territorial networks (e.g., port terminal synchronization with rail and road connections) and to contain the impact of logistic infrastructures on the local communities, reducing congestion and pollution caused by the associated freight movements. The discrete event simulation study shows that the application of the ICT tools allows us to locate goods and the related up-to-date information and to extend it with useful information-based services. Summing up, the simulation results show that integrating ICT into the system leads to a more efficient system management and drastically reduces the system lead times.

5 CONCLUSIONS

The paper presents the new attractive researches and projects in the field of ITS operational control and management. In particular, the key solutions of using effectively and efficiently the latest developments of ICT for ITS operational management are pointed out. The presentation focuses on the most important European Projects in ITS and on two crucial fields of the ITS management and control: the management of Urban Traffic networks and of Intermodal Transportation Networks. In the two cases are emphasized the new results and the challenges of future researches.

REFERENCES

Boschian, V., Fanti, M.P., Iacobellis, G., Ukovich, W., 2009. Using Information and Communication Technologies in Intermodal Freight Transportation: a Case Study. Submitted for publication.

Crainic, T.G., Kim, K.H., 2007 Intermodal transportation, In: C. Barnhart and G. Laporte, Editors, *Transportation, Handbooks in Operations Research and Management Science*, vol. 14, North-Holland, Amsterdam, pp. 467–537.

Danielis, R., Dotoli, M., Fanti, M.P., Mangini, A.M., Pesenti R., Stecco G., Ukovich W., 2009, “Integrating ICT into Logistics Intermodal Systems: A Petri Net Model of the Trieste Port”, The European Control Conference 2009, ECC’09, August 23-26, Budapest, Hungary.

Diakaki, C., Papageorgiou, M., Aboudolas, K. 2002. A multivariable regulator approach to traffic-responsive network-wide signal control. *Control Engineering Practice*, *10*(2), 183-195.

Di Febbraro, A., Giglio, D., Sacco, N., 2002. On applying Petri nets to determine optimal offsets for coordinated traffic light timings. *Proc. 5th IEEE Int. Conf. on Intelligent Transportation Systems* (pp. 773-778), Singapore.

Di Febbraro, A., Giglio, D., Sacco, N., 2004. Urban traffic control structure based on hybrid Petri nets. *IEEE Trans. On Intelligent Transportation Systems 5,* (4), 224-237.

Di Febbraro, A., Porta, G., N. Sacco, N., 2006. A Petri net modelling approach of intermodal terminals based on Metrocargo system, *Proc. Intelligent Transportation Systems Conf.*, pp. 1442–1447.

Dotoli, M., Fanti, M.P., Meloni, C., 2006. A Signal Timing Plan Formulation for Urban Traffic Control. *Control Engineering Practice*, vol. 14, no.11, 2006, pp. 1297-1311.

Fischer, M., Kemper, P., 2000. Modeling and analysis of a freight terminal with stochastic Petri nets. In *Proc. 9th IFAC Symposium Control in Transportation Systems*.

Giannopoulos, G.A., 2004. The application of information and communication technologies in transport. In *European Journal Of Operational Research*, vol. 152, pp. 302-320.

Hunt, P.B., Robertson, D.L., Beterton, R.D., & Royle, M.C., 1982. The SCOOT on-line traffic signal optimization technique. *Traffic Engineering and Control, 23,* 190-199.

Lo, H. K., 2001. A cell-based traffic control formulation: strategies and benefits of dynamic timing plans. *Transportation Science*, *35*(2), 148-164.

Miles, R., Hamilton, K., 2006. *Learning UML 2.0.* O'Reilly Media, Sabastopol CA USA.

Papageorgiou, M., 1999. Automatic control methods in traffic and transportation. In *Operations research and decision aid methodologies in traffic and transportation management*, P. Toint, M. Labbe, K. Tanczos, & G. Laporte (Eds.), Springer-Verlag, 46-83.

Papageorgiou, M., Diakaki, C., Dinopoulou, V., Kotsialos, A., Wang, Y., 2003. Review of road traffic control strategies. *Proceedings of the IEEE*, *91*(12), 2043-2067.

Parola F., Sciomachen, A., 2005. Intermodal container flows in a port system metwor: Analysis of possible growths via simulation models. In *International Journal of Production Economics*, vol. 97, pp. 75-88.

Patel, M., Ranganathan, N., 2001. IDUTC: an intelligent decision-making system for urban traffic control applications. In *IEEE Trans. on Vehicular Technology*, *50*(3), 816-829.

Peterson, J.L., 1981. *Petri Net Theory and the Modeling of Systems*. Prentice Hall, Englewood Cliffs, NJ, USA.

Ramstedt, L., Woxenius, J., 2006. Modelling approaches to operational decision-making in freight transport chains, *Proc. 18th NOFOMA Conf.*.

Wey W.-M., 2000. Model formulation and solution algorithm of traffic signal control in an urban network. In *Computers, Environment and Urban Systems*, *24*(4), 355-377.

Zografos, K.G., Regan, A., 2004. Current Challenges for Intermodal Freight Transport and Logistics in Europe and the US. In *Journal of the Transportation Research Board*, vol. 1873, pp. 70-78.

BRIEF BIOGRAPHY

Maria Pia Fanti is associate professor in Systems and Control Engineering and is with the Department of Electrical and Electronic Engineering of the Polytechnic of Bari (Italy). Maria Pia Fanti received the Laurea degree in Electronic Engineering from the University of Pisa (Italy), in 1983 and obtained an IBM thesis award. She was a visiting researcher at the Rensselaer Polytechnic Institute of Troy, New York, in 1999. Her research interests include discrete event systems, Petri nets, modeling and control of automated manufacturing systems, modeling and management of logistics system, supply chains and health care systems.

Prof. Fanti is Associate Editor of the following journals: IEEE Trans. on Systems, Man, and Cybernetics. Part A, IEEE Trans. on Automation Science and Engineering, The Mediterranean J. of Measurement and Control, Int. J. of Automation and Control, and Enterprise Information Systems. She is Co-Chair of the Technical committee on Discrete Event Systems for the IEEE SMC Society, Chair of the Italy Section SMC Chapter, and member of the IFAC Technical Committee on Discrete Event and Hybrid Systems. She is authors of 120+ papers. She has served in 20+ conference international program committees, she is IPC chair of 2nd IFAC Workshop on Dependable Control of Discrete Systems, Bari, Italy, 2009 and of the IEEE Workshop on Health Care Management, Venice, Italy, 2010.

RECENT ADVANCES IN VERIFICATION AND ANALYSIS OF HYBRID SYSTEMS

Janan Zaytoon
University of Reims Champagne Ardennes
France

Abstract: Formal verification of properties is a very important area of analysis of hybrid systems. It is, indeed, essential to use methods and tools to guarantee that the global behaviour of a system is correct and consistent with the specifications. This is especially true for safety properties that insure that the system is not dangerous for itself or its environment.

Classically, verification of Safety properties may be performed with reachability computation in the hybrid state space. Basic ideas have not really evolved since the first works, however new techniques have been proposed and algorithms have been improved.

The aim of this talk is to present the problem of verification and reachability computation for hybrid systems and to propose a classification of recent improvements. To overcome the difficulties in verification and reachability analysis it is necessary to make choices regarding general principles, algorithms and mathematical representation of regions of the continuous state space. These choices depend on each other and must be consistent. However all approaches are based on common considerations that will be used to structure the talk.

BRIEF BIOGRAPHY

Born in 1962, Janan Zaytoon (BSc Eng./1983, MSc Eng./1986, DEA/1988, PhD/1993, Habilitation/1997) is Professor and Head of the CReSTIC Research Centre (involving 150 researchers) at Reims University. He was a member of the Administration Council of the same University (2003-2006). He is the Chair of the French national research network/group “GDR MACS of CNRS”, which involves all the researchers in the field of Automatic Control Syetems in France (about 2000 researchers and PhD students).

His involvement in IFAC includes his service as member of the IFAC Council from 2008 to 2011, head of the French National Member Organizer since 1999, Chair of Technical Committee on Discrete Event and Hybrid Systems from 2005 to 2008, Vice-Chair of this Technical Committee from 2002 to 2005 and 2008 to 2011, member of the Publication Committee of IFAC from 2008 to 2011, Editor of the IFAC Journal “Control Engineering Practice” and the Affiliated IFAC Journal “Nonlinear Analysis: Hybrid Systems”.

Professor Janan Zaytoon is the author/co-author of 70 journal papers, 3 books, 12 book chapters, 120 conference papers, and 8 patents. His main research interests are in the fields of Discrete Event Systems, Hybrid Dynamic Systems, Intelligent Control and Biomedical Engineering. He is an associate Editor of “IET Control Theory and Applications” and “Discrete Event Dynamic Systems”, IPC and/or NOC Chair/Co-Chair of 15 Conferences, Editor/Co-Editor of 10 Conference Proceedings, Keynote speaker for 6 conferences, supervisor of 20 PhD students, Guest Editor/Co-editor for 18 special issues of 6 international and 2 national journals, leader of 8 industrial contracts, and was Chair of the WODES (International Workshop on Discrete Event Systems) steering Committee.

DISCRETE EVENT DIAGNOSIS USING PETRI NETS

Maria Paola Cabasino, Alessandro Giua and Carla Seatzu
Department of Electrical and Electronic Engineering, University of Cagliari, Piazza D'Armi, 09123 Cagliari, Italy
{*cabasino, giua, seatzu*}*@diee.unica.it*

Keywords: Petri nets, Diagnosis, Discrete event systems.

Abstract: This paper serves as a support for the plenary address given by the second author during the conference. In this paper we present an approach to on-line diagnosis of discrete event systems based on labeled Petri nets, that are a particular class of Petri nets where some events are undistinguishable, i.e., events that produce an output signal that is observable, but that is common to other events. Our approach is based on the notion of basis markings and justifications and it can be applied both to bounded and unbounded Petri nets whose unobservable subnet is acyclic. Moreover it is shown that, in the case of bounded Petri nets, the most burdensome part of the procedure may be moved off-line, computing a particular graph that we call *Basis Reachability Graph*.

Finally we present a diagnosis MATLAB toolbox with some examples of application.

1 INTRODUCTION

Failure detection and isolation in industrial systems is a subject that has received a lot of attention in the past few decades. A failure is defined to be any deviation of a system from its normal or intended behavior. Diagnosis is the process of detecting an abnormality in the system behavior and isolating the cause or the source of this abnormality.

Failures are inevitable in today's complex industrial environment and they could arise from several sources such as design errors, equipment malfunctions, operator mistakes, and so on. As technology advances, as we continue to build systems of increasing size and functionality, and as we continue to place increasing demands on the performance of these systems, then so do we increase the complexity of these systems. Consequently (and unfortunately), we enhance the potential for systems to fail, and no matter how safe our designs are, how improved our quality control techniques are, and how better trained the operators are, system failures become unavoidable.

Given the fact that failures are inevitable, the need for effective means of detecting them is quite apparent if we consider their consequences and impacts not just on the systems involved but on the society as a whole. Moreover we note that effective methods of failure diagnosis can not only help avoid the undesirable effects of failures, but can also enhance the operational goals of industries. Improved quality of performance, product integrity and reliability, and reduced cost of equipment maintenance and service are some major benefits that accurate diagnosis schemes can provide, especially for service and product oriented industries such as home and building environment control, office automation, automobile manufacturing, and semiconductor manufacturing. Thus, we see that accurate and timely methods of failure diagnosis can enhance the safety, reliability, availability, quality, and economy of industrial processes.

The need of automated mechanisms for the timely and accurate diagnosis of failures is well understood and appreciated both in industry and in academia. A great deal of research effort has been and is being spent in the design and development of automated diagnostic systems, and a variety of schemes, differing both in their theoretical framework and in their design and implementation philosophy, have been proposed.

In diagnosis approach two different problems can be solved: the problem of diagnosis and the problem of diagnosability.

Solving a problem of diagnosis means that we associate to each observed string of events a diagnosis state, such as "normal" or "faulty" or "uncertain". Solving a problem of diagnosability is equivalent to determine if the system is diagnosable, i.e., to determine if, once a fault has occurred, the system can detect its occurrence in a finite number of steps.

The diagnosis of discrete event systems (DES) is a research area that has received a lot of attention in the last years and has been motivated by the practical need of ensuring the correct and safe functioning of large complex systems. As discussed in the next session the first results have been presented within the framework of automata. More recently, the diagnosis problem has also been addressed using Petri nets (PNs). In fact, the use of Petri nets offers significant advantages because of their twofold representation: graphical and mathematical. Moreover, the intrinsically distributed nature of PNs where the notion of state (i.e., marking) and action (i.e., transition) is local reduces the computational complexity involved in solving a diagnosis problem.

In this paper we summarize our main contributions on diagnosis of DES using PNs (Giua and Seatzu, 2005; Cabasino et al., 2008; Lai et al., 2008; Cabasino et al., 2009). In particular, we focus on arbitrary labeled PNs where the observable events are the labels associated to transitions, while faults are modeled as silent transitions. We assume that there may also be transitions modeling a regular behavior, that are silent as well. Moreover, two or more transitions that may be simultaneously enabled may share the same label, thus they are undistinguishable. Our diagnosis approach is based on the definition of four diagnosis states modeling different degrees of alarm and it applies to all systems whose unobservable subnet is acyclic. Two are the main advantages of our procedure. First, we do not need an exhaustive enumeration of the states in which the system may be: this is due to the introduction of basis markings. Secondly, in the case of bounded net systems we can move off-line the most burdensome part of the procedure building a finite graph called basis reachability graph.

The paper is organized as follows. In Section 2 the state of art of diagnosis for discrete event systems is illustrated. In Section 3 we provide a background on PNs. In Sections 4 and 5 are introduced the definitions of minimal explanations, justifications and basis markings, that are the basic notions of our diagnosis approach. In Section 6 the diagnosis states are defined and a characterization of them in terms of basis markings and j-vectors is given. In Section 7 we show how the most burdensome part of the procedure can be moved offline in the case of bounded PNs. In Section 8 we present the MATLAB toolbox developed by our group for PNs diagnosis and in Section 9 we present some numerical results obtained applying our tool to a parametric model of manufacturing system. In Section 10 we draw the conclusions.

2 LITERATURE REVIEW

In this section we present the state of art of diagnosis of DES using automata and PNs.

2.1 Diagnosis of DES using Automata

In the contest of DES several original theoretical approaches have been proposed using *automata*.

In (Lin, 1994) and (Lin et al., 1993) a state-based DES approach to failure diagnosis is proposed. The problems of off-line and on-line diagnosis are addressed separately and notions of diagnosability in both of these cases are presented. The authors give an algorithm for computing a diagnostic control, i.e., a sequence of test commands for diagnosing system failures. This algorithm is guaranteed to converge if the system satisfies the conditions for on-line diagnosability.

In (Sampath et al., 1995) and (Sampath et al., 1996) the authors propose an approach to failure diagnosis where the system is modeled as a DES in which the failures are treated as unobservable events. The level of detail in a discrete event model appears to be quite adequate for a large class of systems and for a wide variety of failures to be diagnosed. The approach is applicable whenever failures cause a distinct change in the system status but do not necessarily bring the system to a halt. In (Sampath et al., 1995) a definition of diagnosability in the framework of formal languages is provided and necessary and sufficient conditions for diagnosability of systems are established. Also presented in (Sampath et al., 1995) is a systematic approach to solve the problem of diagnosis using diagnosers.

In (Sampath et al., 1998) the authors present an integrated approach to control and diagnosis. More specifically, authors present an approach for the design of diagnosable systems by appropriate design of the system controller and this approach is called active diagnosis. They formulate the active diagnosis problem as a supervisory control problem. The adopted procedure for solving the active diagnosis problem is the following: given the non-diagnosable language generated by the system of interest, they first select an “appropriate” sublanguage of this language as the legal language. Choice of the legal language is a design issue and typically depends on considerations such as acceptable system behavior (which ensures that the system behavior is not restricted more than necessary in order to eventually make it diagnosable) and detection delay for the failures. Once the appropriate legal language is chosen, they then design a controller (diagnostic controller), that achieves a closed-loop language that is within the legal language and is diagnosable. This controller is designed based on the formal framework and the synthesis techniques that supervisory control theory provides, with the additional constraint of diagnosability.

In (Debouk et al., 2000) is addressed the problem of failure diagnosis in DES with decentralized information. Debouk *et al.* propose a coordinated decentralized architecture consisting of two local sites communicating with a coordinator that is responsible for diagnosing the failures occurring in the system. They extend the notion of diagnosability, originally introduced in (Sampath et al., 1995) for centralized systems, to the proposed coordinated decentralized architecture. In particular, they specify three protocols that realize the proposed architecture and analyze the diagnostic properties of these protocols.

In (Boel and van Schuppen, 2002) the authors address the problem of synthesizing communication protocols and failure diagnosis algorithms for decentralized failure diagnosis of DES with costly communication between diagnosers. The costs on the communication channels may be described in terms of bits and complexity. The costs of communication and computation force the trade-off between the control objective of failure diagnosis and that of minimization of the costs of communication and computation. The results of this paper is an algorithm for decentralized failure diagnosis of DES for the special case of only two diagnosers.

In (Zad et al., 2003) a state-based approach for online passive fault diagnosis is presented. In this framework, the system and the diagnoser (the fault detection system) do not have to be initialized at the same time. Furthermore, no information about the state or even the condition (failure status) of the system before the initiation of diagnosis is required. The design of the fault detection system, in the worst case, has exponential complexity. A model reduction scheme with polynomial time complexity is introduced to reduce the computational complexity of the design. Diagnosability of failures is studied, and necessary and sufficient conditions for failure diagnosability are derived.

2.2 Diagnosis of DES using Petri Nets

Among the first pioneer works dealing with PNs, we recall the approach of Prock. In (Prock, 1991) the author proposes an on-line technique for fault detection that is based on monitoring the number of tokens residing into P-invariants: when the number of tokens inside P-invariants changes, then the error is detected.

In (Sreenivas and Jafari, 1993) the authors employ time PNs to model the DES controller and backfiring transitions to determine whether a given state is invalid. Later on, time PNs have been employed in (Ghazel et al., 2005) to propose a monitoring approach for DES with unobservable events and to represent the “a priori” known behavior of the system, and track on-line its state to identify the events that occur.

In (Hadjicostis and Veghese, 1999) the authors use PN models to introduce redundancy into the system and additional P-invariants allow the detection and isolation of faulty markings.

Redundancy into a given PN is used in (Wu and Hadjicostis, 2005) to enable fault detection and identification using algebraic decoding techniques. In this paper Wu and Hadjicostis consider two types of faults: place faults that corrupt the net marking, and transition faults that cause a not correct update of the marking after event occurrence. Although this approach is general, the net marking has to be periodically observable even if unobservable events occur. Analogously, in (Lefebvre and Delherm, 2007) the authors investigate on the determination of the set of places that must be observed for the exact and immediate estimation of faults occurrence.

In (Ruiz-Beltràn et al., 2007) Interpreted PNs are employed to model the system behavior that includes both events and states partially observable. Based on the Interpreted PN model derived from an on-line methodology, a scheme utilizing a solution of a programming problem is proposed to solve the problem of diagnosis.

Note that, all papers in this topic assume that faults are modeled by unobservable transitions. However, while the above mentioned papers assume that the marking of certain places may be observed, a series of papers have been recently presented that are based on the assumption that no place is observable (Basile et al., 2008; Benveniste et al., 2003; Dotoli et al., 2008; Genc and Lafortune, 2007).

In particular, in (Genc and Lafortune, 2007) the authors propose a diagnoser on the basis of a modular approach that performs the diagnosis of faults in each module. Subsequently, the diagnosers recover the monolithic diagnosis information obtained when all the modules are combined into a single module that preserves the behavior of the underlying modular system. A communication system connects the different modules and updates the diagnosis information. Even if the approach does not avoid the state explosion problem, an improvement is obtained when the system can be modeled as a collection of PN modules coupled through common places.

The main advantage of the approaches in (Genc and Lafortune, 2007) consists in the fact that, if the net is bounded, the diagnoser may be constructed off-line, thus moving off-line the most burdensome part of the procedure. Nevertheless, a characterization of the set of markings consistent with the actual observation is needed. Thus, large memory may be required.

An improvement in this respect has been given in (Benveniste et al., 2003; Basile et al., 2008; Dotoli et al., 2008).

In particular, in (Benveniste et al., 2003) a net unfolding approach for designing an on-line asynchronous diagnoser is used. The state explosion is avoided but the on-line computation can be high due to the on-line building of the PN structures by means of the unfolding.

In (Basile et al., 2008) the diagnoser is built on-line by defining and solving Integer Linear Programming (ILP) problems. Assuming that the fault transitions are not observable, the net marking is computed by the state equation and, if the marking has negative components, an unobservable sequence is occurred. The linear programming solution provides the sequence and detects the fault occurrences. Moreover, an off-line analysis of the PN structure reduces the computational complexity of the ILP problem.

In (Dotoli et al., 2008), in order to avoid the redesign and the redefinition of the diagnoser when the structure of the system changes, the authors propose a diagnoser that works on-line. In particular, it waits for an observable event and an algorithm decides whether the system behavior is normal or may exhibit some possible faults. To this aim, some ILP problems are defined and provide eventually the minimal sequences of unobservable transitions containing the faults that may have occurred. The proposed approach is a general technique since no assumption is imposed on the reachable state set that can be unlimited, and only few properties must be fulfilled by the structure of the PN modeling the system fault behavior.

We also proposed a series of contributions dealing with diagnosis of PNs (Giua and Seatzu, 2005; Cabasino et al., 2008; Lai et al., 2008; Cabasino et al., 2009). Our main results are summarized in the rest of the paper.

Note that none of the above mentioned papers regarding PNs deal with *diagnosability*, namely none of them provide a procedure to determine a priori if a system is *diagnosable*, i.e., if it is possible to reconstruct the occurrence of fault events observing words of finite length.

In fact, whereas this problem has been extensively studied within the framework of automata as discussed above, in the PN framework very few results have been presented.

The first contribution on diagnosability of PNs was given in (Ushio et al., 1998). They extend a necessary and sufficient condition for diagnosability in (Sampath et al., 1995; Sampath et al., 1996) to unbounded PN. They assume that the set of places is partitioned into observable and unobservable places, while all transitions are unobservable in the sense that their occurrences cannot be observed. Starting from the PN they build a diagnoser called *simple* ω *diagnoser* that gives them sufficient conditions for diagnosability of unbounded PNs.

In (Chung, 2005) the authors, in contrast with Ushio's paper, assumes that part of the transitions of the PN modelling is observable and shows as the additional information from observed transitions in general adds diagnosability to the analysed system. Moreover starting from the diagnoser he proposes an automaton called *verifier* that allows a polynomial check mechanism on diagnosability but for finite state automata models.

In (Wen and Jeng, 2005) the authors propose an approach to test diagnosability by checking the structure property of T-invariant of the nets. They use Ushio's diagnoser to prove that their method is correct, however they don't construct a diagnoser for the system to do diagnosis. In (Wen et al., 2005) they also present an algorithm, based on a linear programming problem, of polynomial complexity in the number of nodes for computing a sufficient condition of diagnosability of DES modeled by PN.

3 BACKGROUND

In this section we recall the formalism used in the paper. For more details on PNs we refer to (Murata, 1989).

A *Place/Transition net* (P/T net) is a structure $N = (P, T, Pre, Post)$, where P is a set of m places; T is a set of n transitions; $Pre : P \times T \rightarrow \mathbb{N}$ and $Post : P \times T \rightarrow \mathbb{N}$ are the *pre*– and *post*– incidence functions that specify the arcs; $C = Post - Pre$ is the incidence matrix.

A *marking* is a vector $M : P \rightarrow \mathbb{N}$ that assigns to each place of a P/T net a non–negative integer number of tokens, represented by black dots. We denote $M(p)$ the marking of place p. A P/T *system* or *net system* $\langle N, M_0 \rangle$ is a net N with an initial marking M_0. A transition t is enabled at M iff $M \geq Pre(\cdot, t)$ and may fire yielding the marking $M' = M + C(\cdot, t)$. We write $M\, [\sigma\rangle$ to denote that the sequence of transitions $\sigma = t_{j_1} \cdots t_{j_k}$ is enabled at M, and we write $M\, [\sigma\rangle\, M'$ to denote that the firing of σ yields M'. We also write $t \in \sigma$ to denote that a transition t is contained in σ.

The set of all sequences that are enabled at the initial marking M_0 is denoted $L(N,M_0)$, i.e., $L(N,M_0) = \{\sigma \in T^* \mid M_0[\sigma\rangle\}$.

Given a sequence $\sigma \in T^*$, we call $\pi : T^* \to \mathbb{N}^n$ the function that associates to σ a vector $y \in \mathbb{N}^n$, named the *firing vector* of σ. In particular, $y = \pi(\sigma)$ is such that $y(t) = k$ if the transition t is contained k times in σ.

A marking M is *reachable* in $\langle N, M_0\rangle$ iff there exists a firing sequence σ such that $M_0\ [\sigma\rangle\ M$. The set of all markings reachable from M_0 defines the *reachability set* of $\langle N, M_0\rangle$ and is denoted $R(N,M_0)$.

A PN having no directed circuits is called *acyclic*. A net system $\langle N, M_0\rangle$ is *bounded* if there exists a positive constant k such that, for $M \in R(N,M_0)$, $M(p) \leq k$.

A *labeling function* $\mathcal{L} : T \to L \cup \{\varepsilon\}$ assigns to each transition $t \in T$ either a symbol from a given alphabet L or the empty string ε.

We denote as T_u the set of transitions whose label is ε, i.e., $T_u = \{t \in T \mid \mathcal{L}(t) = \varepsilon\}$. Transitions in T_u are called *unobservable* or *silent*. We denote as T_o the set of transitions labeled with a symbol in L. Transitions in T_o are called *observable* because when they fire their label can be observed. Note that in this paper we assume that the same label $l \in L$ can be associated to more than one transition. In particular, two transitions $t_1, t_2 \in T_o$ are called *undistinguishable* if they share the same label, i.e., $\mathcal{L}(t_1) = \mathcal{L}(t_2)$. The set of transitions sharing the same label l are denoted as T_l.

In the following we denote as C_u (C_o) the restriction of the incidence matrix to T_u (T_o) and denote as n_u and n_o, respectively, the cardinality of the above sets. Moreover, given a sequence $\sigma \in T^*$, $P_u(\sigma)$, resp., $P_o(\sigma)$, denotes the projection of σ over T_u, resp., T_o.

We denote as w the word of events associated to the sequence σ, i.e., $w = P_o(\sigma)$. Note that the length of a sequence σ (denoted $|\sigma|$) is always greater than or equal to the length of the corresponding word w (denoted $|w|$). In fact, if σ contains k' transitions in T_u then $|\sigma| = k' + |w|$.

Definition 3.1 (Cabasino et al., 2009). Let $\langle N, M_0\rangle$ be a labeled net system with labeling function $\mathcal{L} : T \to L \cup \{\varepsilon\}$, where $N = (P, T, Pre, Post)$ and $T = T_o \cup T_u$. Let $w \in L^*$ be an observed word. We define

$$\mathcal{S}(w) = \{\sigma \in L(N,M_0) \mid P_o(\sigma) = w\}$$

the set of firing sequences *consistent* with $w \in L^*$, and

$$\mathcal{C}(w) = \{M \in \mathbb{N}^m \mid \exists \sigma \in T^* : P_o(\sigma) = w \wedge M_0[\sigma\rangle M\}$$

the set of markings *consistent* with $w \in L^*$. ■

In plain words, given an observation w, $\mathcal{S}(w)$ is the set of sequences that may have fired, while $\mathcal{C}(w)$ is the set of markings in which the system may actually be.

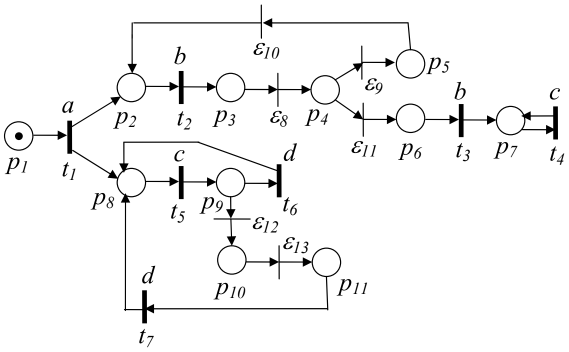


Figure 1: A PN system modeling.

Example 3.2. Let us consider the PN in Figure 1. Let us assume $T_o = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7\}$ and $T_u = \{\varepsilon_8, \varepsilon_9, \varepsilon_{10}, \varepsilon_{11}, \varepsilon_{12}, \varepsilon_{13}\}$, where for a better understanding unobservable transitions have been denoted ε_i rather than t_i. The labeling function is defined as follows: $\mathcal{L}(t_1) = a$, $\mathcal{L}(t_2) = \mathcal{L}(t_3) = b$, $\mathcal{L}(t_4) = \mathcal{L}(t_5) = c$, $\mathcal{L}(t_6) = \mathcal{L}(t_7) = d$.

First let us consider $w = ab$. The set of firing sequences that is consistent with w is $\mathcal{S}(w) = \{t_1 t_2,\ t_1 t_2 \varepsilon_8, t_1 t_2 \varepsilon_8 \varepsilon_9, t_1 t_2 \varepsilon_8 \varepsilon_9 \varepsilon_{10}, t_1 t_2 \varepsilon_8 \varepsilon_{11}\}$, and the set of markings consistent with w is $\mathcal{C}(w) = \{[0\ 0\ 1\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0]^T, [0\ 0\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 0]^T, [0\ 0\ 0\ 0\ 1\ 0\ 0\ 1\ 0\ 0\ 0]^T, [0\ 1\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0]^T, [0\ 0\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 0]^T\}$.

If we consider $w = acd$ the set of firing sequences that are consistent with w is $\mathcal{S}(w) = \{t_1 t_5 t_6, t_1 t_5 \varepsilon_{12} \varepsilon_{13} t_7\}$, and the set of markings consistent with w is $\mathcal{C}(w) = \{[0\ 1\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0]^T\}$. Thus two different firing sequences may have fired (the second one also involving silent transitions), but they both lead to the same marking. ■

4 MINIMAL EXPLANATIONS AND MINIMAL E-VECTORS

In this section we present the notions of minimal explanations and minimal e-vectors for labeled PNs. First we introduce notions of explanations for unlabeled PNs, secondly we define when an explanation is minimal and finally we extend these concepts to labeled PN.

Definition 4.1 (Cabasino et al., 2008). Given a marking M and an observable transition $t \in T_o$, we define

$$\Sigma(M,t) = \{\sigma \in T_u^* \mid M[\sigma\rangle M',\ M' \geq Pre(\cdot,t)\}$$

the set of *explanations* of t at M, and

$$Y(M,t) = \pi(\Sigma(M,t))$$

the *e-vectors* (or *explanation vectors*), i.e., firing vectors associated to the explanations. ■

Thus $\Sigma(M,t)$ is the set of unobservable sequences whose firing at M enables t. Among the above sequences we want to select those whose firing vector is minimal. The firing vector of these sequences are called *minimal e-vectors*.

Definition 4.2 (Cabasino et al., 2008). Given a marking M and a transition $t \in T_o$, we define

$$\Sigma_{\min}(M,t) = \{\sigma \in \Sigma(M,t) \mid \nexists\, \sigma' \in \Sigma(M,t) \ : \ \pi(\sigma') \lneqq \pi(\sigma)\}$$

the set of *minimal explanations* of t at M, and we define

$$Y_{\min}(M,t) = \pi(\Sigma_{\min}(M,t))$$

the corresponding set of *minimal e-vectors*. ■

In (Corona et al., 2004) we proved that, if the unobservable subnet is acyclic and backward conflict-free, then $|Y_{\min}(M,t)| = 1$.

Different approaches can be used to compute $Y_{\min}(M,t)$, e.g., (Boel and Jiroveanu, 2004; Jiroveanu and Boel, 2004). In (Cabasino et al., 2008) we suggested an approach that terminates finding all vectors in $Y_{\min}(M,t)$ if applied to nets whose unobservable subnet is acyclic. It simply requires algebraic manipulations, and is inspired by the procedure proposed in (Martinez and Silva, 1982) for the computation of minimal P-invariants. For the sake of brevity, this algorithm is not reported here.

In the case of labeled PNs what we observe are symbols in L. Thus, it is useful to compute the following sets.

Definition 4.3 (Cabasino et al., 2009). Given a marking M and an observation $l \in L$, we define the set of *minimal explanations of l at M* as

$$\hat{\Sigma}_{\min}(M,l) = \cup_{t \in T_l} \cup_{\sigma \in \Sigma_{\min}(M,t)} (t,\sigma),$$

i.e., the set of pairs (transition labeled l; corresponding minimal explanation), and we define the set of *minimal e-vectors of l at M* as

$$\hat{Y}_{\min}(M,l) = \cup_{t \in T_l} \cup_{e \in Y_{\min}(M,t)} (t,e),$$

i.e., the set of pairs (transition labeled l; corresponding minimal e-vector). ■

Thus, $\hat{\Sigma}_{\min}(M,l)$ is the set of pairs whose first element is the transition labeled l and whose second element is the corresponding minimal explanation $\sigma \in \Sigma_{\min}(M,t)$, namely the corresponding sequence of unobservable transitions whose firing at M enables l and whose firing vector is minimal. Moreover, $\hat{Y}_{\min}(M,l)$ is the set of pairs whose first element is the transition labeled l and whose second element is the firing vector $e \in Y_{\min}(M,t)$ corresponding to the second element in $\hat{\Sigma}_{\min}(M,l)$.

Obviously, $\hat{\Sigma}_{\min}(M,l)$ and $\hat{Y}_{\min}(M,l)$ are a generalization of the sets of minimal explanations and minimal e-vectors introduced for unlabeled PNs with unobservable transitions. Moreover, in the above sets $\hat{\Sigma}_{\min}(M,l)$ and $\hat{Y}_{\min}(M,l)$ different sequences σ and different e-vectors e, respectively, are associated in general to the same $t \in T_l$.

5 BASIS MARKINGS AND J-VECTORS

In this section we introduce the definitions of basis markings and justifications that are the crucial notions of our diagnosis approach.

In particular, given a sequence of observed events $w \in L^*$, a basis marking M_b is a marking reached from M_0 with the firing of the observed word w and of all unobservable transitions whose firing is necessary to enable w. Note that, in general several sequences $\sigma_o \in T_o^*$ may correspond to the same w, i.e., there are several sequences of observable transitions such that $\mathcal{L}(\sigma_o) = w$ that may have actually fired. Moreover, in general, to any of such sequences σ_o a different sequence of unobservable transitions interleaved with it is necessary to make it firable at the initial marking. Thus we need to introduce the following definition of pairs (sequence of transitions in T_o labeled w; corresponding *justification*).

Definition 5.1 (Cabasino et al., 2009). Let $\langle N, M_0 \rangle$ be a net system with labeling function $\mathcal{L} : T \to L \cup \{\varepsilon\}$, where $N = (P,T,Pre,Post)$ and $T = T_o \cup T_u$. Let $w \in L^*$ be a given observation. We define

$$\begin{aligned} \hat{\mathcal{J}}(w) \quad = \{ & (\sigma_o,\sigma_u),\ \sigma_o \in T_o^*,\ \mathcal{L}(\sigma_o) = w,\ \sigma_u \in T_u^* \ \mid \\ & [\exists \sigma \in \mathcal{S}(w) \ : \ \sigma_o = P_o(\sigma),\ \sigma_u = P_u(\sigma)] \wedge \\ & [\nexists \sigma' \in \mathcal{S}(w) : \ \sigma_o = P_o(\sigma'),\ \sigma'_u = P_u(\sigma') \wedge \\ & \qquad \pi(\sigma'_u) \lneqq \pi(\sigma_u)]\} \end{aligned}$$

the set of pairs (sequence $\sigma_o \in T_o^*$ with $\mathcal{L}(\sigma_o) = w$; corresponding *justification* of w). Moreover, we define

$$\begin{aligned} \hat{Y}_{\min}(M_0,w) = \{ & (\sigma_o,y),\ \sigma_o \in T_o^*, \mathcal{L}(\sigma_o) = w, y \in \mathbb{N}^{n_u} \mid \\ & \exists (\sigma_o,\sigma_u) \in \hat{\mathcal{J}}(w) : \pi(\sigma_u) = y\} \end{aligned}$$

the set of pairs (sequence $\sigma_o \in T_o^*$ with $\mathcal{L}(\sigma_o) = w$; corresponding *j-vector*). ■

In simple words, $\hat{\mathcal{J}}(w)$ is the set of pairs whose first element is the sequence $\sigma_o \in T_o^*$ labeled w and whose second element is the corresponding sequence of unobservable transitions interleaved with σ_o whose firing enables σ_o and whose firing vector is minimal.

The firing vectors of these sequences are called *j-vectors*.

Definition 5.2 (Cabasino et al., 2009). Let $\langle N, M_0 \rangle$ be a net system with labeling function $\mathcal{L} : T \rightarrow L \cup \{\varepsilon\}$, where $N = (P, T, Pre, Post)$ and $T = T_o \cup T_u$. Let w be a given observation and $(\sigma_o, \sigma_u) \in \hat{\mathcal{J}}(w)$ be a generic pair (sequence of observable transitions labeled w; corresponding minimal justification). The marking

$$M_b = M_0 + C_u \cdot y + C_o \cdot y', \qquad y = \pi(\sigma_u),\ y' = \pi(\sigma_o),$$

i.e., the marking reached firing σ_o interleaved with the minimal justification σ_u, is called *basis marking* and y is called its *j-vector* (or *justification-vector*). ■

Obviously, because in general more than one justification exists for a word w (the set $\hat{\mathcal{J}}(w)$ is generally not a singleton), the basis marking may be not unique as well.

Definition 5.3 (Cabasino et al., 2009). Let $\langle N, M_0 \rangle$ be a net system with labeling function $\mathcal{L} : T \rightarrow L \cup \{\varepsilon\}$, where $N = (P, T, Pre, Post)$ and $T = T_o \cup T_u$. Let $w \in L^*$ be an observed word. We define

$$\begin{aligned} \mathcal{M}(w) = \{(M, y) \mid\ & (\exists \sigma \in \mathcal{S}(w) \ :\ M_0[\sigma\rangle M) \wedge \\ & (\exists (\sigma_o, \sigma_u) \in \hat{\mathcal{J}}(w) \ :\ \sigma_o = P_o(\sigma), \\ & \sigma_u = P_u(\sigma),\ y = \pi(\sigma_u))\} \end{aligned}$$

the set of pairs (basis marking; relative j-vector) that are *consistent* with $w \in L^*$. ■

Note that the set $\mathcal{M}(w)$ does not keep into account the sequences of observable transitions that may have actually fired. It only keeps track of the basis markings that can be reached and of the firing vectors relative to sequences of unobservable transitions that have fired to reach them. Indeed, this is the information really significant when performing diagnosis. The notion of $\mathcal{M}(w)$ is fundamental to provide a recursive way to compute the set of minimal explanations.

Proposition 5.4 (Cabasino et al., 2009). Given a net system $\langle N, M_0 \rangle$ with labeling function $\mathcal{L} : T \rightarrow L \cup \{\varepsilon\}$, where $N = (P, T, Pre, Post)$ and $T = T_o \cup T_u$. Assume that the unobservable subnet is acyclic. Let $w = w'l$ be a given observation.

The set $\hat{Y}_{\min}(M_0, wl)$ is defined as:

$$\begin{aligned} \hat{Y}_{\min}(M_0, wl) = \quad & \{(\sigma_o, y) \mid \sigma_o = \sigma_0' t \wedge y = y' + e \ : \\ & (\sigma_o', y') \in \hat{Y}_{\min}(M_0, w), \\ & (t, e) \in \hat{Y}_{\min}(M_b', l) \text{ and } \mathcal{L}(t) = l\}, \end{aligned}$$

where $M_b' = M_0 + C_u \cdot y' + C_o \cdot \sigma_o'$.

Example 5.5. Let us consider the PN in Figure 1 previously introduced in Example 3.2.

Let us assume $w = acd$. The set of justifications is $\hat{\mathcal{J}}(w) = \{(t_1 t_5 t_6, \varepsilon), (t_1 t_5 t_7, \varepsilon_{12} \varepsilon_{13})\}$ and the set of j-vectors is $\hat{Y}_{min}(M_0, w) = \{(t_1 t_5 t_6, \vec{0}), (t_1 t_5 t_7, [0\ 0\ 0\ 0\ 1\ 1]^T)\}$. The above j-vectors lead to the same basis marking $M_b = [0\ 1\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0]^T$ thus $\mathcal{M}(w) = \{(M_b, \vec{0}), (M_b, [0\ 0\ 0\ 0\ 1\ 1]^T)\}$.

Now, let us consider $w = ab$. In this case $\hat{\mathcal{J}}(w) = \{(t_1 t_2, \varepsilon)\}$, $\hat{Y}_{min}(M_0, w) = \{(t_1 t_2, \vec{0})\}$ and the basis marking is the same as in the previous case, namely $M_b = [0\ 1\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0]^T$, thus $\mathcal{M}(w) = \{(M_b, \vec{0})\}$. ■

Under the assumption of acyclicity of the unobservable subnet, the set $\mathcal{M}(w)$ can be easily constructed as follows.

Algorithm 5.6 (Computation of the basis markings and j-vectors).

```
1. Let $w = \varepsilon$.
2. Let $\mathcal{M}(w) = \{(M_0, \vec{0})\}$.
3. Wait until a new label $l$ is observed.
4. Let $w' = w$ and $w = w'l$.
5. Let $\mathcal{M}(w) = \emptyset$.
6. For all $M'$ such that $(M', y') \in \mathcal{M}(w')$ , do
  6.1. for all $t \in T_l$, do
    6.1.1. for all $e \in Y_{\min}(M', t)$, do
    6.1.1.1. let $M = M' + C_u \cdot e + C(\cdot, t)$,
    6.1.1.2. for all $y'$ such that $(M', y') \in \mathcal{M}(w')$, do
        6.1.2.1. let $y = y' + e$,
        6.1.2.2. let $\mathcal{M}(w) = \mathcal{M}(w) \cup \{(M, y)\}$.
7. Goto step 3.
```

■

In simple words, the above algorithm can be explained as follows. We assume that a certain word w (that is equal to the empty string at the initial step) has been observed. Then, a new observable t fires and we observe its label $\mathcal{L}(t)$ (e.g., l). We consider all basis markings at the observation w' before the firing of t, and we select among them those that may have allowed the firing of at least one transition $t \in T_l$, also taking into account that this may have required the firing of appropriate sequences of unobservable transitions. In particular, we focus on the minimal explanations, and thus on the corresponding minimal e-vectors (step 6.1.1). Finally, we update the set $\mathcal{M}(w)$ including all pairs of new basis markings and j-vectors, taking into account that for each basis marking at w' it may correspond more than one j-vector.

Let us now recall the following result.

Definition 5.7 (Cabasino et al., 2008). Let $\langle N, M_0 \rangle$ be a net system where $N = (P, T, Pre, Post)$ and $T = T_o \cup T_u$. Assume that the unobservable subnet is acyclic. Let $w \in T_o^*$ be an observed word. We denote

$$\mathcal{M}_{basis}(w) = \{M \in \mathbb{N}^m \ \mid \ \exists y \in \mathbb{N}^{n_u} \ \text{ and } \ (M, y) \in \mathcal{M}(w)\}$$

the set of basis markings at w. Moreover, we denote as

$$\mathcal{M}_{basis} = \bigcup_{w \in T_o^*} \mathcal{M}_{basis}(w)$$

the set of all basis markings for any observation w. ■

Note that if the net system is bounded then the set $\mathcal{M}_{basis}$ is *finite* being the set of basis markings a subset of the reachability set.

Theorem 5.8 (Cabasino et al., 2008). Let us consider a net system $\langle N, M_0 \rangle$ whose unobservable subnet is acyclic. For any $w \in L^*$ it holds that

$$\begin{aligned} \mathcal{C}(w) = \{M \in \mathbb{N}^m \mid \quad & M = M_b + C_u \cdot y \ : \\ & y \geq \vec{0} \ \text{ and } \ M_b \in \mathcal{M}_{basis}(w)\}. \end{aligned}$$

6 DIAGNOSIS USING PETRI NETS

Assume that the set of unobservable transitions is partitioned into two subsets, namely $T_u = T_f \cup T_{reg}$ where T_f includes all fault transitions (modeling anomalous or fault behavior), while T_{reg} includes all transitions relative to unobservable but regular events. The set T_f is further partitioned into r different subsets T_f^i, where $i = 1, \ldots, r$, that model the different fault classes.

The following definition introduces the notion of *diagnoser*.

Definition 6.1 (Cabasino et al., 2009). A *diagnoser* is a function $\Delta : L^* \times \{T_f^1, T_f^2, \ldots, T_f^r\} \to \{0,1,2,3\}$ that associates to each observation $w \in L^*$ and to each fault class T_f^i, $i = 1, \ldots, r$, a *diagnosis state*.

- $\Delta(w, T_f^i) = 0$ if for all $\sigma \in \mathcal{S}(w)$ and for all $t_f \in T_f^i$ it holds $t_f \notin \sigma$.
  In such a case the ith fault cannot have occurred, because none of the firing sequences consistent with the observation contains fault transitions of class i.
- $\Delta(w, T_f^i) = 1$ if:
  (i) there exist $\sigma \in \mathcal{S}(w)$ and $t_f \in T_f^i$ such that $t_f \in \sigma$ but
  (ii) for all $(\sigma_o, \sigma_u) \in \hat{\mathcal{J}}(w)$ and for all $t_f \in T_f^i$ it holds that $t_f \notin \sigma_u$.
  In such a case a fault transition of class i may have occurred but is not contained in any justification of w.
- $\Delta(w, T_f^i) = 2$ if there exist $(\sigma_o, \sigma_u), (\sigma'_o, \sigma'_u) \in \hat{\mathcal{J}}(w)$ such that
  (i) there exists $t_f \in T_f^i$ such that $t_f \in \sigma_u$;
  (ii) for all $t_f \in T_f^i$, $t_f \notin \sigma'_u$.
  In such a case a fault transition of class i is contained in one (but not in all) justification of w.
- $\Delta(w, T_f^i) = 3$ if for all $\sigma \in \mathcal{S}(w)$ there exists $t_f \in T_f^i$ such that $t_f \in \sigma$.
  In such a case the ith fault must have occurred, because all firable sequences consistent with the observation contain at least one fault in T_f^i. ■

Example 6.2. Let us consider the PN in Figure 1 previously introduced in Example 3.2. Let $T_f = \{\varepsilon_{11}, \varepsilon_{12}\}$. Assume that the two fault transitions belong to different fault classes, i.e., $T_f^1 = \{\varepsilon_{11}\}$ and $T_f^2 = \{\varepsilon_{12}\}$.

Let us observe $w = a$. Then $\Delta(w, T_f^1) = \Delta(w, T_f^2) = 0$, being $\hat{\mathcal{J}}(w) = \{(t_1, \varepsilon)\}$ and $\mathcal{S}(w) = \{t_1\}$. In words no fault of both fault classes can have occurred.

Let us observe $w = ab$. Then $\Delta(w, T_f^1) = 1$ and $\Delta(w, T_f^2) = 0$, being $\hat{\mathcal{J}}(w) = \{(t_1t_2, \varepsilon)\}$ and $\mathcal{S}(w) = \{t_1t_2, t_1t_2\varepsilon_8, t_1t_2\varepsilon_8\varepsilon_9, t_1t_2\varepsilon_8\varepsilon_9\varepsilon_{10}, t_1t_2\varepsilon_8\varepsilon_{11}\}$. This means that a fault of the second fault class may have occurred (e.g. $t_1t_2\varepsilon_8\varepsilon_{11}$) but it is not contained in any justification of ab, while no fault of the first fault class can have occurred.

Now, let us consider $w = abb$. In this case $\Delta(w, T_f^1) = 2$ and $\Delta(w, T_f^2) = 0$, being $\hat{\mathcal{J}}(w) = \{(t_1t_2t_2, \varepsilon_8\varepsilon_9\varepsilon_{10}), (t_1t_2t_3, \varepsilon_8\varepsilon_{11})\}$ and $\mathcal{S}(w) = \{t_1t_2\varepsilon_8\varepsilon_9\varepsilon_{10}t_2, t_1t_2\varepsilon_8\varepsilon_9\varepsilon_{10}t_2\varepsilon_8, t_1t_2\varepsilon_8\varepsilon_9\varepsilon_{10}t_2\varepsilon_8\varepsilon_9, t_1t_2\varepsilon_8\varepsilon_9\varepsilon_{10}t_2\varepsilon_8\varepsilon_9\varepsilon_{10}, t_1t_2\varepsilon_8\varepsilon_9\varepsilon_{10}t_2\varepsilon_8\varepsilon_{11}\}$. This means that no fault of the first fault class can have occurred, while a fault of the second fault class may have occurred since one justification does not contain ε_{11} and one justification contains it.

Finally, let us consider $w = abbccc$. In this case $\Delta(w, T_f^1) = 1$ and $\Delta(w, T_f^2) = 3$. In fact since $\hat{\mathcal{J}}(w) = \{(t_1t_2t_3t_5t_4t_4, \varepsilon_8\varepsilon_{11}), (t_1t_2t_3t_4t_5t_4, \varepsilon_8\varepsilon_{11}), (t_1t_2t_3t_4t_4t_5, \varepsilon_8\varepsilon_{11}), (t_1t_2t_3t_4t_4t_4, \varepsilon_8\varepsilon_{11})\}$ a fault of the first fault class must have occurred, while a fault of the second fault class may have occurred (e.g. $t_1t_2\varepsilon_8\varepsilon_{11}t_3t_4t_4t_5\varepsilon_{12}$) but it is not contained in any justification of w. ■

The following proposition presents how the diagnosis states can be characterized analyzing basis markings and justifications.

Proposition 6.3 (Cabasino et al., 2009). Consider an observed word $w \in L^*$.

- $\Delta(w, T_f^i) \in \{0,1\}$ iff for all $(M, y) \in \mathcal{M}(w)$ and for all $t_f \in T_f^i$ it holds $y(t_f) = 0$.
- $\Delta(w, T_f^i) = 2$ iff there exist $(M, y) \in \mathcal{M}(w)$ and $(M', y') \in \mathcal{M}(w)$ such that:
  (i) there exists $t_f \in T_f^i$ such that $y(t_f) > 0$,
  (ii) for all $t_f \in T_f^i$, $y'(t_f) = 0$.

- $\Delta(w,T_f^i) = 3$ iff for all $(M,y) \in \mathcal{M}(w)$ there exists $t_f \in T_f^i$ such that $y(t_f) > 0$.

The following proposition shows how to distinguish between diagnosis states 0 and 1.

Proposition 6.4 (Cabasino et al., 2009). For a PN whose unobservable subnet is acyclic, let $w \in L^*$ be an observed word such that for all $(M,y) \in \mathcal{M}(w)$ it holds $y(t_f) = 0 \ \forall \ t_f \in T_f^i$. Let us consider the constraint set

$$\mathcal{T}(M) \quad = \quad \left\{ \begin{array}{l} M + C_u \cdot z \geq \vec{0}, \\ \sum\limits_{t_f \in T_f^i} z(t_f) > 0, \\ z \in \mathbb{N}^{n_u}. \end{array} \right. \qquad (1)$$

- $\Delta(w,T_f^i) = 0$ if $\forall \ (M,y) \in \mathcal{M}(w)$ the constraint set (1) is not feasible.
- $\Delta(w,T_f^i) = 1$ if $\exists \ (M,y) \in \mathcal{M}(w)$ such that the constraint set (1) is feasible.

On the basis of the above two results, if the unobservable subnet is acyclic, diagnosis may be carried out by simply looking at the set $\mathcal{M}(w)$ for any observed word w and, should the diagnosis state be either 0 or 1, by additionally evaluating whether the corresponding integer constraint set (1) admits a solution.

Example 6.5. Let us consider the PN in Figure 1 where $T_f^1 = \{\varepsilon_{11}\}$ and $T_f^2 = \{\varepsilon_{12}\}$.

Let $w = ab$. In this case $\mathcal{M}(w) = \{(M_b^1, \vec{0})\}$, where $M_b^1 = [0\,1\,0\,0\,0\,0\,0\,1\,0\,0\,0]^T$. Being $\mathcal{T}(M_b^1)$ feasible only for the fault class T_f^1 it holds $\Delta(w,T_f^1) = 1$ and $\Delta(w,T_f^2) = 0$.

Let $w = abb$. It is $\mathcal{M}(w) = \{(M_b^1, [1\ 1\ 1\ 0\ 0\ 0]^T), (M_b^2, [1\ 0\ 0\ 1\ 0\ 0]^T)\}$, where $M_b^2 = [0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 0]^T$. It is $\Delta(w,T_f^1) = 2$ and $\Delta(w,T_f^2) = 0$ being both $\mathcal{T}(M_b^1)$ and $\mathcal{T}(M_b^2)$ not feasible.

Let $w = abbccc$. In this case $\mathcal{M}(w) = \{(M_b^3, [1\ 1\ 1\ 0\ 0\ 0]^T), (M_b^4, [1\ 1\ 1\ 0\ 0\ 0]^T)\}$, where $M_b^3 = [0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 0]^T$ and $M_b^4 = [0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 0]^T$. It is $\Delta(w,T_f^1) = 3$ and being $\mathcal{T}(M_b^4)$ feasible for the second fault class T_f^2 it holds $\Delta(w,T_f^2) = 1$. ■

7 BASIS REACHABILITY GRAPH

Diagnosis approach described in the previous section can be applied both to bounded and unbounded PNs. The proposed approach is an on-line approach that for each new observed event updates the diagnosis state for each fault class computing the set of basis markings and j-vectors. Moreover if for a given fault class is necessary to distinguish between diagnosis states 0 and 1, it is also necessary to solve for each basis marking M_b the constraint set $\mathcal{T}(M_b)$.

In this section we show that if the considered net system is bounded, the most burdensome part of the procedure can be moved off-line defining a graph called *Basis Reachability Graph* (BRG).

Definition 7.1. The BRG is a deterministic graph that has as many nodes as the number of possible basis markings.

To each node is associated a different basis marking M and a row vector with as many entries as the number of fault classes. The entries of this vector may only take binary values: 1 if $\mathcal{T}(M)$ is feasible, 0 otherwise.

Arcs are labeled with observable events in L and e-vectors. More precisely, an arc exists from a node containing the basis marking M to a node containing the basis marking M' if and only if there exists a transition t for which an explanation exists at M and the firing of t and one of its minimal explanations leads to M'. The arc going from M to M' is labeled $(\mathcal{L}(t), e)$, where $e \in Y_{\min}(M,t)$ and $M' = M + C_u \cdot e + C(\cdot,t)$. ■

Note that the number of nodes of the BRG is always finite being the set of basis markings a subset of the set of reachable markings, that is finite being the net bounded. Moreover, the row vector of binary values associated to the nodes of the BRG allows us to distinguish between the diagnosis state 1 or 0.

The main steps for the computation of the BRG in the case of labeled PNs are summarized in the following algorithm.

Algorithm 7.2 (Computation of the BRG).

```
1. Label the initial node $(M_0, x_0)$ where $\forall i = 1, \ldots, r$,
     $x_0(T_f^i) = \begin{cases} 1 & \text{if } \mathcal{T}(M_0) \text{ is feasible,} \\ 0 & \text{otherwise.} \end{cases}$
   Assign no tag to it.
2. While nodes with no tag exist
   select a node with no tag and do
   2.1. let $M$ be the marking in the node $(M, x)$,
   2.2. for all $l \in L$
     2.2.1. for all $t : L(t) = l \wedge Y_{\min}(M,t) \neq \emptyset$, do
            • for all $e \in Y_{\min}(M,t)$, do
              • let $M' = M + C_u \cdot e + C(\cdot,t)$,
              • if $\nexists$ a node $(M, x)$ with $M = M'$, do
                • add a new node to the graph containing
                  $(M', x')$ where $\forall i = 1, \ldots, r$,
                  $x'(T_f^i) = \begin{cases} 1 & \text{if } \mathcal{T}(M') \text{ is feasible,} \\ 0 & \text{otherwise.} \end{cases}$
                  and arc $(l, e)$ from $(M, x)$ to $(M', x')$
              • else
                • add arc $(l, e)$ from $(M, x)$ to $(M', x')$
```

```
          if it does not exist yet
   2.3. tag the node "old".
3. Remove all tags.
```

■

The algorithm constructs the BRG starting from the initial node to which it corresponds the initial marking and a binary vector defining which classes of faults may occur at M_0. Now, we consider all the labels $l \in L$ such that there exists a transition t with $L(t) = l$ for which a minimal explanation at M_0 exists. For any of these transitions we compute the marking resulting from firing t at $M_0 + C_u \cdot e$, for any $e \in Y_{\min}(M_0, t)$. If a pair (marking, binary vector) not contained in the previous nodes is obtained, a new node is added to the graph. The arc going from the initial node to the new node is labeled (l, e). The procedure is iterated until all basis markings have been considered. Note that, our approach always requires to enumerate a state space that is a strict subset of the reachability space. However, as in general for diagnosis approaches, the combinatory explosion cannot be avoided.

Example 7.3. Let us consider the PN in Figure 1, where $T_o = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7\}$, $T_u = \{\varepsilon_8, \varepsilon_9, \varepsilon_{10}, \varepsilon_{11}, \varepsilon_{12}, \varepsilon_{13}\}$, $T_f^1 = \{\varepsilon_{11}\}$ and $T_f^2 = \{\varepsilon_{12}\}$. The labeling function is defined as follows: $\mathcal{L}(t_1) = a$, $\mathcal{L}(t_2) = \mathcal{L}(t_3) = b$, $\mathcal{L}(t_4) = \mathcal{L}(t_5) = c$, $\mathcal{L}(t_6) = \mathcal{L}(t_7) = d$.

The BRG is shown in Figure 2. The notation used in in this figure is detailed in Tables 1 and 2. Each node contains a different basis marking and a binary row vector of dimension two, being two the number of fault classes. As an example, the binary vector [0 0] is associated to M_0 because $\mathcal{T}(M_0)$ is not feasible for both fault classes. From node M_0 to node M_1 there is one arc labeled a and with the null vector as minimal explanation. The node containing the basis marking M_2 has binary vector [0 1], because $\mathcal{T}(M_2)$ is feasible only for T_f^2. Node $(M_2, [0\ 1])$ has two output arcs both labeled with d and both directed to node $(M_1, [0\ 0])$ with two different minimal explanations $\vec{0}$ and e_1, respectively, plus another output arc $(b, \vec{0})$ directed to node $(M_4, [1\ 1])$. ■

The following algorithm summarizes the main steps of the on-line diagnosis carried out by looking at the BRG.

Algorithm 7.4 (Diagnosis using the BRG).

```
1. Let $w = \varepsilon$.
2. Let $\mathcal{M}(w) = \{(M_0, \vec{0})\}$.
3. Wait until a new observable transition fires.
   Let $l$ be the observed event.
4. Let $w' = w$ and $w = w'l$.
5. Let $\mathcal{M}(w) = \emptyset$,        [Computation of $\mathcal{M}(w)$]
6. For all nodes containing $M'$ : $(M', y') \in \mathcal{M}(w')$, do
   6.1. for all arcs exiting from the node with $M'$, do
      6.1.1. let $M$ be the marking of the output node
             and $e$ be the minimal e-vector on the edge
             from $M'$ to $M$,
      6.1.2. for all $y'$ such that $(M', y') \in \mathcal{M}(w')$, do
         6.1.2.1. let $y = y' + e$,
         6.1.2.2. let $\mathcal{M}(w) = \mathcal{M}(w) \cup \{(M, y)\}$,
7. for all $i = 1, \ldots, r$, do
                                [Computation of the diagnosis state]
   7.1. if $\forall\ (M, y) \in \mathcal{M}(w) \wedge \forall t_f \in T_f^i$ it is $y(t_f) = 0$, do
      7.1.1. if $\forall\ (M, y) \in \mathcal{M}(w)$ it holds $x(i) = 0$,
             where $x$ is the binary vector in node $M$, do
         7.1.1.1. let $\Delta(w, T_f^i) = 0$,
      7.1.2. else
         7.1.2.1. let $\Delta(w, T_f^i) = 1$,
   7.2. if $\exists\ (M, y) \in \mathcal{M}(w)$ and $(M', y') \in \mathcal{M}(w)$ s.t.:
         (i) $\exists t_f \in T_f^i$ such that $y(t_f) > 0$,
         (ii) $\forall t_f \in T_f^i$, $y'(t_f) = 0$, do
      7.2.1. let $\Delta(w, T_f^i) = 2$,
   7.3. if $\forall\ (M, y) \in \mathcal{M}(w)\ \exists t_f \in T_f^i : y(t_f) > 0$, do
      7.3.1. let $\Delta(w, T_f^i) = 3$.
8. Goto step 3.
```

■

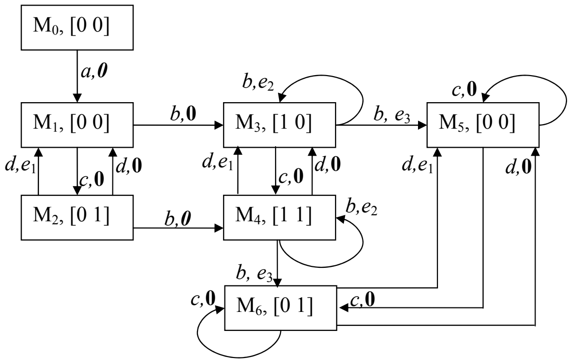


Figure 2: The BRG of the PN in Figure 1.

Steps 1 to 6 of Algorithm 7.4 enables us to compute the set $\mathcal{M}(w)$. When no event is observed, namely $w = \varepsilon$, then $\mathcal{M}(w) = \{(M_0, \vec{0})\}$. Now, assume that a label l is observed. We include in the set $\mathcal{M}(l)$ all couples (M, y) such that an arc labeled l exits from the initial node and ends in a node containing the basis marking M. The corresponding value of y is equal to the e-vector in the arc going from M_0 to M, being $\vec{0}$ the j-vector relative to M_0. In general, if w' is the actual observation, and a new event labeled l fires, we consider all couples $(M', y') \in \mathcal{M}(w')$ and all nodes that can be reached from M' with an arc labeled l. Let M be the basis marking of the generic resulting node. We include in $\mathcal{M}(w) = \mathcal{M}(w't)$ all couples (M, y), where for any M, y is equal to the sum of y' plus the e-vector labeling the arc from M' to M.

Step 7 of Algorithm 7.4 computes the diagnosis

Table 1: The markings of the BRG in Figure 2.

M_0	$[\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ ]^T$
M_1	$[\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ ]^T$
M_2	$[\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ ]^T$
M_3	$[\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ ]^T$
M_4	$[\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ ]^T$
M_5	$[\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 0\ ]^T$
M_6	$[\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ ]^T$

Table 2: The e-vectors of the BRG in Figure 2.

	ε_8	ε_9	ε_{10}	ε_{11}	ε_{12}	ε_{13}
e_1	0	0	0	0	1	1
e_2	1	1	1	0	0	0
e_3	1	0	0	1	0	0

state. Let us consider the generic ith fault class. If $\forall (M,y) \in \mathcal{M}(w)$ and $\forall t_f \in T_f^i$ it holds $y(t_f) = 0$, we have to check the ith entry of all the binary row vectors associated to the basis markings M, such that $(M,y) \in \mathcal{M}(w)$. If these entries are all equal to 0, we set $\Delta(w,T_f^i) = 0$, otherwise we set $\Delta(w,T_f^i) = 1$. On the other hand, if there exists at least one pair $(M,y) \in \mathcal{M}(w)$ with $y(t_f) > 0$ for any $t_f \in T_f^i$, and there exists at least one pair $(M',y') \in \mathcal{M}(w)$ with $y(t_f) = 0$ for all $t_f \in T_f^i$, then $\Delta(w,T_f^i) = 2$. Finally, if for all pairs $(M,y) \in \mathcal{M}(w)$ $y(t_f) > 0$ for any $t_f \in T_f^i$, then $\Delta(w,T_f^i) = 3$.

The following example shows how to perform diagnosis on-line simply looking at the BRG.

Example 7.5. Let us consider the PN in Figure 1 and its BRG in Figure 2. Let $w = \varepsilon$. By looking at the BRG we establish that $\Delta(\varepsilon,T_f^1) = \Delta(\varepsilon,T_f^2) = 0$ being both entries of the row vector associated to M_0 equal to 0.

Now, let us consider $w = ab$. In such a case $\mathcal{M}(w) = \{(M_3,\vec{0})\}$. It holds $\Delta(ab,T_f^1) = 1$ and $\Delta(ab,T_f^2) = 0$ being the row vector in the node equal to [1 0].

Finally, for $w = abbc$ it holds $\Delta(abbc,T_f^1) = 2$ and $\Delta(abbc,T_f^2) = 1$. In fact $\mathcal{M}(w) = \{(M_4,y_1),(M_5,y_2)\}$, where $y_1 = e_2$, $y_2 = e_2 + e_3$, and the row vectors associated to M_4 and M_5 are respectively [1 1] and [0 0]. ■

8 MATLAB TOOLBOX

Our group at the University of Cagliari has developed a MATLAB toolbox for PNs.

In this section we illustrate how it can be used for the diagnosis of labeled PNs. In particular, we consider the function that given a bounded labeled PN builds the basis reachability graph.

The input of the MATLAB function BRG.m are:

- the structure of the net, i.e., the matrices *Pre* and *Post*;
- the initial marking M_0;
- a cell array F that has as many rows as the number of fault classes, that contains in each row the fault transitions that belong to the corresponding fault class;
- a cell array L that has as many rows as the cardinality of the considered alphabet, that contains in each row the observable transitions having the same label;
- a cell array E that contains in each row a string of characters, each one corresponding to a different label in the considered alphabet. Obviously, the cell array E is ordered according to L.

The output of the MATLAB function BRG.m is a cell array T that univocally identifies the resulting BRG. It has as many rows as the number of nodes of the BRG. A different row is associated to each node and contains the following information:

- an identifier number of the node;
- a matrix whose rows are equal to the transpose of the basis markings associated to the node;
- a matrix with as many rows as the number of basis markings associated to the node and as many columns as the number of fault classes: the jth element in the ith row (corresponding to M_b^i) is equal to $x_i(T_f^j)$ evaluated at M_b^i. Thus, $x_i(T_f^j) = 0$ is $\mathcal{T}(M_b^i)$ is not feasible with respect to T_j^f, 1 otherwise;

- the transitions enabled at node;
- the identifier number of the nodes that are reached firing an enabled transition and the corresponding j-vector.

9 NUMERICAL SIMULATIONS

Let us consider the Petri net in Figure 3 (Lai et al., 2008), where thick transitions represent observable event and thin transitions represent unobservable events. It models a family of manufacturing systems characterized by three parameters: n, m and k.

— n is the number of production lines.

— m is the number of units of the final product that can be simultaneously produced. Each unit of product is composed of n parts.

— k is the number of operations that each part must undergo in each line.

To obtain one unit of final product n orders are sent, one to each line; this is represented by observable event t_s. Each line will produce a part (all parts are identical) and put it in its final buffer. An assembly station will take one part from each buffer (observable event t_e) to produce the final product.

The part in line i ($i = 1, \ldots, n$) undergoes a series of k operations, represented by unobservable events $\varepsilon_{i,1}, \varepsilon_{i,2}, \cdots, \varepsilon_{i,k}$.

After this series of operations two events are possible: either the part is regularly put in the final buffer of the line, or a fault may occur.

— Putting the part in the final buffer of line 1 corresponds to unobservable event $\varepsilon_{1,k+1}$, while putting the part in the final buffer of line i ($i = 2, \ldots, n$) corresponds to observable event $t_{i,k+1}$.

— There are $n-1$ faults, represented by unobservable events f_i ($i = 1, \ldots, n-1$). Fault f_i moves a part from line i to line $i+1$. Note that on line i ($i = 1, \ldots, n-1$) the fault may only occur when the part has finished processing and is ready to be put in its final buffer; the part goes to the same processing stage in line $i+1$.

In this section we present the results of the computation of the BRG for several numerical simulations. Results obtained for different values of n, k and m are summarized in Tables 3, 4 and 5.

Note that for the sake of simplicity we assumed that all faults belong to the same class.

In these tables we also detail the cardinality of the reachability set R. This is an extremely important parameter to appreciate the advantage of using basis markings. The value of $|R|$ has been computed using a function we developed in MATLAB. For completeness we also reported the time necessary to compute it.

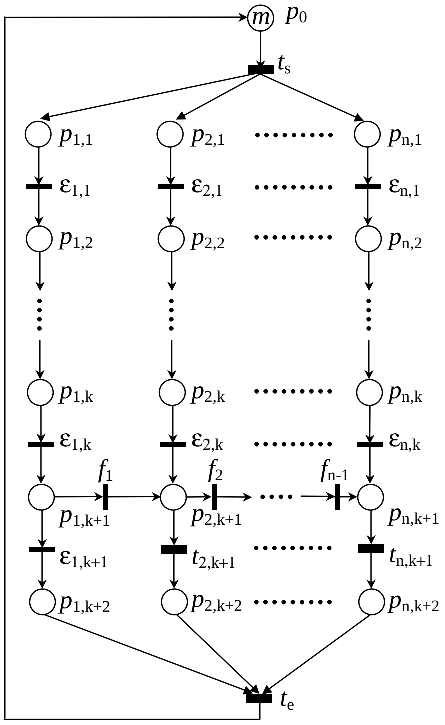


Figure 3: A manufacturing system.

Let us observe that some boxes of the above tables contain the non numerical values o:t: (out of time), that denotes that the corresponding value has not been computed within 6 hours.

All simulations have been run on a PC Athlon 64, 4000+ processor.

— Columns 1 and 2 show the values of n and k.

— Column 3 shows the number of nodes $|R|$ of the reachability graph.

— Column 4 shows the time t_R in seconds we spent to compute the reachability graph.

— Column 4 shows the number of nodes $|BRG|$ of the BRG.

— Column 5 shows the time t_{BRG} in seconds we spent to compute the BRG using the function BRG.m.

Tables 3, 4 and 5 show that the time spent to compute the reachability graph highly increases with the dimension of the net, namely with n and k, and with the number of products m.

On the contrary, the time spent to compute the BRG is always reasonable even for high values of n, k and m.

Tables 3, 4 and 5 also show that the number of nodes of the BRG only depends on n and m, while it is invariant with respect to k. On the other hand, $|R|$ also highly increases with k.

Table 3: Numerical results in the case of $m = 1$.

n	k	$\|R\|$	t_R [sec]	$\|BRG\|$	t_{BRG} [sec]
2	1	15	0.031	5	0.062
2	2	24	0.031	5	0.062
2	3	35	0.047	5	0.062
2	4	48	0.062	5	0.07
2	5	63	0.078	5	0.07
2	6	80	0.094	5	0.07
3	1	80	0.094	17	0.101
3	2	159	0.25	17	0.101
3	3	274	0.672	17	0.109
3	4	431	1.72	17	0.117
3	5	636	3.938	17	0.125
3	6	895	8.328	17	0.132
4	1	495	2.375	69	0.375
4	2	1200	16.969	69	0.43
4	3	2415	77.828	69	0.477
4	4	4320	272.53	69	0.531
4	5	7119	824.69	69	0.594
4	6	11040	2122.4	69	0.664
5	1	3295	155.81	305	4.345
5	2	9691	1615.7	305	4.765
5	3	22707	10288	305	5.25
5	4	*o.t.*	*o.t.*	305	5.75
5	5	*o.t.*	*o.t.*	305	6.897
5	6	*o.t.*	*o.t.*	305	7.894

Table 4: Numerical results in the case of $m = 2$.

n	k	$\|R\|$	t_R [sec]	$\|BRG\|$	t_{BRG} [sec]
2	1	96	0.11	17	0.086
2	2	237	0.469	17	0.094
2	3	496	2.078	17	0.1
3	1	1484	24.204	140	0.78
3	2	5949	486.39	140	0.844
3	3	18311	5320.9	140	0.906
4	1	28203	14006	1433	73.5
4	2	*o.t.*	*o.t.*	1433	76.5
4	3	*o.t.*	*o.t.*	1433	76.5

For the considered Petri net, on the basis of the above simulations, we can conclude that the diagnosis approach here presented is suitable from a computational point of view. In fact, thanks to the basis markings the reachability space can be described in a compact manner.

Table 5: Numerical results in the case of $m = 3$.

n	k	$\|R\|$	t_R [sec]	$\|BRG\|$	t_{BRG} [sec]
2	1	377	1.203	39	0.145
2	2	1293	17.203	39	0.145
3	1	12048	2113.9	553	8.219
3	2	*o.t.*	*o.t.*	553	9.016
4	1	*o.t.*	*o.t.*	9835	4095.06
4	2	*o.t.*	*o.t.*	9835	4095.06

10 CONCLUSIONS AND FUTURE WORK

This paper presents a diagnosis approach for labeled PNs using basis markings. This enables us to avoid an exhaustive enumeration of the reachability set. This approach applies to all bounded and unbounded Petri net systems whose unobservable subnet is acyclic. However, if we consider bounded net systems the most burdensome part of the procedure may be moved off-line computing the Basis Reachability Graph. Finally, we have presented a tool for the diagnosis of labeled bounded PNs and we have shown the simulation results using as diagnosis benchmark a family of manufacturing systems.

We have also studied the problem of diagnosability of bounded and unbounded PNs giving for both cases necessary and sufficient conditions for diagnosability. These results are not reported here, but they have been already submitted to an international conference.

Our future work will be that of studying the diagnosis problem for distributed systems investigating the possibility of extending the approach here presented to this case.

ACKNOWLEDGEMENTS

We thank Marco Pocci, a Master student of Electronic Engineering at the University of Cagliari, for his help in the development of the MATLAB tool for the construction of the BRG for labeled PNs.

REFERENCES

Basile, F., Chiacchio, P., and Tommasi, G. D. (2008). An efficient approach for online diagnosis of discrete event systems. *IEEE Trans. on Automatic Control*. in press.

Benveniste, A., Fabre, E., Haar, S., and Jard, C. (2003). Diagnosis of asynchronous discrete event systems: A net unfolding approach. *IEEE Trans. on Automatic Control*, 48(5):714–727.

Boel, R. and Jiroveanu, G. (2004). Distributed contextual diagnosis for very large systems. In *Proc. IFAC WODES'04: 7th Work. on Discrete Event Systems*, pages 343–348.

Boel, R. and van Schuppen, J. (2002). Decentralized failure diagnosis for discrete-event systems with costly communication between diagnosers. In *Proc. WODES'02: 6th Work. on Discrete Event Systems*, pages 175–181.

Cabasino, M., Giua, A., and Seatzu, C. (2008). Fault detection for discrete event systems using Petri nets with unobservable transitions. *Automatica*. Preliminary accepted.

Cabasino, M., Giua, A., and Seatzu, C. (2009). Diagnosis of discrete event systems using labeled Petri nets. In *Proc. 2nd IFAC Workshop on Dependable Control of Discrete Systems (Bari, Italy)*.

Chung, S. (2005). Diagnosing pn-based models with partial observable transitions. *International Journal of Computer Integrated Manufacturing*, 12 (2):158–169.

Corona, D., Giua, A., and Seatzu, C. (2004). Marking estimation of Petri nets with silent transitions. In *Proc. IEEE 43rd Int. Conf. on Decision and Control (Atlantis, The Bahamas)*.

Debouk, R., Lafortune, S., and Teneketzis, D. (2000). Coordinated decentralized protocols for failure diagnosis of discrete-event systems. *Discrete Events Dynamical Systems*, 10(1):33–86.

Dotoli, M., Fanti, M., and Mangini, A. (2008). Fault detection of discrete event systems using Petri nets and integer linear programming. In *Proc. of 17th IFAC World Congress*, Seoul, Korea.

Genc, S. and Lafortune, S. (2007). Distributed diagnosis of place-bordered Petri nets. *IEEE Trans. on Automation Science and Engineering*, 4(2):206–219.

Ghazel, M., Togueni, A., and Bigang, M. (2005). A monitoring approach for discrete events systems based on a time Petri net model. In *Proc. of 16th IFAC World Congress*, Prague, Czech Republic.

Giua, A. and Seatzu, C. (2005). Fault detection for discrete event systems using Petri nets with unobservable transitions. In *Proc. 44th IEEE Conf. on Decision and Control*, pages 6323–6328.

Hadjicostis, C. and Veghese, G. (1999). Monitoring discrete event systems using Petri net embeddings. *Lecture Notes in Computer Science*, 1639:188–207.

Jiroveanu, G. and Boel, R. (2004). Contextual analysis of Petri nets for distributed applications. In *16th Int. Symp. on Mathematical Theory of Networks and Systems (Leuven, Belgium)*.

Lai, S., Nessi, D., Cabasino, M., Giua, A., and Seatzu, C. (2008). A comparison between two diagnostic tools based on automata and Petri nets. In *Proc. IFAC WODES'08: 9th Work. on Discrete Event Systems*, pages 144–149.

Lefebvre, D. and Delherm, C. (2007). Diagnosis of DES with Petri net models. *IEEE Trans. on Automation Science and Engineering*, 4(1):114–118.

Lin, F. (1994). Diagnosability of discrete event systems and its applications. *Discrete Event Dynamic Systems*, 4(2):197–212.

Lin, F., Markee, J., , and Rado, B. (1993). Design and test of mixed signal circuits: a discrete event approach. In *Proc. 32rd IEEE Conf. on Decision and Control*, pages 246–251.

Martinez, J. and Silva, M. (1982). A simple and fast algorithm to obtain all invariants of a generalized Petri net. In *Informatik-Fachberichte 52: Application and Theory of Petri Nets.*, pages 301–310. Springer-Verlag.

Murata, T. (1989). Petri nets: properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580.

Prock, J. (1991). A new tecnique for fault detection using Petri nets. *Automatica*, 27(2):239–245.

Ruiz-Beltràn, A. R.-T. E., Rivera-Rangel, I., and Lopez-Mellado, E. (2007). Online fault diagnosis of discrete event systems. A Petri net-based approach. *IEEE Trans. on Automation Science and Engineering*, 4(1):31–39.

Sampath, M., Lafortune, S., and Teneketzis, D. (1998). Active diagnosis of discrete-event systems. *IEEE Trans. on Automatic Control*, 43(7):908–929.

Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K., and Teneketzis, D. (1995). Diagnosability of discrete-event systems. *IEEE Trans. on Automatic Control*, 40 (9):1555–1575.

Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K., and Teneketzis, D. (1996). Failure diagnosis using discrete-event models. *IEEE Trans. Control Systems Technology*, 4(2):105–124.

Sreenivas, V. and Jafari, M. (1993). Fault detection and monitoring using time Petri nets. *IEEE Trans. Systems, Man and Cybernetics*, 23(4):1155–1162.

Ushio, T., Onishi, L., and Okuda, K. (1998). Fault detection based on Petri net models with faulty behaviors. In *Proc. SMC'98: IEEE Int. Conf. on Systems, Man, and Cybernetics (San Diego, CA, USA)*, pages 113–118.

Wen, Y. and Jeng, M. (2005). Diagnosability analysis based on T-invariants of Petri nets. In *Networking, Sensing and Control, 2005. Proceedings, 2005 IEEE.*, pages 371– 376.

Wen, Y., Li, C., and Jeng, M. (2005). A polynomial algorithm for checking diagnosability of Petri nets. In *Proc. SMC'05: IEEE Int. Conf. on Systems, Man, and Cybernetics*, pages 2542– 2547.

Wu, Y. and Hadjicostis, C. (2005). Algebraic approaches for fault identification in discrete-event systems. *IEEE Trans. Robotics and Automation*, 50(12):2048–2053.

Zad, S. H., Kwong, R., and Wonham, W. (2003). Fault diagnosis in discrete-event systems: framework and model reduction. *IEEE Trans. on Automatic Control*, 48(7):1199–1212.

BRIEF BIOGRAPHY

Alessandro Giua is professor of Automatic Control at the Department of Electrical and Electronic Engineering of the University of Cagliari, Italy. He received the Laurea degree in electric engineering from the University of Cagliari, Italy in 1988, and the M.S. and Ph.D. degrees in computer and systems engineering from Rensselaer Polytechnic Institute, Troy, New York, in 1990 and 1992.

His research interests include discrete event systems, hybrid systems, networked control systems, automated manufacturing, Petri nets, control of mechanical systems, failure diagnosis. He has co-authored two textbooks on Automatic Control (in Italian) and over 150 technical papers.

Dr. Giua is a member of the editorial board of the journals: Discrete Event Dynamic Systems: Theory and Applications; IEEE Trans. on Control Systems Technology; Nonlinear Analysis: Hybrid Systems. He has served in the program committee of over 60 international conferences.

He is chair for Chapter Activities of the Member Activities Board of the IEEE Control Systems Society and chair of the IFAC Technical Committee 1.3 on Discrete Event and Hybrid Systems.

MEETING THE WORLD CHALLENGES
From Philosophy to Information Technology to Applications

Peter Simon Sapaty
Institute of Mathematical Machines and Systems, National Academy of Sciences
Glushkova Ave 42, 03187 Kiev, Ukraine
sapaty@immsp.kiev.ua

Keywords: World crises, Atomism, holism, Gestalt theory, System integrity, Waves, Distributed scenario language, Networked interpretation, Unmanned systems, Task level, Behavioral level, Crisis management, Electronic warfare, Directed energy systems.

Abstract: We have been witnessing numerous world crises and disasters—from ecological to military to economic, with global world dynamics likely to be increasing this century further. The paper highlights known holistic and gestalt principles mainly used for a single brain, extending them to any distributed systems which may need high integrity and performance in reaction to unpredictable situations. A higher organizational layer is proposed enabling any distributed resources and systems to behave as an organism having global "consciousness" and pursuing global goals. This "over-operability" layer is established by implanting into key system points the same copy of a universal intelligent module, which can communicate with other such modules and interpret collectively global mission scenarios presented in a special Distributed Scenario Language. The scenarios can be injected from any module, and then self-replicate, self-modify, and self-spread throughout the system to be managed, tasking components, activating distributed resources, and establishing runtime infrastructures supporting system's integrity. Numerous existing and prospective applications are outlined and discussed, confirming paradigm's usefulness for solving hot world problems.

1 INTRODUCTION

To understand mental state of a handicapped person, problems of economy and ecology, or how to win on a battlefield, we must consider the system as a whole -- not just as a collection and interaction of parts. The situation may complicate dramatically if the system is dynamic and open, spreads over large territories, comprises unsafe or varying components, and cannot be observed in its entirety from a single point. Numerous world crises we have been witnessing at the beginning of this century, including the current economic one, may have emerged, first of all, due to our inability of seeing and managing complex systems as a whole.

To withstand the unwanted events and their consequences (ideally: predict and prevent them) we need effective worldwide integration of numerous efforts and often dissimilar and scattered resources and systems. Just establishing advanced communications between parts of the distributed systems and providing the possibility of sharing local and global information from any point, often called "interoperability", is becoming insufficient (even insecure and harmful) for solving urgent problems in dynamic environments, in real time and ahead of it.

We may need the whole distributed system to behave as an integral organism, with parts not so interoperating but rather complementing each other and representing altogether an integral whole pursuing global goals and having a sort of global awareness and consciousness. This whole should be essentially more than the sum of its parts, with the latter having sense, possibly even existence, in the context of this whole, rather than vice versa.

This paper develops further the over-operability principle researched in Sapaty, 1993, 1999, 2002, 2005 and other works (the term "over-operability" coined in Sapaty, 2002), which can establish intelligent dominant layer over distributed resources and systems, and help solve urgent world problems in a parallel, distributed, and dynamic way.

The rest of this paper compares the dominant atomistic approach in system design, implementation and management with holistic and gestalt principles, and describes a novel ideology and technology for integral solutions in distributed worlds, which can avoid many traditional management routines in solving global problems, with its numerous practical applications outlined and discussed.

2 ATOMISM, HOLISM, GESTALT

We used to exercise predominantly atomistic, parts-to-whole philosophy of the system design, comprehension and implementation, which extends even to the organization of management facilities themselves -- as a collection of interacting parts, or *agents*. (This philosophy actually being the same as a century ago.)

Originally a system or campaign idea and the functionality needed emerge in a very general form (in a single human mind or in a close collective of such minds). Then this general idea (shown symbolically in Fig. 1*a*) is partitioned into individual chunks, or "atoms", each detailed and studied further (Fig. 1*b*). This logical partitioning already causes swelling of the problem complexity (as indicated in Fig. 1*b*).

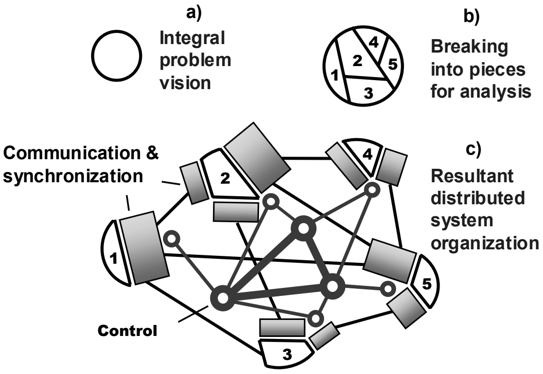

a)
Integral problem vision
b)
Breaking into pieces for analysis
1
2
3
4
5
Communication & synchronization
c)
Resultant distributed system organization
Control

Figure 1: System overhead under atomistic organization.

The next step is materialization of the defined parts and their distribution in physical or virtual space. To make these parts work or behave together within the original idea of Fig. 1*a*, we may need a good deal of their communication and synchronization, also sophisticated control infrastructures, as depicted in Fig. 1*c*. This overhead may be considerable, outweighing and shadowing the original project definition.

The main problem is that the initial idea (Fig. 1*a*) and even its second stage (Fig. 1*b*) are usually non formalized, remaining in the minds of creators only, and the *real system description and implementation start from the already partitioned-interlinked stage, with its huge overhead* (as Fig. 1*c*).

This parts-to-whole approach also dominates in the controversial "society of mind" theory (Minsky, 1988), which is trying to explain even human thinking from the atomistic positions.

Holism (see, for example, Smuts, 2007) has quite an opposite vision of systems:

- Holism as an idea or philosophical concept is diametrically opposed to atomism.
- Where the atomist believes that any whole can be broken down or analyzed into its separate parts and the relationships between them, the holist maintains that the whole is primary and often greater than the sum of its parts.
- The atomist divides things up in order to know them better; the holist looks at things or systems in aggregate.

Gestalt theory (Koffka, 1913; Wertheimer, 1922) is based on the holistic principles too:

- For the gestaltists, "Gestalten" are not the sums of aggregated contents erected subjectively upon primarily given pieces.
- Instead, we are dealing with wholes and whole–processes possessed of inner intrinsic laws.
- *Elements* are determined as parts by the intrinsic conditions of their wholes and are to be understood *as parts* relative to such wholes."

Although gestalt psychology and theory was a general approach, most of the work on gestalt was done in the area of perception. *In our research, we are trying to use the holistic and gestalt principles for the organization of distributed systems with highest possible integrity and performance* (see Sapaty, 2009).

3 WAVES, FIELDS, SCENARIOS

We describe here a novel organizational philosophy and model, based on the idea of spreading *interdependent parallel waves* (as shown in Fig. 2), as an alternative to the dominant atomistic approach briefed above, also under the influence of mentioned holistic and gestalt ideas.

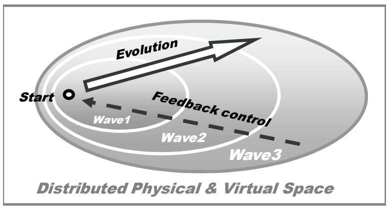


Figure 2: Grasping the entirety with spatial waves.

It allows us for an integral, parallel, and seamless navigation and coverage of virtual, physical or combined spaces where the solutions need to be found. Atomism emerges on the automatic implementation level only, which allows us to get high-level formal semantic definitions of systems and global operations in them, while omitting numerous organizational details (shown in Fig. 1*c*) and concentrating on global goals and overall performance instead.

An automatic materialization of this approach is carried out by the network of universal intelligent modules (U), embedded into important system points, which collectively interpret integral mission scenarios expressed in the waves formalism, which can start from any U, subsequently covering the distributed system at runtime.

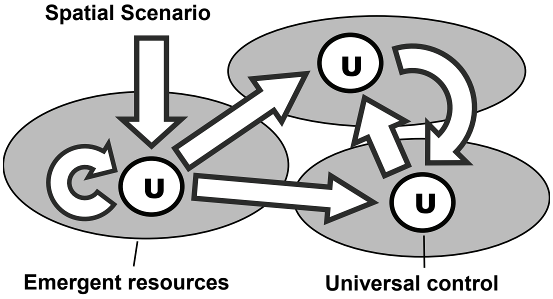


Figure 3: Self-spreading mission scenarios.

The wavelike scenarios are usually very compact and can be created and modified on the fly. They can cooperate or compete with each other in the distributed networked space as overlapping fields of parallel solutions.

Spreading waves can create knowledge infrastructures arbitrarily distributed between system components (robots, sensors, humans). These, subsequently or simultaneously navigated by same or other waves, can effectively support distributed databases, command and control, situation awareness, and autonomous decisions.

This paradigm is much in line with the existing abundant evidence that certain aspects of cognition, morals, needs, object relations, motor skills, and language acquisition proceed in developmental stages. These stages appear to be fluid, flowing, overlapping waves (Wilber, 2009), where also:

- Each stage has a holistic pattern that blends all of its elements into a structured whole;
- These patterns unfold in a relational sequence, with each senior wave transcending but including its juniors.

Our approach is also consistent with the ideas of self-actualization and person-centered approach (Rogers, 1978; Kriz, 2008), where the self is considered as an organized, consistent, conceptual gestalt exhibiting active forward thrust -- against tension reduction, equilibrium, or homeostasis (as in Freud, 2007, and others). In our case, instead of a single person we have the whole distributed system with high integrity and “active global thrust” behavior.

4 THE SCENARIO LANGUAGE

Distributed Scenario Language, or DSL (and its previous versions, WAVE including, as in Sapaty, 1999, 2005) reflects the waves model proposed, and allows us to directly express semantics of problems to be solved in distributed worlds, also the needed global system behavior in a non-atomistic manner. DSL operates with:

- *Virtual World (VW)*, which is discrete and consists of nodes and links connecting these nodes.
- Continuous *Physical World (PW)*, any point in which may be accessed by physical coordinates (taking into account certain precision).
- *Virtual-Physical World (VPW)*, which is an extension of VW where nodes additionally associate with certain coordinates in PW.

It also has the following key features:

- A DSL scenario develops as a transition between sets of progress points (or *props*) in the form of parallel *waves*.
- Starting from a prop, an action may result in one or more props (the resultant set of props may include the starting prop too).
- Each prop has a resulting *value* (which can be multiple) and a resulting *state* (being one of the four: *thru*, *done*, *fail*, and *abort*).
- Different actions may evolve independently or interdependently from the *same* prop, contributing to (and forming altogether) the resultant set of props.
- Actions may also *spatially succeed each other*, with new ones applied in parallel from all the props reached by preceding actions.
- Elementary operations can directly use local or remote values of props obtained from other actions (or even from the whole scenarios).
- Elementary operations can result either in open values that can be directly used as *operands* by other operations in an expression, or by the *next*

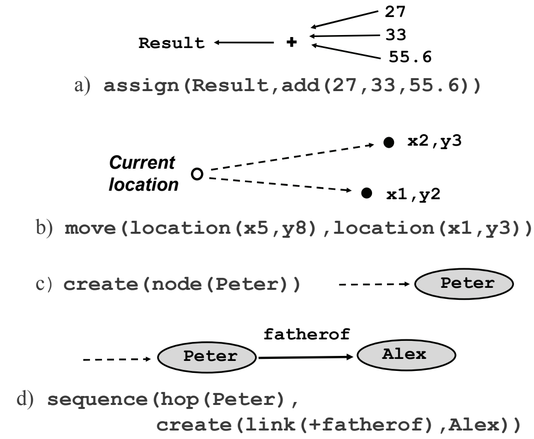
operations in a sequence. They can also be directly assigned to *local or remote variables* (for the latter case, an access to these variables may invoke scenarios of any complexity).

- Any prop can associate with a *node* in VW or a *position* in PW, or *both* -- when dealing with VPW.
- Any number of props can be simultaneously linked with the same points of the worlds.
- Staying with world points (virtual, physical, or combined) it is possible to *directly access* and update local data in them.
- Moving in physical, virtual or combined worlds, with their possible modification or even creation from scratch, are as routine operations as, say, arithmetic or logical operations of traditional programming languages.
- DSL can also be used as a usual universal programming language (like C, Java, or FORTRAN).

DSL has a recursive syntax, which on top level is as follows:

wave	→	***phenomenon* | *rule* ({ *wave* , })**
phenomenon	→	***constant* | *variable* | *special***
constant	→	***information* | *matter* | *combined***
variable	→	***heritable* | *frontal* | *environmental* | *nodal***
rule	→	***movement* | *creation* | *elimination* | *echoing* | *fusion* | *verification* | *assignment* | *advancing* | *branching* | *transference* | *timing* | *granting***

Elementary programming examples in DSL are shown in Fig. 4 for: a) assignment of a sum of values to a variable; b) parallel movement into two physical locations; c) creation of a node in a virtual space, and d) extension of the latter with a new link and node.



a) `assign(Result,add(27,33,55.6))`

b) `move(location(x5,y8),location(x1,y3))`

c) `create(node(Peter))`

d)
```
sequence(hop(Peter),
         create(link(+fatherof),Alex))
```

Figure 4: Elementary examples in DSL.

Traditional abbreviations of operations and delimiters can also be used, as in many further examples throughout this text, to simplify and shorten DSL programs, remaining however within the general recursive syntactic structure shown above.

5 COMPOSITION OF WAVES

The language allows for an integral parallel navigation of distributed worlds in a controlled *depth and breadth mode*, with any combinations of the two. We will highlight here key possibilities of doing this by composition of DSL scenarios, or waves.

5.1 Single Wave Features

Single wave (let it be *W1*) development features are shown in Fig. 5. Starting from a prop, which may be associated with a point in the world, the related scenario evolves, grasps, and covers certain region in it, performing any operations needed in the distributed space.

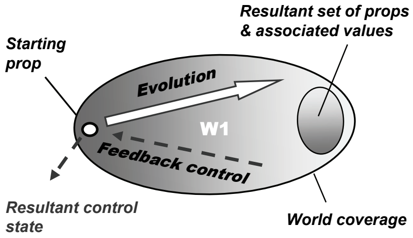
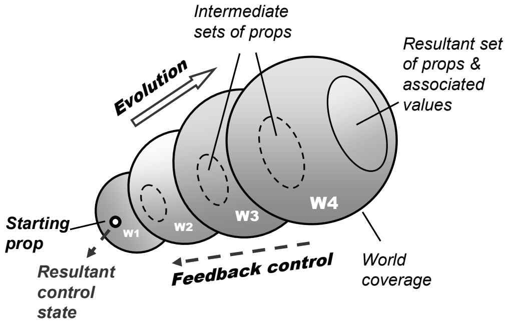


Figure 5: Single wave features.

The result of this spatial evolution may be multiple, and may lie in a (final) sub-region of the region covered, being represented by a set of resultant props (each linked to world points) and associated with them values. After termination of the wave, its resultant control state (which, in a parallel feedback process, merges termination states throughout the region covered) is available in the starting prop, and may be taken into account for decisions at higher levels. Also, if requested from higher levels, the values associated with the resultant props (which may be remote) can be lifted, spatially raked, and returned to the starting prop for a further processing.

5.2 Advancing in Space

The depth mode development of waves is shown in Fig. 6. For this type of composition, each subsequent wave is applied in parallel from all props in space reached by the previous wave, with the resultant set of props (and associated values) on the whole group being the one of the last applied wave (i.e. *W4* in the figure).



advance(*W1*,*W2*,*W3*,*W4*)

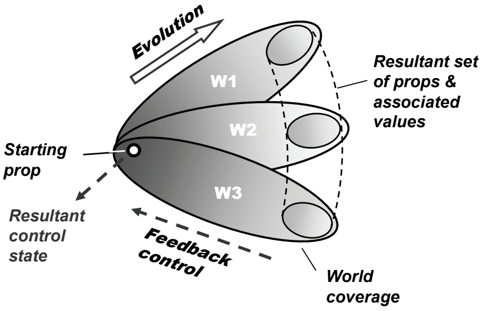
Figure 6: Depth mode composition of waves.

This spatial advancement of waves returns the resultant control state which is available at the starting prop, and the values of the resultant set of props can also be echoed to the starting prop if requested. Examples of other advancing rules:

- advance synchronized – the one where any new wave is applied only after all invocations of the previous wave have been terminated;
- repeat – where the same wave is applied repeatedly from all props reached by its previous invocation;
- repeat synchronized – where in the repeated invocation of a wave each new invocation starts only after full completion of the previous one.

5.3 Branching in Space

The branching breadth mode composition of waves is shown in Fig. 7, where all waves in the group are evolving from the same starting prop, and each wave, with its own resultant set of props and associated values, contributes to the final result.



sequence(*W1*,*W2*,*W3*)

Figure 7: Breadth mode composition of waves.

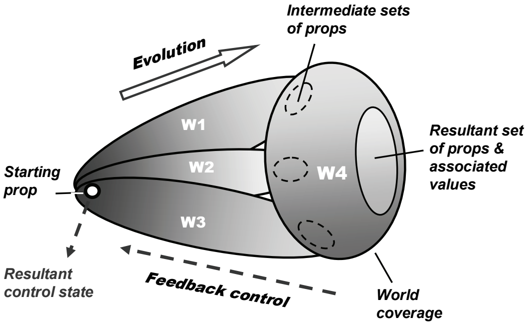
The merge of results from different waves depends on the branching rule used, with their repertoire (besides the sequence in Fig. 7) including:

```
if, while, parallel, or, parallel or,
and, parallel and,  cycle, loop, and
sling.
```

(More details on these and other rules can be found, say, from Sapaty, 1999, 2005.)

5.4 Combined Branching-Advancing

Any combination of advancing and branching modes in a distributed space can be expressed and implemented in DSL (as shown in Fig. 8).



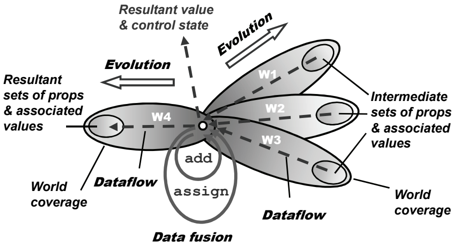
advance(sequence(*W1*,*W2*,*W3*),*W4*)

Figure 8: Breadth–depth composition mode.

These combinations, when embraced by the existing variety of composition rules, can provide any imaginable and even so far unimaginable spatial algorithms that can solve distributed problems in highly integral and compact ways, without explicit descending to the traditional atomistic level shifted to the automatic implementation only.

5.5 Operations on Remote Values

Due to fully recursive organization of DSL, it is possible to program in it arbitrary complex expressions directly operating not only on local but also arbitrarily remote values, where any programs (scenarios) can happen to be operands of any operations (expressed by rules). This gives an enormous expressive power and compactness to complex spatial scenarios evolving in distributed environments. An example of such compact expression of spatial operations on remote values and variables is shown in Fig. 9.



`assign(W4,add(W1,W2,W3))`

Figure 9: Direct operations on remote values.

6 DISTRIBUTED INTERPRETER

DSL interpreter, as from the previous language version called WAVE (Sapaty, 1993, 1999, 2005), has been prototyped in different countries on various platforms. Its public domain version (financed in the past by Siemens/Nixdorf) is being used for applications like intelligent network management or simulation of distributed dynamic systems. The DSL interpreter basics include:

- It consists of a *number of specialized modules* working in parallel and handling and sharing specific data structures, which are supporting persistent virtual worlds and temporary hierarchical control mechanisms.
- The whole network of the interpreters can be *mobile and open*, changing at runtime the number of nodes and communication structure between them.
- The heart of the distributed interpreter is its *spatial track system* enabling hierarchical command and control and remote data and code access, with high integrity of emerging parallel and distributed solutions.

The DSL interpreter structure is shown in Fig. 10.

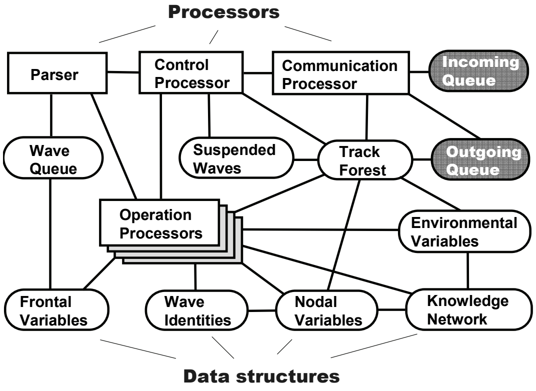


Figure 10: Structure of DSL interpreter.

It can be easily implemented in both software and hardware on any platforms, where the intelligent “wave chip” can be implanted into a great variety of devices, making them working together as an integral unit under the spatial DSL scenarios.

7 PROGRAMMING EXAMPLES

We will show here examples of solution in DSL of some important problems on networks and graphs in a fully distributed way, where each node may reside in a separate computer.

7.1 Shortest Paths

The solution for finding a path between two nodes by navigating the network with parallel waves is sown in Fig. 11, and the scenario that follows.

```
sequence(
 (direct # a; Ndist = 0; repeat(
  any #; Fdist += LINK;
  Ndist == nil, Ndist > Fdist;
  Ndist = Fdist; Npred = BACK))
 (direct # e; repeat(
  Fpath &= CONT; any # Npred);
  USER = Fpath))
```

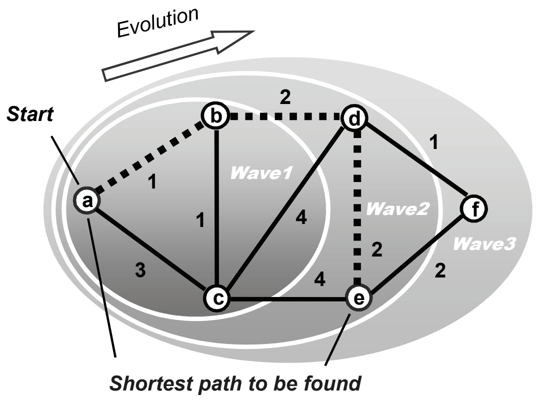


Figure 11: Finding shortest path with waves.

Many problems of optimization and control may be expressed as finding shortest paths in a distributed solution space.

7.2 Spatial Topology Analysis

DSL allows us to directly analyze and process distributed topologies in a parallel and extremely concise way.

7.2.1 Articulation Points

To find the weakest nodes in a network (called *articulation points*) which, when removed, split it into disjoint parts, as in Fig. 12 for node d, we need only the program that follows.

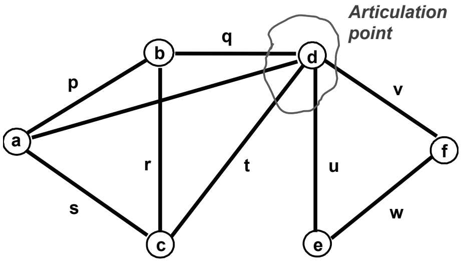


Figure 12: Articulation points.

```
direct # all; ID = CONT; Nm = 1;
and((random(all #);
 repeat(Nm ==; Nm = 1; all #)),
 (all #; Nm ==), USER = CONT)
```

Result: d.

7.2.2 Cliques

Cliques (or fully connected sub-graphs of a graph, as in Fig. 13), on the opposite, may be considered as strongest parts of a system. They can be found in parallel by the program that follows.

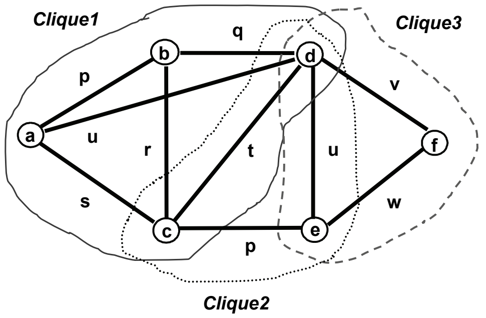


Figure 13: Cliques.

```
direct # all; Fclique = CONT;
repeat(all #; CONT !~ Fclique;
 and(andpar(any # Fclique; done !),
  or((BACK > CONT; done !),
   Fclique &= CONT))); USER = Fclique
```

Result: (a,b,c,d), (c,d,e), (d,e,f)

7.2.3 All Triangles

Any topological patterns can be found in any distributed network. For example, finding all triangles in a graph in Fig. 13 needs a simple code:

```
direct # all; Ftr = CONT;
2(all#; BACK > CONT; Ftr &= CONT);
 any # Ftr : 1; USER = Ftr
```

Result: (a,b,c), (b,c,d), (c,d,e), (d,e,f), (a,b,d), (a,c,d)

7.2.4 Network Creation

Any network can be created in a distributed space, and in parallel mode, by a very simple code too, as follows, as for the network in Fig. 13.

```
create(direct#a; p#b; q#d; u##a,(v#f;
w#e; u##d,(p#c; s##a, r##b, t##d)))
```

Arbitrary infrastructures can be created at runtime, on the fly, which can become active by putting certain procedures into their nodes and links. Any other existing models (incl. Petri nets, neural nets, contract nets, etc.) can also be implemented in a fully distributed and parallel way in DSL. Many related examples can be found in Sapaty, 1999.

8 COLLECTIVE ROBOTICS

Installing DSL interpreter into mobile robots (ground, aerial, or underwater) may allow us to organize any group solutions of complex problems in distributed physical spaces in a concise and effective way, shifting traditional management routines to automatic level. It is possible to express tasks and behaviors on different levels, as follows.

8.1 Task Level

Heterogeneous groups of mobile robots (as in Fig. 14) can be tasked at a highest possible level, just telling what they should do together, without detailing how, and what are the duties of every unit. An example task may be formulated as follows.

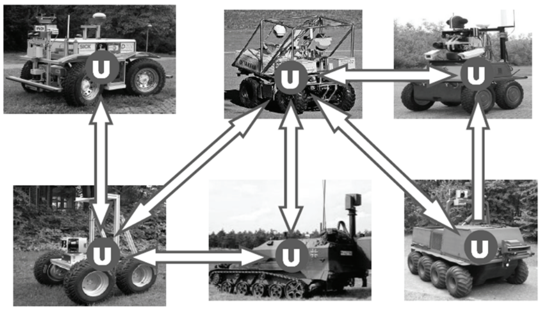


Figure 14: Grouping ground vehicles.

Go to physical locations of the disaster zone with coordinates (50.433, 30.633), (50.417, 30.490), and (50.467, 30.517). Evaluate damage in each location, find and transmit the maximum destruction value, together with exact coordinates of the corresponding location, to a management center.

The DSL program will be as follows:

```
transmit(maximum(
  move((50.433, 30.633),
       (50.417, 30.490),
       (50.467, 30.517));
  evaluate(destruction)& WHERE))
```

Details of automatic implementation of this scenario by different numbers of mobile robots are discussed in (Sapaty, 2009c).

8.2 Behavioral Level

After embedding DSL interpreters into robotic vehicles (like the aerial ones in Fig. 15), we can also provide any needed detailed collective behavior of them (at a lower than top task level, as before)—from *loose swarms* to a *strictly controlled integral unit* obeying external orders. Any mixture of different behaviors within the same scenario can be easily programmed too.

The following DSL scenario combines loose, random swarm movement in a distributed space with periodic finding/updating topologically central unit, and setting runtime hierarchical infrastructure between the units. The latter controls observation of distributed territory, collecting potential targets, distributing them between the vehicles, and then impacting potential targets by them individually. More on the implementation of this scenario can be found in Sapaty, 2008.

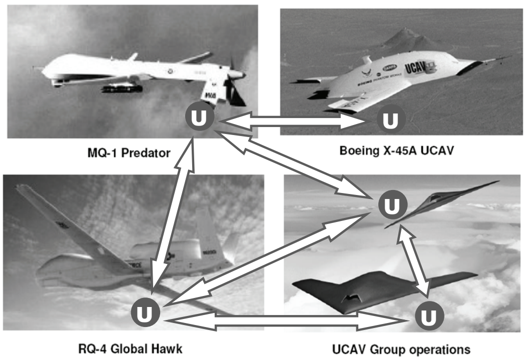


Figure 15: Grouping aerial vehicles.

```
(hop(allnodes); Range = 500;
 Limits = (dx(0,8), dy(-2,5));
 repeat(Shift = random(Limits);
  if(empty(hop(Shift, Range),
     move(Shift)))),
(repeat(hop(
  Faver =average(hop(allnodes);WHERE);
  min(hop(allnodes);
  distance(Aver, WHERE)& ADDRESS):2));
  stay(hop(nodes,all);rem(links,all);
  Frange = 20; repeat(
   linkup(+infra, firstcome, Frange));
  orpar(
   loop(nonempty(Fseen =
    repeat(free(detect(targets)),
    hoplinks(+ infra));
    repeat(
     free(select_move_shoot(Fseen),
     hoplinks(+ infra))),
   sleep(360)))
```

9 OTHER APPLICATIONS

9.1 Distributed Avionics

Distributed communicating DSL Interpreters, embedded into aircraft's key mechanisms (as in Fig. 16), can provide highest possible integrity of the aircraft that may continue to *function as a whole* even under *physical disintegration* -- which may help find critical runtime solutions saving lives and equipment (see also Sapaty, 2008a).

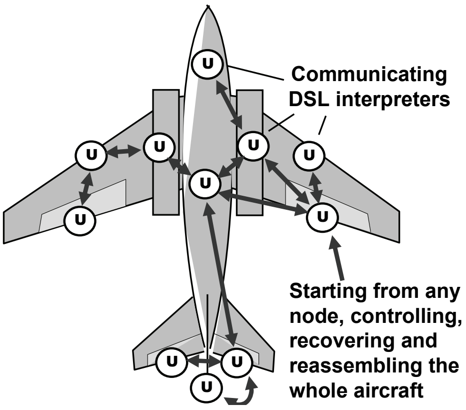

U
Communicating DSL interpreters
Starting from any node, controlling, recovering and reassembling the whole aircraft

Figure 16: Distributed control infrastructure.

Collecting availability of aircraft's basic mechanisms, and establishing overall aircraft control from any available DSL interpreter, may be organized as follows:

```
Available =
 repeat(free(belong(CONT,
  (left_aileron, right_aileron,
   left_elevator,right_elevator,
   rudder, left_engine,right_engine,
   left_chassis, right_chassis, …));
   CONT), hop(firstcome, neighbors));
if(sufficient(Available),
   control(Available), set(alarm))
```

9.2 Objects Tracking

In a large distributed space, each embedded (or moving) sensor can handle only a limited part of space, so to keep the whole observation continuous, the mobile object seen should be handed over between neighboring sensors during its movement, along with the data accumulated on it (see also Sapaty, 1999, 2007, 2008).

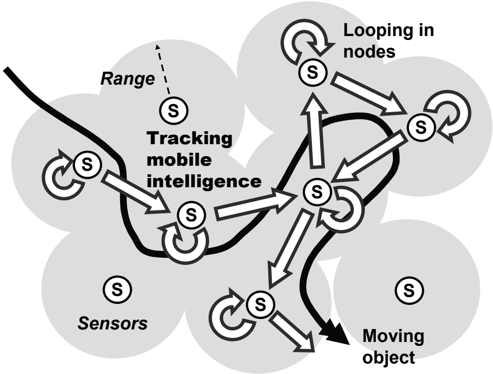


Figure 17: Tracking mobile objects.

The following program, starting in all sensors, *catches the object* it sees and *follows* it wherever it goes, if not observable from this point any more.

```
hop(allnodes); Fthr = 0.1;
Fobj = search(aerial);
visibility(Fobj) > Fthr; repeat(
 loop(visibility(Fobj) > Fthr);
 maxdest(hop(neighbors); (Seen =
 visibility(Fobj)) > Fthr; Seen))
```

9.3 Emergency Management

Embedded communicating DSL Interpreters can convert any post-disaster wreckage into a universal spatial machine capable of self-analysis and self-recovery under integral management scenarios (as in Sapaty, Sugisaka, Finkelstein, Delgado-Frias, Mirenkov, 2006; Sapaty, 2006). For example, all casualties counting program may be as follows (with its distributed operation shown in Fig. 18):

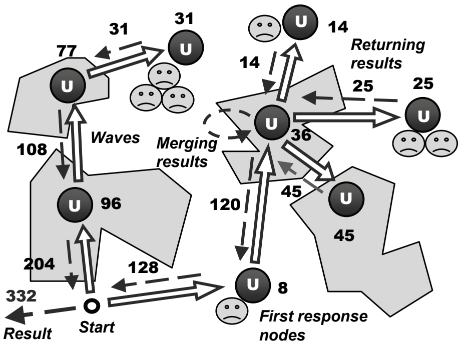


Figure 18: Counting all casualties.

```
Farea = disaster area definition;
output(sum(hop(Farea);
 repeat(free(count(casualties)),
 hop(alllinks, firstcome, Farea))))
```

Counting casualties in each region separately and organizing proportional relief delivery to each of them, may be expressed as follows:

```
Frea = disaster area definition;
split(collect(hop(Farea));
 repeat(done(count(casualties)&WHERE),
   hop(anylinks, firstcome, Farea))));
Fsupply = replicate(“package”, VAL:1);
move(VAL:2); distribute(Fsupply)
```

9.4 Directed Energy Systems

Directed energy systems and weapons are of rapidly growing importance in many areas, and especially in critical infrastructure protection, also on advanced battlefields (as shown in fig. 19). With the hardware equipment operating with the speed of light, traditional manned C2 is becoming a bottleneck for these advanced technical capabilities. With the technology offered, we may organize any runtime C2 infrastructures operating automatically, with the “speed of light” too, fitting the hardware capabilities and excluding men from the loop in time critical situations.

Figure 19: DEW in an advanced battlespace.

The following is an example of setting an automatic runtime C2 in a system with direct energy (DE) source, relay mirror (RM), and a target discovered, with an operational snapshot shown in Fig. 20.

```
sequence(
  parallel(
    (hop(DE); adjust(RM)),
    (hop(RM); adjust(DE, Target))),
  (hop(DE); activate(DE)))
```

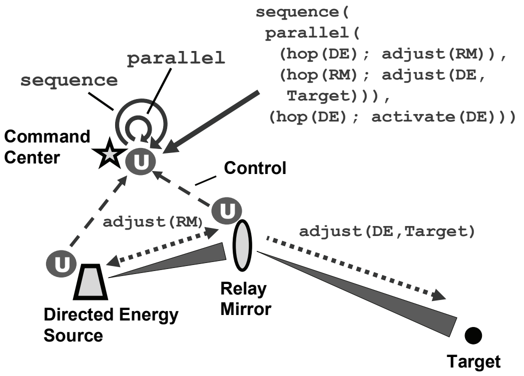


Figure 20: DE-RM-target operational snapshot.

There also exist advanced projects of global dominance with transference of directed energy, like the Boeing's Advanced Relay Mirror System (ARMS) concept. It plans to entail a constellation of as many as two dozen orbiting mirrors that would allow 24/7 coverage of every corner of the globe. When activated, this would enable a directed energy response to critical trouble spots anywhere.

We can use the distributed shortest path solution shown in section 7.1 for providing a runtime path in a worldwide distributed *dynamic* set of relay mirrors (as some of which may happen to be out of order) -- between the DE source and destination needed. This will enable optimal directed energy transfer, as shown in Fig. 21 (see also Sapaty, Morozov, Sugisaka, 2007).

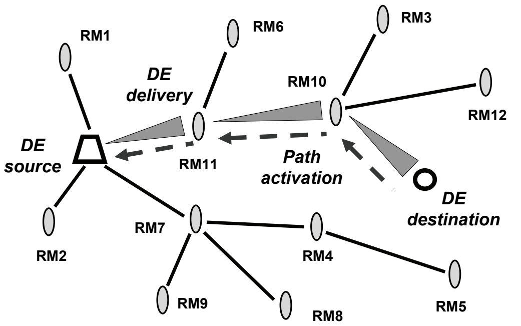


Figure 21: DE delivery via network of relay mirrors.

9.5 Electronic Warfare

Often the picture in Fig. 22 is shown as a typical example of electronic warfare. But this may rather be the last chance to survive from a missile attack. Involvement of many diverse and interlinked systems, especially for preventing and anticipating the attacks, which may be multiple and simultaneous, should be of paramount importance. All existing and being developed electronic support, attack, and protection measures have very limited scope and effect if used alone. But taken together they may provide a capability for fulfilling the objectives required. And the technology offered can readily organize this (as in Sapaty, 2007a, 2009a).

Figure 22: A Lockheed plane releasing decoy flares.

Instead of *physical flares* thrown from a plane in the final moments, we may throw, throughout the region in danger, which may be worldwide, the "*DSL scenario flares"* that can dynamically unite any available DE facilities and systems in an overwhelming electronic response to any threats.

9.6 Robotized Armies

Distributed robotized systems are of rapidly growing importance in defense (Singer, 2009, 2009a), where robotic swarming on asymmetric battlefields is becoming a major dimension of the new military doctrine for 21st century. But, as admitted by Singer, 2009, swarming, along with its simple rules of individual behavior and fully distributed nature, agility, and ubiquity, may also result in unpredictability of behavior for both sides of the conflict.

The approach briefed in this paper, also investigated in previous publications on this paradigm, is very much in line with these modern trends. Moreover, we are offering a unified solution that can harness loosely coupled swarms, always guaranteeing their global-goal-driven behavior, where the watershed between loose swarming and strict hierarchical control may be situation dependent and changing over time (as programmed in Section 8.2).

These new doctrine trends will inevitably influence the role and sense of communications on battlefields, as with the planned drastic reduction of centralized C2 much more emphasis will be paid to intelligent tactical communications, where the scenario mobility in networked systems, offered by the approach proposed, may constitute an effective solution, with the key points (as in Sapaty, 2009b):

- Dramatic shift of global organization to intelligent tactical communications;
- Self-spreading and self-recovering mission scenarios and emergent command and control;
- Embedding intelligent protocol module into existing communication equipment;
- Situation-dependent watershed between global control and local communications.

In relation to the said above, different (including new) types of commands and control strategies for distributed robotized systems were investigated in DSL (Sapaty, Morozov, Sugisaka, Finkelstein, Lambert, 2008).

10 THE FIRST COMPUTERS

The approach offered may be compared with the invention of the first world computers (Rojas, 1997) and first high-level programming languages (Zuse, 1948/49). In our case, this computer may not only operate with data stored in a localized memory, but can cover, grasp, and manage any distributed system, the whole world including, and can work not only with information but with physical matter or objects too.

If compared with the Turing computational model, instead of the head moving through tape in performing calculations, we have a recursive formula that unwraps, replicates, covers and matches the distributed world in parallel, scanning it forth and back, bringing operations and data directly to the points of their consumption, automatically setting distributed command and control infrastructures, and organizing local and global behaviors needed.

The term "computer" first referred to the people who did scientific calculations by hand (Grier, 2005). In the end, they were rewarded by a new electronic machine that took the place and the name of those who were, once, the computers.

We can draw the following symbolic parallel with this. Despite the overwhelming automation of human activity (in both information and matter processing) the world as a whole may still be considered as remaining a *human machine*, as main decisions and global management still remain the human prerogative.

With the approach offered, we can effectively automate this top-level human supervision, actually converting the whole world into a *universal programmable machine* spatially executing global scenarios in DSL or a similar language. Despite certain science fiction flavor of this comparison, we can find numerous applications for such a global approach, some mentioned above, where top level decision automation could withstand unwanted events and save lives, and where timely human reaction may not be possible, even in principle.

11 CONCLUSIONS

We have developed and tested a novel system approach, which can describe what the system should do and how to behave on a higher level, while delegating traditional management details (like partitioning into components, their distribution, interaction and synchronization) to the effective automatic layer.

A DSL scenario is not a usual program -- it is rather a recursive active spatial pattern dynamically matching structures of distributed worlds. It has a hierarchical organization, which is grasping, by means of spreading parallel waves, the whole of the system to be comprehended and impacted.

The DSL scenarios can also create, in a parallel and fully distributed way, active distributed worlds, which become persistent and operate independently. They may spatially intervene into operation of these and other worlds and systems, changing their structures and behaviors in the way required, also self-recover from indiscriminate failures and damages, as well as repair and recover the systems managed.

Prospective applications of this work can also be linked with economy, ecology and weather prediction—by using the whole networked world as a spatial supercomputer, self-optimizing its performance. Also, for terrorism and piracy fight, where the powerful parallel ability of analyzing distributed systems and finding strong and weak patterns in them, as well as any structures (as shown in Section 7.2) may be the key to global solutions.

Crises may spark anywhere and anytime like, say, birds or swine flu or the current global economic disaster. We must be ready to react on them quickly and asymmetrically, withstanding and eradicating them -- in a "pandemic" way too, highly organized and intelligent, however.

We already have technical capabilities for this, as for example, the number of mobile phone owners in the world is approaching 3bn, and installing DSL interpreter in at least a fraction of them, can allow us to organize collective runtime (and ahead of it) response to any world events.

ACKNOWLEDGEMENTS

This work has been sponsored by the Alexander von Humboldt Foundation in Germany. Special thanks to Anatoly Morozov, Vitaly Klimenko, Yuri Korovitsky, Vladimir Voloboev, Eugene Bondarenko, and Anatoly Belyaev from the National Academy of Sciences of Ukraine for years of invaluable support and productive, often hot, discussions of these ideas. Recent chats with Juergen Kriz at the 16th GTA Convention in Osnabrueck, Germany, strengthened author's admiration of the gestalt psychology and theory, influencing general orientation of this paper.

REFERENCES

Freud, S., 1997. *General Psychological Theory*. Touchstone.

Grier, D. A., 2005. When Computers Were Human, Princeton University Press.

Koffka, K., 1913. Beiträge zur Psychologie der Gestalt. *von F. Kenkel. Zeits. f.Psychol.*, 67.

Kriz, J., 2008. *Self-Actualization: Person-Centred Approach and Systems Theory*. PCCS Books.

Minsky, M., 1988. *The Society of Mind.* Simon and Schuster, New York.

Rojas, R., 1997. Konrad Zuse's Legacy: The Architecture of the Z1 and Z3. *IEEE Annals of the History of Computing*, Vol. 19, No. 2.

Rogers, C. R., 1978. *Carl Rogers on Personal Power: Inner Strength and Its Revolutionary Impact*. Trans-Atlantic Publications.

Sapaty P., 2009. Gestalt-Based Ideology and Technology for Spatial Control of Distributed Dynamic Systems, *Proc. International Gestalt Theory Congress, 16th Scientific Convention of the GTA*. University of Osnabrück, Germany.

Sapaty, P., 2009a. Distributed Capability for Battlespace Dominance. In *Electronic Warfare 2009 Conference & Exhibition*, Novotel London West Hotel & Conference Center, London.

Sapaty, P., 2009b. High-Level Communication Protocol for Dynamically Networked Battlefields. *Proc. International Conference Tactical Communications 2009 (Situational Awareness & Operational Effectiveness in the Last Tactical Mile)*. One Whitehall Place, Whitehall Suite & Reception, London, UK.

Sapaty, P. S., 2009c. Providing Spatial Integrity For Distributed Unmanned Systems". *Proc. 6th International Conference in Control, Automation and Robotics ICINCO 2009*. Milan, Italy.

Sapaty, P., 2008. Distributed Technology for Global Dominance. *Proc. of SPIE* -- Volume 6981, Defense Transformation and Net-Centric Systems 2008, Raja Suresh, Editor, 69810T.

Sapaty, P., 2008a. Grasping the Whole by Spatial Intelligence: A Higher Level for Distributed Avionics. *Proc. International Conference Military Avionics 2008*, Cafe Royal, London, UK.

Sapaty, P., Morozov, A., Finkelstein, R., Sugisaka, M., Lambert, D., 2008. A new concept of flexible organization for distributed robotized systems. *Artificial Life and Robotics*, Volume 12, Nos. 1-2/ March, Springer Japan.

Sapaty, P., 2007. Intelligent management of distributed sensor networks. In *Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense VI*, edited by Edward M. Carapezza, *Proc. of SPIE*, Vol. 6538, 653812.

Sapaty, P., 2007a. Global Management of Distributed EW-Related Systems. *Proc. International Conference Electronic Warfare: Operations & Systems*. Thistle Selfridge, London, UK.

Sapaty, P., Morozov, A., Sugisaka, M., 2007. DEW in a Network Enabled Environment. *Proc. International Conference Directed Energy Weapons 2007*. Le Meridien Piccadilly, London, UK.

Sapaty, P., 2006. Crisis Management with Distributed Processing Technology. *International Transactions on Systems Science and Applications*. Vol. 1, no. 1.

Sapaty, P., Sugisaka, M., Finkelstein, R., Delgado-Frias, J., Mirenkov, N., 2006. Advanced IT Support of Crisis Relief Missions. *Journal of Emergency Management.* Vol.4, No.4, July/August.

Sapaty, P. S., 2005. *Ruling Distributed Dynamic Worlds*. John Wiley & Sons, New York.

Sapaty, P. S., 2002. Over-Operability in Distributed Simulation and Control. *The MSIAC's M&S Journal Online*. Winter Issue, Volume 4, No. 2, Alexandria, VA, USA.

Sapaty, P. S., 1999. *Mobile Processing in Distributed and*

Sapaty, P. S., 1993. A distributed processing system, *European Patent No. 0389655*. European Patent Office.

Singer, P. W., 2009. Wired for War. Robots and Military Doctrine. *JFQ / issue 52*, 1st quarter.

Singer, P. W., 2009a. *Wired for War: The Robotics Revolution and Conflict in the 21st Century*. Penguin.

Smuts, J. C., 2007. *Holism And Evolution*. Kessinger Publishing, LLC

Wertheimer, M., 1924. *Gestalt Theory*. Erlangen, Berlin.

Wilber, K. 2009. *Ken Wilber Online: Waves, Streams, States, and Self--A Summary of My Psychological Model (Or, Outline of An Integral Psychology)*. Shambhala Publications.

Zuse, K., 1948/49. Uber den Plankalk, als Mittel zur Formulierung schematisch kombinativer Aufgaben. In *Archiv Mathematik*, Band I.

BRIEF BIOGRAPHY

Dr Peter Simon Sapaty, chief research scientist and director of distributed simulation and control project at the Institute of Mathematical Machines and Systems, National Academy of Sciences of Ukraine, is with networked systems for more than four decades. A power network engineer on education, he created citywide heterogeneous computer networks from the end of the sixties, implemented a multiprocessor macro-pipeline supercomputer in the seventies-eighties, and since then used distributed computer networks for solving complex problems of most different natures—from distributed knowledge bases to intelligent network management to road traffic control to simulation of battlefields. He also worked in Germany, UK, Canada, and Japan as Alexander von Humboldt Foundation fellow, project leader, research professor, department head, and special invited professor; created and chaired a SIG on mobile cooperative technologies within Distributed Interactive Simulation project in the US. Peter invented a higher-level distributed networking technology used in different countries and resulted in a European Patent and two John Wiley books. His interests include coordination and simulation of large distributed dynamic systems under the holistic and gestalt principles, with application in advanced command and control, cooperative robotics, infrastructure protection, crisis management, and especially for finding asymmetric solutions in unpredictable and hostile environments.

ROBOTICS AND AUTOMATION

FULL PAPERS

A PRACTICAL STEREO SYSTEM BASED ON REGULARIZATION AND TEXTURE PROJECTION

Federico Tombari [1,2] and Kurt Konolige [1]
[1] *Willow Garage Inc., Menlo Park, CA, U.S.A.*
[2] *DEIS - ARCES, University of Bologna, Italy*
{*tombari,konolige*}*@willowgarage.com, federico.tombari@unibo.it*

Keywords: Stereo vision, Spacetime stereo, Robot vision.

Abstract: In this paper we investigate the suitability of stereo vision for robot manipulation tasks, which require high-fidelity real-time 3D information in the presence of motion. We compare spatial regularization methods for stereo and spacetime stereo, the latter relying on integration of information over time as well as space. In both cases we augment the scene with textured projection, to alleviate the well-known problem of noise in low-textured areas. We also propose a new spatial regularization method, *local smoothing*, that is more efficient than current methods, and produces almost equivalent results. We show that in scenes with moving objects spatial regularization methods are more accurate than spacetime stereo, while remaining computationally simpler. Finally, we propose an extension of regularization-based algorithms to the temporal domain, so to further improve the performance of regularization methods within dynamic scenes.

1 INTRODUCTION

As part of the Personal Robot project at Willow Garage, we are interested in building a mobile robot with manipulators for ordinary household tasks such as setting or clearing a table. An important sensing technology for object recognition and manipulation is short-range (30cm – 200cm) 3D perception. Criteria for this device include:

- Good spatial and depth resolution (1/10 degree, 1 mm).
- High speed (>10 Hz).
- Ability to deal with moving objects.
- Robust to ambient lighting conditions.
- Small size, cost, and power.

Current technologies fail on at least one of these criteria. Flash ladars (Anderson et al., 2005) lack depth and, in some cases, spatial resolution, and have non-gaussian error characteristics that are difficult to deal with. Line stripe systems (Curless and Levoy, 1995) have the requisite resolution but cannot achieve 10 Hz operation, nor deal with moving objects. Structured light systems (Salvi et al., 2004) are achieving reasonable frame rates and can sometimes incorporate motion, but still rely on expensive and high-powered projection systems, while being sensitive to ambient illumination and object reflectance. Standard block-matching stereo, in which small areas are matched between left and right images (Konolige, 1997), fails on objects with low visual texture.

An interesting and early technology is the use of stereo with *unstructured* light (Nishihara, 1984). Unlike structured light systems with single cameras, stereo does not depend on the relative geometry of the light pattern – the pattern just lends texture to the scene. Hence the pattern and projector can be simplified, and standard stereo calibration techniques can be used to obtain accurate 3D measurements.

Even with projected texture, block-matching stereo still forces a tradeoff between the size of the match block (larger sizes have lower noise) and the precision of the stereo around depth changes (larger sizes “smear” depth boundaries). One possibility is to use smaller matching blocks, but reduce noise by using many frames with different projection patterns, thereby adding information at each pixel. This technique is known as *Spacetime Stereo* (STS) (Davis et al., 2005),(Zhang et al., 2003). It produces outstanding results on static scenes and under controlled illumination conditions, but moving objects create obvious difficulties (see Figure 1, bottom-left). While there have been a few attempts to deal with motion within a STS framework (Zhang et al., 2003), (Williams et al., 2005), the results are either compu-

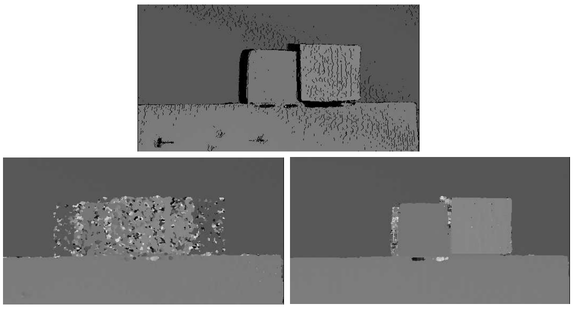


Figure 1: The top figure shows the disparity surface for a static scene; disparities were computed by integrating over 30 frames with varying projected texture using block-matching (3x3x30 block). The bottom-left figure is the same scene with motion of the center objects, integrated over 3 frames (5x5x3 block). The bottom-right figure is our local smoothing method for a single frame (5x5 block).

tationally expensive or perform poorly, especially for fast motions and depth boundaries.

In this paper, we apply regularization methods to attack the problem of motion in spacetime stereo. One contribution we propose is to enforce not only spatial, but also temporal smoothness constraints that benefit from the texture-augmented appearance of the scene. Furthermore, we propose a new regularization method, *local smoothing*, that yields an interesting efficiency-accuracy trade-off. Finally, this paper also aims at comparing STS with regularization methods, since a careful reading of the spacetime stereo literature (Davis et al., 2005; Zhang et al., 2003) shows that this has not been addressed before. Experimentally we found that, using a projected texture, regularization methods applied on single frames perform better than STS on dynamic scenes (see Figure 1) and produces interesting results also on static scenes.

In the next section we review several standard regularization methods, and introduce our novel method, *local smoothness*, which is more efficient and almost as effective. We then show how regularization can be applied across time as well as space, to help alleviate the problem of object motion in STS. In the experimental section, the considered methods are compared on static scenes and in the presence of moving objects.

2 SMOOTHNESS CONSTRAINTS IN STEREO MATCHING

Stereo matching is difficult in areas with low texture and at depth boundaries. Regularization methods add a smoothness constraint to model the regularity of surfaces in the real world. The general idea is to penalize those candidates lying at a different depth from their neighbors. A standard method is to construct a disparity map giving the probability of each disparity at each pixel, and compute a global energy function for the disparity map as a multi-class Pairwise Markov Random Field. The energy is then minimized using approximate methods such as Belief Propagation (BP) (Klaus et al., 2006), (Yang, 2006) or Graph Cuts (GC) (Kolmogorov and Zabih, 2001). Even though efficient BP-based algorithms have been proposed (Yang et al., 2006), (Felzenszwalb and Huttenlocher, 2004), overall the computational load required by global approaches does not allow real-time implementation on standard PCs.

Rather than solving the full optimization problem over the disparity map, scanline methods enforce smoothness along a line of pixels. Initial approaches based on Dynamic Programming (DP) and Scanline Optimization (SO) (Scharstein and Szeliski, 2002) use only horizontal scanlines, but suffer from streaking effects. More sophisticated approaches apply SO over multiple, variably-oriented scanlines (Hirschmuller, 2005) or use multiple horizontal and vertical passes (Kim et al., 2005), (M. Bleyer, 2008), (Gong and Yang, 2005). These methods tend to be faster than global regularization, though the use of several DP or SO passes tends to increase the computational load of the algorithms.

Another limit to the applicability of these approaches within a mobile robotic platform is their fairly high memory requirements. This section we review scanline methods and proposes a new method called *local smoothness*.

2.1 Global Scanline Methods

Let I_L, I_R be a rectified stereo image pair sized $M \cdot N$ and $W(p)$ a vector of points belonging to a squared window centered on p. The *standard* block-matching stereo algorithm computes a local cost $C(p,d)$ for each point $p \in I_L$ and each possible correspondence at disparity $d \in D$ on I_R:

$$C(p,d) = \sum_{q \in W(p)} e(I_L(q), I_R(\delta(q,d))). \tag{1}$$

where $\delta(q,d)$ is the function that offsets q in I_R according to the disparity d, and e is a (dis)similarity function. A typical dissimilarity function is the L_1 distance:

$$e(I_L(q), I_R(\delta(q,d))) = |I_L(q) - I_R(\delta(q,d))|. \tag{2}$$

In this case, the best disparity for point p is selected as:

$$d^* = \arg\min_d \{C(p,d)\}. \tag{3}$$

In the usual SO or DP-based framework, the global energy functional being minimized along a scanline S is:

$$E\left(d(\cdot)\right) = \sum_{p \in S} C\left(p, d(p)\right) + \sum_{p \in S} \sum_{q \in \mathcal{N}(p)} \rho\left(d(p), d(q)\right) \tag{4}$$

where $d(\cdot)$ denotes now a function that picks out a disparity for its pixel argument, and $q \in \mathcal{N}(p)$ are the neighbors of p according to a pre-defined criterion. Thus to minimize (4) one has to minimize two different terms, the first acting as a local evidence and the other enforcing smooth disparity variations along the scanline, resulting in a non-convex optimization problem. The smoothness term ρ is usually derived from the Potts model (Potts, 1995):

$$\rho\left(d(p), d(q)\right) = \begin{cases} 0 & d(p) = d(q) \\ \pi & d(p) \neq d(q) \end{cases} \tag{5}$$

π being a penalty term inversely proportional to the temperature of the system. Usually for stereo a Modified Potts model is deployed, which is able to handle slanted surfaces by means of an additional penalty term $\pi_s << \pi$:

$$\rho\left(d(p), d(q)\right) = \begin{cases} 0 & d(p) = d(q) \\ \pi_s & |d(p) - d(q)| = 1 \\ \pi & elsewhere \end{cases} \tag{6}$$

Thanks to (6), smooth variations of the disparity surface are permitted at the cost of the small penalty π_s. Usually in SO and DP-based approaches the set of neighbours for a point p includes only the previous point along the scanline, p_{-1}. From an algorithmic perspective, an aggregated cost $A(p,d)$ has to be computed for each $p \in S$, $d \in D$:

$$A(p,d) = C(p,d) + \min_{d'}\{A(p_{-1}, d') + \rho(d, d')\} \tag{7}$$

Because of the nature of (7) the full cost for each disparity value at the previous point p_{-1} must be stored in memory. If a single scanline is used, this typically requires $O(M \cdot D)$ memory, while if multiple passes along non-collinear scanlines are concerned, this usually requires $O(M \cdot N \cdot D)$ memory (Hirschmuller, 2005).

2.2 Local Smoothness

Keeping the full correlation surface over $M \cdot N \cdot D$ is expensive; we seek a more local algorithm that aggregates costs incrementally. In a recent paper (Zhao and Katupitiya, 2006), a penalty term is added in a local fashion to improve post-processing of the disparity image based on left-right consistency check. Here, we apply a similar penalty during the construction of the disparity map and generalize its use for multiple scanlines. Given a scanline S, we can modify (7) as follows:

$$A_{LS}(p,d) = C(p,d) + \rho(d, \tilde{d}) \tag{8}$$

where

$$\tilde{d} = \arg\min_{d}\{C(p_{-1}, d)\} \tag{9}$$

is the best disparity computed for the previous point along the scanline. Hence, each local cost is penalized if the previously computed correspondence along the scanline corresponds to a different disparity value. In this approach, there is no need to keep track of an aggregated cost array, since the aggregated cost for the current point only depends on the previously computed disparity. In practice the computation of (8) for the current disparity surface might be performed simply by subtracting π from $C(p,\tilde{d})$ and $\pi - \pi_s$ from $C(p,\tilde{d}-1)$, $C(p,\tilde{d}+1)$.

Enforcing smoothness in just one direction helps handle low-textured surfaces, but tends to be inaccurate along depth borders, especially in the presence of negative disparity jumps. Using two scans, e.g. horizontally from left to right and from right to left, helps to reduce this effect, but suffers from the well-known streaking effect (Scharstein and Szeliski, 2002). In order to enforce inter-scanline consistency, we run local smoothness over 4 scans, 2 vertical and 2 horizontal (see Figure 8). In this case, which we will refer to as *Spatial Local Smoothness* (LS_s), the aggregated cost (8) is modified as follows:

$$A_{LS_s}(d) = C(p,d) + \sum_{q \in \mathcal{N}(p)} \rho\left(d, d(q)\right). \tag{10}$$

Here $\mathcal{N}$ refers to the 4 disparities previously computed on p. The computation of d^* benefits from propagated smoothness constraints from 4 different directions, which reduces noise in low-textured surfaces, and also reduces streaking and smearing effects typical of scanline-based methods.

It is worth pointing out that the LS_s approach can be implemented very efficiently by means of a two-stage algorithm. In particular, during the first stage of the algorithm, the forward-horizontal and forward-vertical passes are computed, and the result

is stored into two $M \cdot N$ arrays. Then, during the second pass, the backward-horizontal and backward-vertical passes are processed, and within the same step the final aggregated cost (10) is also computed.

Then the best disparity is determined as in (9). Overall,

computational cost is between 3 and 4 times that of the standard local stereo algorithm. Memory requirements are also small – $O(2 \times M \times N)$.

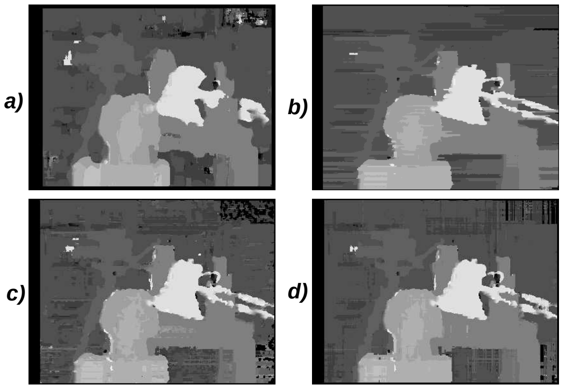


Figure 2: Qualitative comparison of different algorithms based on the smoothness constraint: a)standard b)SO-based c)local smoothness (2 horizontal scanlines) d)local smoothness (4 scanlines).

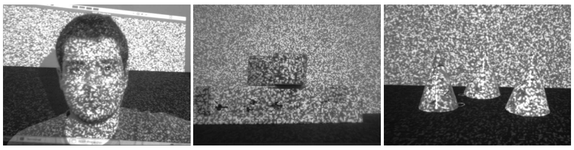


Figure 3: Dataset used for experiments: from left to right, *Face*, *Cubes*, *Cones* sequences.

2.3 Experimental Evaluation

In this section we briefly present some experimental results showing the capabilities of the previously introduced regularization methods on stereo data by comparing them to a standard block-correlation stereo algorithm. In particular, in addition to the LS_s algorithm, we consider a particularly efficient approach based only on one forward and one backward horizontal SO pass (M. Bleyer, 2008). This algorithm accounts for low memory requirements and fast performance, though it tends to suffer the streaking effect. We will refer to this algorithm as SO_s.

Fig. 2 shows some qualitative results on the *Tsukuba* dataset (Scharstein and Szeliski, 2002). The standard local algorithm is in (a), SO_s (b) and the LS_s algorithm in (d). Also, the figure shows the disparity map obtained by the use of the Local Smoothness criterion over only 2 horizontal scanlines in (c). It can be noticed that, compared to the standard approach, regularization methods allow for improved accuracy along depth borders. Furthermore, while methods based only on horizontal scanlines (b, c) present typical horizontal streaking effects, these are less noticeable in the LS_s algorithm (d). In our implementation, using standard incremental techniques but no SIMD or multi-thread optimization, time requirements on a standard PC for the standard, SO_s and LS_s algorithms are 18, 62 and 65 ms, respectively.

In addition, we show some results concerning images where a pattern is projected on the scene. As for the pattern, we use a randomly-generated grayscale chessboard, which is projected using a standard video projector. Fig. 3 shows 3 frames taken from 3 stereo sequences used here and in Section 3.4 for our experiments. Sequence *Face* is a static sequence, while *Cubes* and *Cones* are dynamic scenes where the objects present in the scene rapidly shift towards one side of the table. All frames of all sequences are 640 × 480 in resolution.

Figure 4 shows experimental results for the standard algorithm as well as SO_s and LS_s over different window sizes. Similarly to what done in (Davis et al., 2005), ground truth for this data is the disparity map obtained by the spacetime stereo technique (see next Section) over all frames of the sequence using a 5 × 5 window patch. A point in the disparity map is considered erroneous if the absolute difference between it and the groundtruth is higher than one.

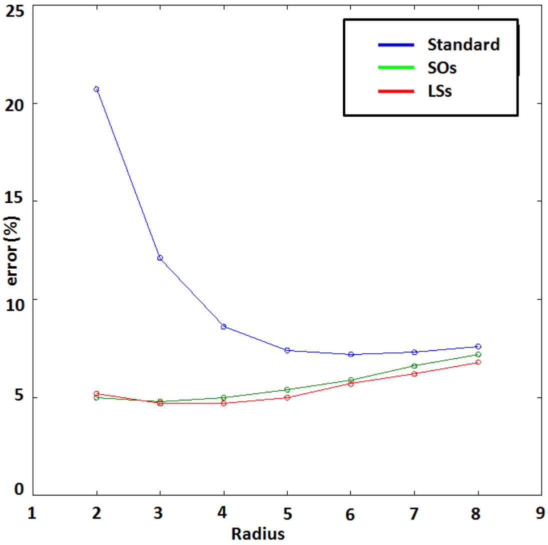


Figure 4: Quantitative comparison between different spatial approaches: standard algorithm, SO_S, LS_S.

From the figure it is clear that, even on this real dataset, regularization methods allow for improved results compared to standard methods since the curve concerning the standard algorithm is always above the other two. It is worth pointing out that both SO_s and LS_s achieve their minimum with a smaller spatial window compared to the standard algorithm, allowing for reduced smearing effect along depth borders. Conversely, the use of regularization methods with big windows increase the error rate which tends to converge to the one yielded by the standard method. It is

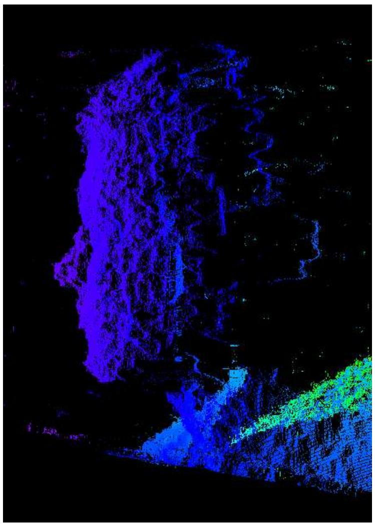


Figure 5: Point cloud showing the 3D profile of the face in Fig. 3 (left), computed using a single frame and LS_s algorithm.

also worth pointing out that overall the best result is yielded by the proposed LS_s algorithm. Finally, Figure 5 shows the 3D point cloud of the face profile obtained by using the LS_s algorithm over one frame on the *Face* dataset. From the Figure it can be noted that despite being fast and memory-efficient, this algorithm is able to obtain good accuracy in the reconstructed point cloud.

3 SPACETIME STEREO

Block-correlation stereo uses a spatial window to smooth out noise in stereo matching. A natural extension is to extend the window over time, that is, to use a spatio-temporal window to aggregate information at a pixel (Zhang et al., 2003), (Davis et al., 2005) (Figure 6). The intensity at position $I(p,t)$ is now dependent on time, and the block-matching sum over a set of frames F and a spatial window W can be written as

$$C(p,d) = \sum_{t \in F} \sum_{q \in W(p)} e(I_L(q,t), I_R(\delta(q),t)). \qquad (11)$$

Minimizing C over d yields an estimated disparity at the pixel p. Note that we obtain added information only if the scene illumination changes within F.

As pointed out in (Zhang et al., 2003), block matching in Equation (11) assumes that the disparity d is constant over both the local neighborhood W and the frames F. Assuming for the moment that the scene is static, by using a large temporal window F we can reduce the size of the window W while still reducing matching noise. This strategy has the further salutary effect of minimizing the smearing of object boundaries. Figure 1 (top) shows a typical result for spacetime block matching of a static scene with small spatial windows.

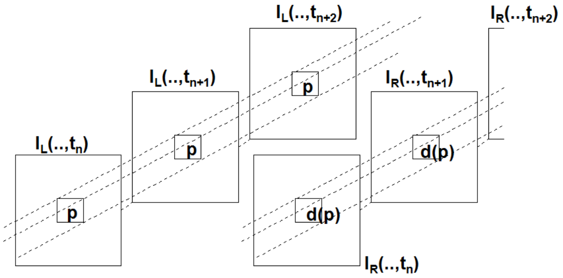


Figure 6: Spacetime window for block matching. Spatial patches centered on p are matched against corresponding patches centered on $d(p)$, and the results summed over all frames.

3.1 Moving Objects

In a scene with moving objects, the assumption of constant d over F is violated. A simple scheme to deal with motion is to trade off between spatial and temporal window size (Davis et al., 2005). In this method, a temporal window of the last k frames is kept, and when a new frame is added, the oldest frame is popped off the window, and $C(p,d)$ is calculated over the last k frames. We will refer to this approach as *sliding windows* (STS-SW). The problem is that any large image motion between frames will completely erase the effects of temporal integration, especially at object boundaries (see Figure 1, bottom-left). It is also suboptimal, since some areas of the image may be static, and would benefit from longer temporal integration.

A more complex method is to assume locally linear changes in disparity over time, that is, $d(p,t)$ is a linear function of time (Zhang et al., 2003):

$$d(p,t) \approx d(p,t_0) + \alpha(p)(t-t_0). \qquad (12)$$

For smoothly-varying temporal motion at a pixel, the linear assumption works well. Unfortunately, searching over the space of parameters $\alpha(p)$ makes minimizing the block-match sum (11) computationally difficult. Also, the linear assumption is violated at the boundaries of moving objects, where there are abrupt changes in disparity from one frame to the next (see Figure 7). These temporal boundaries present the same kind of challenges as spatial disparity boundaries in single-frame stereo.

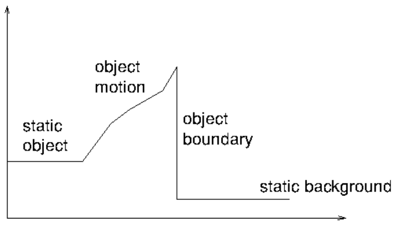


Figure 7: Disparity at a single pixel during object motion. Initially disparity is constant (no motion); then varies smoothly as the object moves past the pixel. At the object boundary there is an abrupt change of disparity.

A more sophisticated strategy would be to detect the temporal boundaries and apply temporal smoothness only up to that point. In this way, static image areas enjoy long temporal integration, while those with motion use primarily spatial information. Hence, we propose a novel method with the aim of efficiently dealing with dynamic scenes and rapidly-varying temporal boundaries. In particular, the main idea is to avoid using the spacetime stereo formulation as in (11) which blindly averages all points of the scene over time, instead enforcing a temporal smoothness constraint similarly to what is done spatially. In particular, this can be done either modelling the spatio-temporal structure with a MRF and solving using an SO or DP-based approach, or enforcing a local smoothness constraint as described in Section 2.

3.2 Temporal Regularization using SO

The idea of looking for temporal discontinuities was first discussed in (Williams et al., 2005), which proposed an MRF framework that extends over three frames. The problem with this approach is that the cost in storage and computation is prohibitive, even for just 3 frames. Here we propose a much more efficient method that consists in defining a scanline over time, analogous to the SO method over space. Given a cost array for each point and time instant $C(p,d,t)$ being computed by means of any spatial method (local, global, DP-based, $\cdots$), a SO-based approach is used for propagating forward a smoothness constraint over time:

$$A_{SO}(p,d,t) = C(p,d,t) + \min_{d'}\{A_{SO}(p,d',t-1) + \rho(d,d')\} \tag{13}$$

Instead of backtracking the minimum cost path as in the typical DP algorithm, here it is more convenient to compute the best disparity over space and time as follows:

$$d^*(p,t) = \arg\min_{d}\{A_{SO}(p,d,t)\} \tag{14}$$

so that for each new frame its respective disparity image can be readily computed. As shown in Figure 8, accumulated costs from previous frames $t_{i<n}$ are propagated forward to influence the correlation surface at time t_n. Here we propose to use as spatial algorithm the SO-based approach deploying two horizontal scanlines as discussed in Section 2. This algorithm is referred to as $SO_{s,t}$.

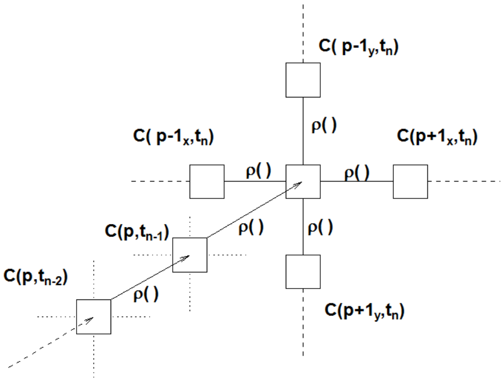


Figure 8: Local smoothing applied in the temporal domain. Disparity values influence the center pixel at time t_n from vertical and horizontal directions, and also from previous frames $t_i, i < n$.

3.3 Temporal Regularization using Local Smoothness

In a manner similar to applying SO across frames, we can instead use local smoothness. The key idea is to modify the correlation surface at position p and time t according to the best disparity found at the same point p at the previous instant $t-1$. This does not require storing and propagating a cost array, only the correspondences found at the previous time instant.

The local temporal smoothness criterion is orthogonal to the strategy adopted for solving stereo over the spatial domain, hence any local or global stereo techniques can be used together with it. Here we propose to use local spatial smoothness described in Section 2. The cost function at pixel p and time t becomes:

$$A_{LS_{s,t}}(p,d,t) = C(p,d,t) + \sum_{q\in\mathcal{N}} \rho\left(d,d(q,t)\right) + \rho\left(d,d(p,t-1)\right), \tag{15}$$

That is, the penalty terms added to the local cost are those coming from the 4 independent scanline-based processes at time t plus an additional one that depends on the best disparity computed at position p at the previous time instant (see Figure 8). This algorithm will be referred to as $LS_{s,t}$.

It is possible to propagate information both forwards and backwards in time, but there are several reasons for only going forwards. First, it keeps the

Table 1: Percentage of errors, *Cubes* stereo sequence.

Radius	STS-SW	Standard	SO_s	$SO_{s,t}$	LS_s	LSs,t
2	12.8	12.1	1.1	1.0	1.1	0.7

Table 2: Percentage of errors, *Cones* stereo sequence.

Radius	STS-SW	Standard	SO_s	$SO_{s,t}$	LS_s	LSs,t
1	46.9	49.9	5.3	5.2	14.8	12.2
3	35.4	15.9	4.2	4.1	8.2	6.9
5	31.9	9.6	4.6	4.5	7.0	6.1

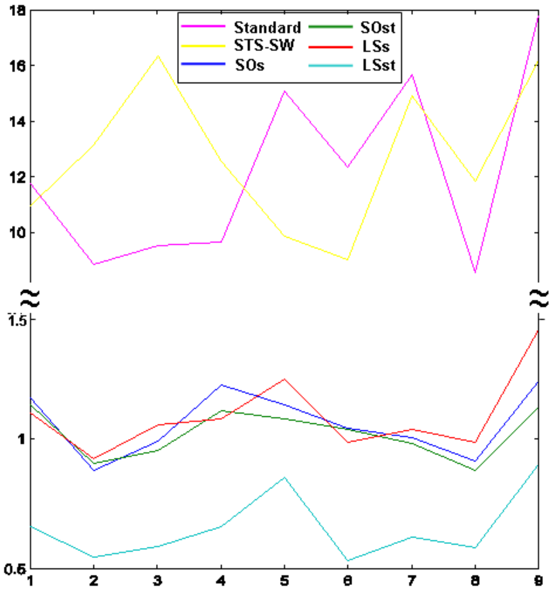


Figure 9: Comparison of error percentages between different approaches for the *Cubes* sequence at each frame of the sequence. [The graph uses two different scales for better visualization].

data current – previous frames may not be useful for a realtime system. Second, the amount of computation and storage is minimal for forward propagation. Only the previous image local costs have to be maintained, which is $O(M \cdot N)$. In contrast, to do both forwards and backwards smoothing we would need to save local costs over k frames ($O(k \cdot M \cdot N)$), and worse, recompute everything for the previous k frames, where k is the size of the temporal window for accumulation.

3.4 Experiments

This section presents experimental results over two stereo sequences with moving objects and a projected pattern, referred to as *Cubes* and *Cones* (see Fig. 3). To obtain ground truth for the stereo data, each different position of the objects is captured over 30 frames with a 3×3 spatial window, and stereo depths are averaged over time by means of spacetime stereo. Then, a sequence is built up by using only one frame for each different position of the objects.

As a comparison, we compute spacetime stereo using the sliding window approach (STS-SW). This approach is compared with regularization techniques based only on spatial smoothness (i.e. SO_s, LS_s) as well as with those enforcing temporal regularization (i.e. $SO_{s,t}$, $LS_{s,t}$).

Figure 9 shows the error rates of each algorithm for each frame of the *Cubes* dataset, with a fixed spatial window of radius 2. Table 1 reports the average error over the whole sequence. In addition, Figure 1 shows the ground truth for one frame of the sequence as well as the results obtained by $STS-SW$ and $LS_{s,t}$. As can be seen, due to the rapid shift of the objects in the scene, the approach based on spacetime stereo is unable to improve the results compared to the standard algorithm. Instead, approaches based on spatial regularization yield very low error rates, close to those obtained by the use of spacetime stereo over the same scene but with no moving objects. Furthermore, Figure 9 shows that the error variance of the methods enforcing the smoothness constraint is notably lower than that reported by the standard and STS-SW algorithms. It is worth pointing out that the use of the proposed LS regularization technique both in space and time yields the best results over all the considered frames.

As in the previous experiment, Table 2 shows the mean error percentages over the *Cones* dataset with different spatial windows (i.e., radius 1, 3, and 5). Also in this case, regularization approaches achieve notably lower error rates compared to standard and spacetime approaches. From both experiments it is possible to observe that the introduction of temporal smoothness always helps improving the performance of the considered regularization methods.

4 CONCLUSIONS AND FUTURE WORK

In this paper we investigated the capabilities of a 3D sensor comprised of a stereo camera and a texture projector. With off-the-shelf hardware and under real illumination conditions, we have shown that in the presence of moving objects single-frame stereo with regularization produces much better results than STS. Moreover, the proposed regularization approach based on local smoothness, though not based on a global optimization, shows good performance and reduced computational requirements. Finally, we have found that the proposed introduction of temporal smoothness helps improving the performance of the considered regularization methods.

We are currently actively developing a small, low-power stereo device with texture projection. There are two tasks that need to be accomplished. First, we are trying to optimize the local smoothness constraint to be real time on standard hardware, that is, to run at about 30 Hz on 640x480 images. Second, we are designing a small, fixed pattern projector that will replace the video projector. The challenge here is to project enough light while staying eye-safe and having a compact form factor. Using the methods developed in this paper, we believe we can make a truly competent realtime 3D device for near-field applications.

The code concerning the regularization methods and the STS algorithms used in this paper is open source and available online [1].

REFERENCES

Anderson, D., Herman, H., and Kelly, A. (2005). Experimental characterization of commercial flash ladar devices. In *Int. Conf. of Sensing and Technology*.

Curless, B. and Levoy, M. (1995). Better optical triangulation through spacetime analysis. In *ICCV*.

Davis, J., Nehab, D., Ramamoorthi, R., and Rusinkiewicz, S. (2005). Spacetime stereo: a unifying framework dor depth from triangulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(2).

Felzenszwalb, P. and Huttenlocher, D. (2004). Efficient belief propagation for early vision. In *Proc. CVPR*, volume 1, pages 261–268.

Gong, M. and Yang, Y. (2005). Near real-time reliable stereo matching using programmable graphics hardware. In *Proc. CVPR*, volume 1, pages 924–931.

Hirschmuller, H. (2005). Accurate and efficient stereo processing by semi-global matching and mutual information. In *Proc. CVPR*, volume 2, pages 807–814.

Kim, J., Lee, K., Choi, B., and Lee, S. (2005). A dense stereo matching using two-pass dynamic programming with generalized ground control points. In *Proc. CVPR*, pages 1075–1082.

Klaus, A., Sormann, M., and Karner, K. (2006). Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In *Proc. ICPR*, volume 3, pages 15–18.

Kolmogorov, V. and Zabih, R. (2001). Computing visual correspondence with occlusions via graph cuts. In *Proc. ICCV*, volume 2, pages 508–515.

Konolige, K. (1997). Small vision systems: hardware and implementation. In *Eighth International Symposium on Robotics Research*, pages 111–116.

M. Bleyer, M, G. (2008). Simple but effective tree structures for dynamic programming-based stereo matching. In *Proc. Int. Conf. on Computer Vision Theory and Applications (VISAPP)*, volume 2.

Nishihara, H. K. (1984). Prism: A practical real-time imaging stereo matcher. Technical report, Cambridge, MA, USA.

Potts, R. (1995). Some generalized order-disorder transitions. In *Proc. Cambridge Philosophical Society*, volume 48, pages 106–109.

Salvi, J., Pages, J., and Batlle, J. (2004). Pattern docification strategies in structured light systems. *Pattern Recognition*, 37(4).

Scharstein, D. and Szeliski, R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Computer Vision*, 47(1/2/3):7–42.

Williams, O., Isard, M., and MacCormick, J. (2005). Estimating disparity and occlusions in stereo video sequences. In *Proc. CVPR*.

Yang, Q., Wang, L., Yang, R., Wang, S., Liao, M., and Nister, D. (2006). Real-time global stereo matching using hierarchical belief propagation. In *Proc. British Machine Vision Conference*.

Yang, Q. e. a. (2006). Stereo matching with color-weighted correlation, hierachical belief propagation and occlusion handling. In *Proc. CVPR*, volume 2, pages 2347 – 2354.

Zhang, L., Curless, B., and Seitz, S. (2003). Spacetime stereo: shape recovery for dynamic scenes. In *Proc. CVPR*.

Zhao, J. and Katupitiya, J. (2006). A fast stereo vision algorithm with improved performance at object borders. In *Proc. Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 5209–5214.

[1] prdev.willowgarage.com/trac/personalrobots/browser/pkg/trunk/vision/

LMI-BASED TRAJECTORY PLANNING FOR CLOSED-LOOP CONTROL OF ROBOTIC SYSTEMS WITH VISUAL FEEDBACK

Graziano Chesi
Department of Electrical and Electronic Engineering, University of Hong Kong
Pokfulam Road, Hong Kong, China
chesi@eee.hku.hk

Keywords: Robot control, Visual feedback, Trajectory planning, LMI.

Abstract: Closed-loop robot control based on visual feedback is an important research area, with useful applications in various fields. Planning the trajectory to be followed by the robot allows one to take into account multiple constraints during the motion, such as limited field of view of the camera and limited workspace of the robot. This paper proposes a strategy for path-planning from an estimate of the point correspondences between the initial view and the desired one, and an estimate of the camera intrinsic parameters. This strategy consists of generating a parametrization of the trajectories connecting the initial location to the desired one via polynomials. The trajectory constraints are then imposed by using suitable relaxations and LMIs (linear matrix inequalities). Some examples illustrate the proposed approach.

1 INTRODUCTION

An important research area in robotics is represented by visual servoing. This area studies the application of closed-loop control in robotic system with visual feedback. Specifically, the problem consists of steering a robot end-effector from an unknown initial location to an unknown desired location by using the visual information provided by a camera. This camera is typically mounted on the robot end-effector, and the configuration is known as eye-in-hand configuration. The camera is firstly located at a certain location, called desired location, and the image projections of some object points visible from this location are recorded. Then, the camera is moved to another location of the robot workspace, from which the same object points are visible, and whose relative motion with respect to the desired location is unknown. The problem, hence, consists of reaching again the desired location from this new location, which is called initial location. See for instance (Hashimoto, 1993; Chaumette and Hutchinson, 2006; Chaumette and Hutchinson, 2007) and references therein.

The procedure just described is known as teaching-by-showing approach. It is well-known that the teaching-by-showing approach has numerous and various applications, for example in the industrial manufacture for the construction of complex components such as parts of a ship, where its function consists of allowing a robotic arm to grasp and position tools and objects. Other applications are in surveillance, where a mobile camera observes some areas of interest such as the entrance of a building in order to identify people, and in airplane alignment, where the system to be positioned is represented by the airplane that has to be aligned with respect to the runway in order to land. Also, the teaching-by-showing approach finds application in surgery, where an instrument is automatically guided to the organ to operate, in navigation, where a mobile robot has to explore a scene, and in dangerous environments such as nuclear stations and spatial missions, where humans should be replaced.

In last years, various methods have been developed for addressing this approach. Some of these methods have proposed the use of the camera pose as feedback information (known as position-based visual servoing, see e.g. (Thuilot et al., 2002)), definition of the feedback error in the image domain (known as image-based visual servoing, see e.g. (Hashimoto et al., 1991)), use of both camera pose error and image error (known as 2 $1/2$ D visual servoing, see e.g. (Malis et al., 2003)), partition of the degrees of freedoms (Corke and Hutchinson, 2001), switching strategies for ensuring constraints and improving performance (Chesi et al., 2004; Gans and Hutchinson, 2007; Lopez-Nicolas et al., 2007), generation of circular-like trajectories for minimizing the trajectory length (Chesi and Vicino, 2004), control invariant to intrinsic parameters (Malis, 2004), use of complex image features via image moments (Tahri and Chaumette, 2005), global motion plan via navigation functions (Cowan and Chang, 2005), use of cylindrical coordinate systems (Iwatsuki and Okiyama, 2005), enlargement of stability regions (Tarbouriech et al., 2005), and model-less control (Miura et al., 2006).

Path-planning strategies have also been proposed in order to take into account multiple constraints, such as limited field of view of the camera and limited workspace of the robot. See for instance (Mezouar and Chaumette, 2002; Park and Chung, 2003; Deng et al., 2005; Allotta and Fioravanti, 2005; Yao and Gupta, 2007; Kazemi et al., 2009) and references therein. These methods generally adopt potential fields along a reference trajectory in order to fulfill the required constraints, in particular the potential fields do not affect the chosen reference trajectory wherever the constraints are not violated, while they make the camera deviating from this path wherever a constraint does not hold. The planned trajectory is then followed by tracking the image projection of this trajectory through an image-based controller such as the one proposed in (Mezouar and Chaumette, 2002).

In this paper we propose the use of a parametrization of the trajectories connecting the initial location to the desired one, together with the use of dedicated optimization techniques for identifying the trajectories which satisfy the required constraints. Specifically, this parametrization is obtained by estimating the camera pose existing between these two locations and by estimating the position of the object points in the three-dimensional space. These estimations are performed by exploiting the available image point correspondences between the initial and desired views, and by exploiting the available estimate of the camera intrinsic parameters. Then, typical trajectory constraints such as the limited field of view of the camera and the limited workspace of the robot, are formulated in terms of positivity of certain polynomials. The positivity of these polynomials is then imposed by using some suitable relaxations for constrained optimization. These relaxations can be formulated in terms of LMIs (linear matrix inequalities), whose feasibility can be checked via convex programming tools. Some examples are reported to illustrate the application of the proposed approach.

This paper extends our previous works (Chesi and Hung, 2007), where a path-planning method based on the computation of the roots of polynomials was proposed (the advantage with respect to this method is the use of LMIs), and (Chesi, 2009b), where a planning strategy is derived by using homogeneous forms (the advantage with respect to this method is the use of more general relaxations which may allow one to take into account more complex constraints).

The organization of the paper is as follows. Section 2 introduces the notation, problem formulation, and some preliminaries about representation of polynomials. Section 3 describes the proposed strategy for trajectory planning. Section 4 illustrates the simulation and experimental results. Lastly, Section 5 provides some final remarks.

2 PRELIMINARIES

In this section we introduce some preliminaries, namely the notation, problem formulation, and a tool for representing polynomials.

2.1 Notation and Problem Formulation

Let us start by introducing the notation adopted throughout the paper:

- $\mathbb{R}$: real numbers space;
- 0_n: $n \times 1$ null vector;
- I_n: $n \times n$ identity matrix;
- $\|v\|$: euclidean norm of vector v.

We consider a generic stereo vision system, where two cameras are observing a common set of object points in the scene. The symbols F^{ini} and F^{des} represent the frames of the camera in the initial and desired location respectively. These frames are expressed as

$$\begin{array}{rcl} F^{ini} & = & \{R^{ini}, t^{ini}\} \\ F^{des} & = & \{R^{des}, t^{des}\} \end{array} \tag{1}$$

where $R^{ini}, R^{des} \in \mathbb{R}^{3\times 3}$ are rotation matrices, and $t^{ini}, t^{des} \in \mathbb{R}^3$ are translation vectors. These quantities R^{ini}, R^{des}, t^{ini} and t^{des} are expressed with respect to an absolute frame, which is indicated by F^{abs}.

The observed object points project on the image plane of the camera in the initial and desired location onto the image points $p_1^{ini}, \ldots, p_n^{ini} \in \mathbb{R}^3$ (initial view) and $p_1^{des}, \ldots, p_n^{des} \in \mathbb{R}^3$ (desired view). These image points are expressed in homogeneous coordinates according to

$$p_i^{ini} = \begin{pmatrix} p_{i,1}^{ini} \\ p_{i,2}^{ini} \\ 1 \end{pmatrix}, \quad p_i^{des} = \begin{pmatrix} p_{i,1}^{des} \\ p_{i,2}^{des} \\ 1 \end{pmatrix}. \tag{2}$$

where $p_{i,1}^{ini}, p_{i,1}^{des} \in \mathbb{R}$ are the components on the x-axis of the image screen, while $p_{i,2}^{ini}, p_{i,2}^{des} \in \mathbb{R}$ are those on the y-axis. The projections p_i^{ini} and p_i^{des} are determined by the projective law

$$\begin{array}{rcl} d_i^{ini} p_i^{ini} & = & K R^{ini'} \left(q_i - t^{ini}\right) \\ d_i^{des} p_i^{des} & = & K R^{des'} \left(q_i - t^{des}\right) \end{array} \quad (3)$$

where $d_i^{ini}, d_i^{des} \in \mathbb{R}$ are the depths of the ith point, $q_i \in \mathbb{R}^3$ is the ith point expressed with respect to F^{abs}, and $K \in \mathbb{R}^{3\times 3}$ is the upper triangular matrix containing the intrinsic parameters of the camera.

The problem we consider in this paper consists of planning a trajectory from the initial location F^{ini} to the desired one F^{des} (which are unknown) by using the available estimates of:

1. the image projections $\hat{p}_1^{ini}, \hat{p}_1^{des}, \ldots, \hat{p}_n^{ini}, \hat{p}_n^{des}$;
2. and intrinsic parameters matrix $\hat{K}$.

This trajectory must ensure that the object points are kept inside the field of view of the camera, and that the camera does not exit its allowed workspace.

In the sequel, we will indicate the set of rotation matrices in $\mathbb{R}^{3\times 3}$ as $SO(3)$, and the set of frames in the three-dimensional space as $SE(3)$, where $SE(3) = SO(3) \times \mathbb{R}^3$.

2.2 Representation of Polynomials

Before proceeding, let us briefly introduced a tool for representing polynomials which will be exploited in the sequel. Let $p(x)$ be a polynomial of degree $2m$ in the variable $x = (x_1, \ldots, x_n)' \in \mathbb{R}^n$, i.e.

$$p(x) = \sum_{\substack{i_1 + \ldots + i_n \leq 2m \\ i_1 \geq 0, \ldots, i_n \geq 0}} c_{i_1,\ldots,i_n} x_1^{i_1} \cdots x_n^{i_n} \quad (4)$$

for some coefficients $c_{i_1,\ldots,i_n} \in \mathbb{R}$. Then, $p(x)$ can be expressed as

$$p(x) = x^{\{m\}'} P(\alpha) x^{\{m\}} \quad (5)$$

where $x^{\{m\}}$ is any vector containing a base for the polynomials of degree m in x, and hence can be simply chosen as the set of monomials of degree less than or equal to m in x, for example via

$$x^{\{m\}} = (1, x_1, \ldots, x_n, x_1^2, x_1 x_2, \ldots, x_n^m)', \quad (6)$$

and

$$P(\alpha) = P + L(\alpha) \quad (7)$$

where $P = P'$ is a symmetric matrix such that

$$p(x) = x^{\{m\}'} P x^{\{m\}}, \quad (8)$$

while $L(\alpha)$ is a linear parametrization of the linear space

$$\mathcal{L}(n,m) = \{L = L' : \ x^{\{m\}'} L x^{\{m\}} = 0 \ \ \forall x\} \quad (9)$$

being α a vector of free parameters. The dimension of $x^{\{m\}}$ is given by

$$\sigma(n,m) = \frac{(n+m)!}{n!m!} \quad (10)$$

while the dimension of α (i.e., the dimension of $\mathcal{L}$) is

$$\tau(n,m) = \frac{1}{2}\sigma(n,m)(\sigma(n,m)+1) - \sigma(n,2m). \quad (11)$$

The representation in (5) was introduced in (Chesi et al., 1999) with the name SMR (square matricial representation). The matrices P and $P(\alpha)$ are known as SMR matrices of $p(x)$, and can be computed via simple algorithms. See also (Chesi et al., 2003; Chesi et al., 2009).

The SMR was introduced in (Chesi et al., 1999) in order to investigate positivity of polynomials via convex optimizations. Indeed, $p(x)$ is clearly positive if it is a sum of squares of polynomials, and this latter condition holds if and only if there exists α such that

$$P(\alpha) \geq 0 \quad (12)$$

which is an LMI (linear matrix inequality). It turns out that, establishing whether an LMI admits a feasible solution or not, amounts to solving a convex optimization.

3 TRAJECTORY PLANNING

This section describes the proposed approach. Specifically, we first introduce the adopted parametrization of the trajectories, then we describe the computation of the trajectory satisfying the required constraints, and lastly we explain how the camera pose and object points can be estimated from the available data.

3.1 Trajectory Parametrization

Let us start by parameterizing the trajectory of the camera from the initial location to the desired one. This can be done by denoting the frame of the camera along the trajectory as

$$F(a) = \{R(a), t(a)\} \quad (13)$$

where $a \in [0,1]$ is the normalized trajectory abscise, $R(a) \in SO(3)$ is the rotation matrix of $F(a)$, and $t(a) \in \mathbb{R}^3$ is the translation vector. We choose the convention

$$\begin{array}{lcl} a = 0 & \rightarrow & F(a) = F^{ini} \\ a = 1 & \rightarrow & F(a) = F^{des}. \end{array} \quad (14)$$

The functions $R : [0,1] \rightarrow SO(3)$ and $t : [0,1] \rightarrow \mathbb{R}^3$ must satisfy the boundary conditions

$$\begin{array}{rclrcl} R(0) & = & \hat{R}^{ini}, & R(1) & = & \hat{R}^{des} \\ t(0) & = & \hat{t}^{ini}, & t(1) & = & \hat{t}^{des} \end{array} \quad (15)$$

where $\hat{R}^{ini}$, $\hat{R}^{des}$, $\hat{t}^{ini}$ and $\hat{t}^{des}$ are the available estimates of R^{ini}, R^{des}, t^{ini} and t^{des} (the computation of these estimates will be addressed in Section 3.3). We adopt polynomials in order to parameterize $R(a)$ and $t(a)$. Specifically, we parameterize $t(a)$ according to

$$t(a) = \sum_{i=0}^{\delta} \check{t}_i a^i \tag{16}$$

where δ is an integer representing the chosen degree for $t(a)$, and $\check{t}_0, \ldots, \check{t}_\delta \in \mathbb{R}^3$ are vectors to be determined. Then, we parameterize $R(a)$ as

$$R(a) = \frac{E(r(a))}{\|r(a)\|^2} \tag{17}$$

where $E : \mathbb{R}^4 \to SO(3)$ is the parametrization of a rotation matrix via Euler parameters, which is given by

$$E(r) = \begin{pmatrix} r_1^2 - r_2^2 - r_3^2 + r_4^2 & 2\,(r_1 r_2 - r_3 r_4) & 2\,(r_1 r_3 + r_2 r_4) \\ 2\,(r_1 r_2 + r_3 r_4) & -r_1^2 + r_2^2 - r_3^2 + r_4^2 & 2\,(r_2 r_3 - r_1 r_4) \\ 2\,(r_1 r_3 - r_2 r_4) & 2\,(r_2 r_3 + r_1 r_4) & -r_1^2 - r_2^2 + r_3^2 + r_4^2 \end{pmatrix} \tag{18}$$

while $r : [0,1] \in \mathbb{R}^4$ denotes the Euler parameter along the trajectory. It turns out that

$$E(r) \in SO(3) \;\; \forall r \in \mathbb{R}^4 \setminus \{0_4\}, \tag{19}$$

and moreover

$$\forall R \in SO(3) \; \exists \xi(R) \in \mathbb{R}^4 \setminus \{0_4\} : \; E(\xi(R)) = R, \tag{20}$$

in particular

$$\xi(R) = \begin{pmatrix} \sin\dfrac{\theta}{2} u \\ \cos\dfrac{\theta}{2} \end{pmatrix} \tag{21}$$

where $\theta \in [0, \pi]$ and $u \in \mathbb{R}^3$, $\|u\| = 1$, are respectively the rotation angle and axis in the exponential coordinates of R, i.e.

$$R = e^{[\theta u]_\times}. \tag{22}$$

We parameterize $r(a)$ according to

$$r(a) = \sum_{i=0}^{\gamma} \check{r}_i a^i \tag{23}$$

where $\check{r}_0, \ldots, \check{r}_\gamma \in \mathbb{R}^4$ are vectors for some integer γ. The boundary conditions in (15) become, hence,

$$\begin{array}{rclcrcl} \check{r}_0 & = & \xi(\hat{R}^{ini}), & & \sum_{i=0}^{\gamma} \check{r}_i & = & \xi(\hat{R}^{des}) \\ \check{t}_0 & = & \hat{t}^{ini}, & & \sum_{i=0}^{\delta} \check{t}_i & = & \hat{t}^{des} \end{array} \tag{24}$$

which imply that $r(a)$ and $t(a)$ can be re-parameterized as

$$\begin{array}{rcl} r(a) & = & \left(\xi(\hat{R}^{des}) - \xi(\hat{R}^{ini}) - \sum_{i=1}^{\gamma-1} \bar{r}_i\right) a^\gamma \\ & & + \sum_{i=1}^{\gamma-1} \bar{r}_i a^i + \xi(\hat{R}^{ini}) \\ t(a) & = & \left(\hat{t}^{des} - \hat{t}^{ini} - \sum_{i=1}^{\delta-1} \bar{t}_i\right) a^\delta + \sum_{i=1}^{\delta-1} \bar{t}_i a^i + \hat{t}^{ini} \end{array} \tag{25}$$

where $\bar{r}_1, \ldots, \bar{r}_{\gamma-1} \in \mathbb{R}^4$ and $\bar{t}_1, \ldots, \bar{t}_{\delta-1} \in \mathbb{R}^3$ are free vectors.

Let us observe that the derived parametrization can describe arbitrarily complicated trajectories, simply by selecting sufficiently large degrees γ and δ. Moreover, it is useful to observe that special cases such as straight lines are simply recovered by the choices

$$\begin{array}{c} \gamma = 1 \text{ (straight line in the domain of } E) \\ \delta = 1 \text{ (straight line in the translational space).} \end{array} \tag{26}$$

For ease of description we will assume $\gamma = 1$ in the following sections.

3.2 Trajectory Computation

In this section we address the problem of identifying which trajectories inside the introduced parametrization satisfy the required trajectory constraints. Due to space limitation, we describe only two fundamental constraints, in particular the visibility constraint (the object points must remain in the field of view of the camera) and the workspace constraint (the camera cannot exit from its allowed workspace). Other constraints can be similarly considered.

Let us indicate with $p_i(a) = (p_{i,1}(a), p_{i,2}(a), 1)' \in \mathbb{R}^3$ the image projection of the ith object point along the trajectory. The visibility constraint is fulfilled whenever

$$p_{i,j}(a) \in (s_{i,1}, s_{i,2}) \;\; \forall i = 1, \ldots, n \; \forall j = 1, 2 \; \forall a \in [0,1] \tag{27}$$

where $s_{1,1}, s_{1,2}, s_{2,1}, s_{2,2} \in \mathbb{R}$ are the screen limits. We estimate $p_i(a)$ via

$$\begin{array}{rcl} p_i(a) & = & \dfrac{f_i(a)}{f_{i,3}(a)} + (1-a)\left(\hat{p}_i^{ini} - \dfrac{f_i(0)}{f_{i,3}(0)}\right) \\ & & + a\left(\hat{p}_i^{des} - \dfrac{f_i(1)}{f_{i,3}(1)}\right) \end{array} \tag{28}$$

where $f_i(a) = (f_{i,1}(a), f_{i,2}(a), f_{i,3}(a))' \in \mathbb{R}^3$ is

$$f_i(a) = \hat{K} E(r(a))' (\hat{q}_i - t(a)) \tag{29}$$

and $\hat{q}_i \in \mathbb{R}^3$ is the estimate of the object point q_i (the computation of this estimate will be addressed in Section 3.3). Let us observe that this choice ensures

$p_i(0) = \hat{p}_i^{ini}$ and $p_i(1) = \hat{p}_i^{des}$. We can rewrite $p_i(a)$ as

$$p_i(a) = \frac{1}{f_{i,3}(a)} \begin{pmatrix} g_{i,1}(a) \\ g_{i,2}(a) \\ f_{i,3}(a) \end{pmatrix} \tag{30}$$

where $g_{i,1}(a), g_{i,2}(a) \in \mathbb{R}$ are polynomials.

Then, let us consider the workspace constraint. A possible way to define the workspace constraint is via inequalities such as

$$d_i'(t(a_i) - o_i) > w_i \ \ \forall i = 1, \ldots, n_w \tag{31}$$

where $d_i \in \mathbb{R}^3$ is the direction along which the constraint is imposed, $a_i \in [0,1]$ specifies where the constraint is imposed on the trajectory, $o_i \in \mathbb{R}^3$ locates the constraint, $w_i \in \mathbb{R}$ specifies the minimum distance allowed from the point o_i along the direction d_i, and n_w is the number of constraints.

Hence, let us define the set of polynomials

$$\begin{aligned} \mathcal{H} \ = \ & \{ s_{j,k} f_{i,3}(a) - (-1)^k g_{i,j}(a) \ \forall i = 1, \ldots, n, \\ & j,k = 1,2 \} \ \cup \ \{ f_{i,3}(a) \ \forall i = 1, \ldots, n \} \\ & \cup \{ d_i'(t(a_i) - o_i) - w_i \ \forall i = 1, \ldots, n_w \} . \end{aligned} \tag{32}$$

The visibility and workspace constraints are hence fulfilled whenever

$$h(a) > 0 \ \ \forall h(a) \in \mathcal{H} \ \forall a \in [0,1]. \tag{33}$$

For each polynomial $h(a)$ in $\mathcal{H}$, let us introduce an auxiliary polynomial $u_h(a)$ of some degree, and let us define

$$v_h(a) = h(a) - a(1-a)u_h(a). \tag{34}$$

Let us express these polynomials via the SMR as

$$\begin{aligned} u_h(a) &= y_h(a)' U_h y_h(a) \\ v_h(a) &= z_h(a)' V_h(\alpha_h) z_h(a) \end{aligned} \tag{35}$$

where $y_h(a), z_h(a)$ are vectors containing polynomial bases, and $U, V(\alpha_h)$ are symmetric SMR matrices (see Section 2.2 for details). It can be verified that (33) holds whenever the following set of LMIs is satisfied:

$$\left. \begin{aligned} U_h &> 0 \\ V_h(\alpha_h) &> 0 \end{aligned} \right\} \ \forall h(a) \in \mathcal{H}. \tag{36}$$

The LMI feasibility test (36) provides a sufficient condition for the existence of a trajectory satisfying the required constraints. Hence, it can happen that this condition is not satisfied even if a trajectory does exist. However, it should be observed that the conservatism of this condition decreases by increasing the degree of the polynomials used to parameterize the trajectory.

3.3 Camera Pose and Scene Estimation

In the previous sections we have described how the trajectory of the camera can be parameterized and computed. In particular, the parametrization was based on the estimates $\hat{R}^{ini}$, $\hat{R}^{des}$, $\hat{t}^{ini}$ and $\hat{t}^{des}$ of the components of the initial and desired frames F^{ini} and F^{des}, while the computation was based on the estimates $\hat{q}_1, \ldots, \hat{q}_n$ of the object points $q_1, \ldots, q_n$. Here we describe some ways to obtain these estimates.

Given the estimates $\hat{p}_1^{ini}, \hat{p}_1^{des}, \ldots, \hat{p}_n^{ini}, \hat{p}_n^{des}$ of the image projections and $\hat{K}$ of the intrinsic parameters matrix, one can estimate the camera pose between F^{ini} and F^{des}, and hence R^{ini} and t^{ini} since F^{des} can be chosen without loss of generality equal to F^{abs}. This estimation can be done, for example, through the essential matrix or through the homography matrix, see for instance (Malis and Chaumette, 2000; Chesi and Hashimoto, 2004; Chesi, 2009a) and references therein.

Once that the estimates $\hat{R}^{ini}$ and $\hat{t}^{ini}$ have been found, one can compute the estimates $\hat{q}_1, \ldots, \hat{q}_n$ of the object points via a standard triangulation scheme, which amounts to solving a linear least-squares problem.

Let us observe that, if no additional information is available, the translation vector and the object points can be estimated only up to a scale factor. In this case, the workspace constraint has to be imposed in a normalized space. This problem does not exist if a CAD model of the object (or part of it) is available, since this allows to estimate the distance between the origins of F^{ini} and F^{des}.

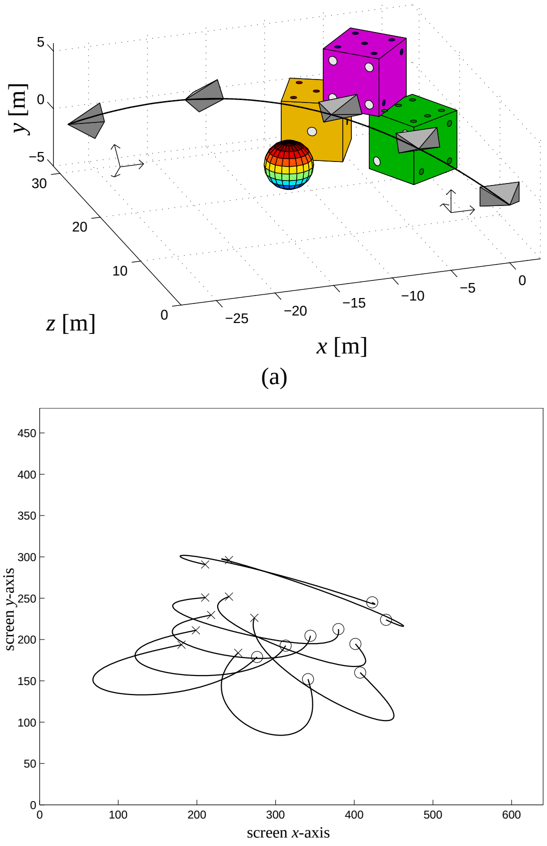
4 ILLUSTRATIVE EXAMPLES

In this section we present some illustrative examples of the proposed approach. Let us consider the situation shown in Figure 1a, where a camera observes some object points (the centers of the nine large dots in the “2”, “3” and ”4” faces of the three dices) from the initial and desired locations (leftmost and rightmost cameras respectively). Figure 1b shows the image projections of these points in the initial view (“o” marks) and desired view (“x” marks). The intrinsic parameters matrix is chosen as

$$K = \begin{pmatrix} 400 & 0 & 320 \\ 0 & 300 & 240 \\ 0 & 0 & 1 \end{pmatrix}. \tag{37}$$

The problem consists of planning a trajectory from the initial location to the desired one which ensures that the object points are kept inside the field of view of the camera and the camera does not collide with the sphere interposed between F^{ini} and F^{des} (which represents an obstacle to avoid).

Let us use the proposed approach. We parameterize the trajectory as described in Section 3.1 with polynomials of degree two by estimating the camera pose between F^{ini} and F^{des} via the essential matrix. Then, we build the set of polynomials $\mathcal{H}$, which impose the visibility and workspace constraints. The workspace constraint is chosen by requiring that the trajectory must remain at a certain distance from the obstacle in two directions. Hence, we compute the SMR matrices U_h and $V_h(\alpha_h)$ in (35), and by using the LMI toolbox of Matlab we find that the LMIs in (36) are feasible, in particular the obtained feasible trajectory is shown in Figures 1a and 1b.



(a)

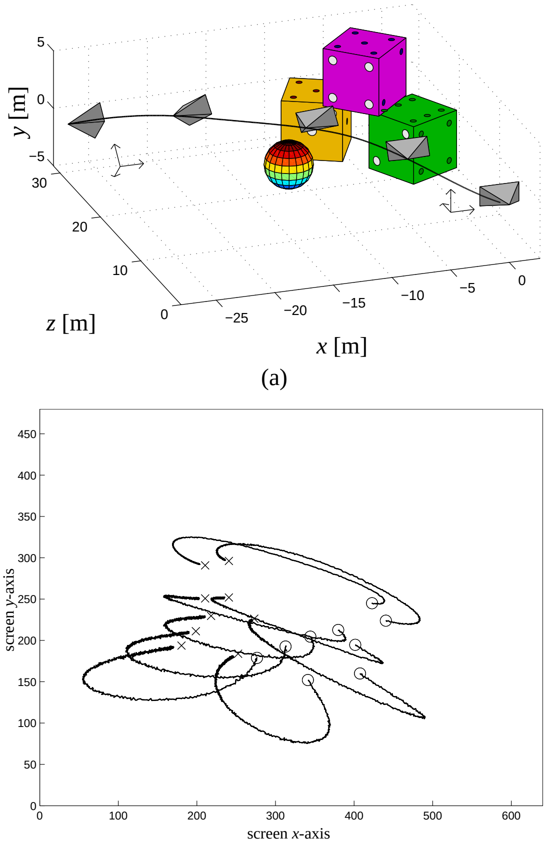
(b)

Figure 1: (a) Initial frame F^{ini} (leftmost camera), desire frame F^{des} (rightmost camera), object points (centers of the nine large dots in the "2", "3" and "4" faces of the three dices), planned trajectory (solid line), and some intermediate locations of the camera along the planned trajectory. (b) Image projections of the object points in the initial view ("o" marks) and desired view ("x" marks), and image projection of the planned trajectory (solid line).

Now, in order to introduce typical uncertainties of real experiments, we corrupt the image projections of the object points by adding image noise with uniform distribution in $[-1,1]$ pixels to each component. Moreover, we suppose that the camera is coarsely calibrated, in particular the available estimate of the intrinsic parameters matrix is

$$\hat{K} = \begin{pmatrix} 430 & 0 & 338 \\ 0 & 275 & 250 \\ 0 & 0 & 1 \end{pmatrix}. \qquad (38)$$

We repeat the previous steps in the presence of these uncertainties, and then we track the planned trajectory by using the image-based controller proposed in (Mezouar and Chaumette, 2002). Figures 2a and 2b show the obtained results: as we can see, the camera reaches the desired location by avoiding collisions with the obstacle in spite of the introduced uncertainties.

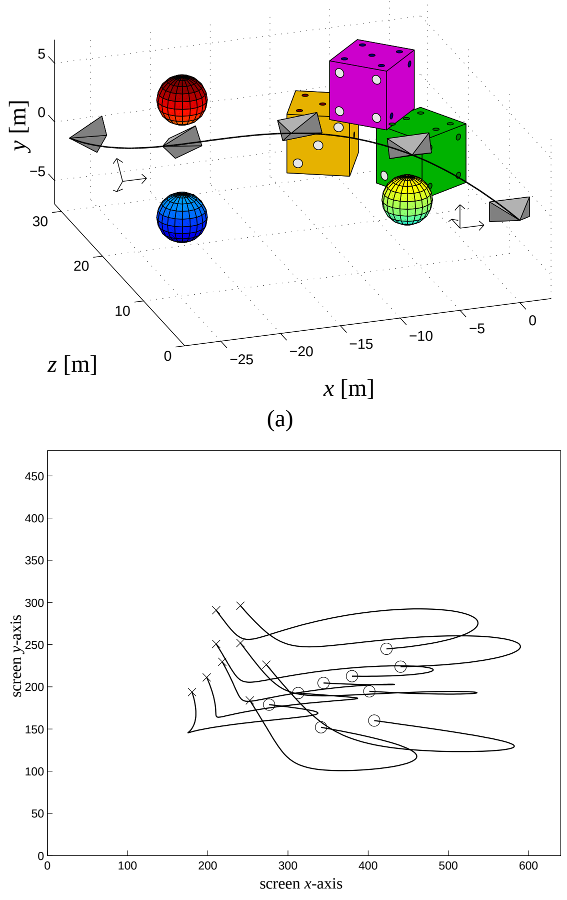


(a)

(b)

Figure 2: Results obtained by planning the trajectory with image noise and calibration errors, and by tracking the planned trajectory with an image-based controller.

Lastly, we consider a more difficult case by introducing three obstacles as shown in Figure 3a. We find that the LMIs are feasible by using polynomials of degree three, and the found solution provides the trajectory shown in Figures 3a and 3b, which satisfies the required constraints.



(a)

(b)

Figure 3: Results obtained for a different set of obstacles.

5 CONCLUSIONS

We have proposed a trajectory planning strategy for closed-loop control of robotic systems with visual feedback, which allows one to take into account multiple constraints during the motion such as limited field of view of the camera and limited workspace of the robot. This strategy is based on generating a parametrization of the trajectories connecting the initial location to the desired one. The trajectory constraints are imposed by using polynomial relaxations and LMIs. Future work will investigate the application of the proposed approach in real experiments.

ACKNOWLEDGEMENTS

The author would like to thank the Reviewers for their time and useful comments. This work was supported by the Research Grants Council of Hong Kong Special Administrative Region under Grant HKU711208E.

REFERENCES

Allotta, B. and Fioravanti, D. (2005). 3D motion planning for image-based visual servoing tasks. In *Proc. IEEE Int. Conf. on Robotics and Automation*, Barcelona, Spain.

Chaumette, F. and Hutchinson, S. (2006). Visual servo control, part I: Basic approaches. *IEEE Robotics and Automation Magazine*, 13(4):82–90.

Chaumette, F. and Hutchinson, S. (2007). Visual servo control, part II: Advanced approaches. *IEEE Robotics and Automation Magazine*, 14(1):109–118.

Chesi, G. (2009a). Camera displacement via constrained minimization of the algebraic error. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 31(2):370–375.

Chesi, G. (2009b). Visual servoing path-planning via homogeneous forms and LMI optimizations. *IEEE Trans. on Robotics*, 25(2):281–291.

Chesi, G., Garulli, A., Tesi, A., and Vicino, A. (2003). Solving quadratic distance problems: an LMI-based approach. *IEEE Trans. on Automatic Control*, 48(2):200–212.

Chesi, G., Garulli, A., Tesi, A., and Vicino, A. (2009). *Homogeneous Polynomial Forms for Robustness Analysis of Uncertain Systems*. Springer (in press).

Chesi, G. and Hashimoto, K. (2004). A simple technique for improving camera displacement estimation in eye-in-hand visual servoing. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(9):1239–1242.

Chesi, G., Hashimoto, K., Prattichizzo, D., and Vicino, A. (2004). Keeping features in the field of view in eye-in-hand visual servoing: a switching approach. *IEEE Trans. on Robotics*, 20(5):908–913.

Chesi, G. and Hung, Y. S. (2007). Global path-planning for constrained and optimal visual servoing. *IEEE Trans. on Robotics*, 23(5):1050–1060.

Chesi, G., Tesi, A., Vicino, A., and Genesio, R. (1999). On convexification of some minimum distance problems. In *5th European Control Conf.*, Karlsruhe, Germany.

Chesi, G. and Vicino, A. (2004). Visual servoing for large camera displacements. *IEEE Trans. on Robotics*, 20(4):724–735.

Corke, P. I. and Hutchinson, S. (2001). A new partitioned approach to image-based visual servo control. *IEEE Trans. on Robotics and Automation*, 17(4):507–515.

Cowan, N. J. and Chang, D. E. (2005). Geometric visual servoing. *IEEE Trans. on Robotics*, 21(6):1128–1138.

Deng, L., Janabi-Sharifi, F., and Wilson, W. J. (2005). Hybrid motion control and planning strategy for visual servoing. *IEEE Trans. on Industrial Electronics*, 52(4):1024–1040.

Gans, N. and Hutchinson, S. (2007). Stable visual servoing through hybrid switched-system control. *IEEE Trans. on Robotics*, 23(3):530–540.

Hashimoto, K. (1993). *Visual Servoing: Real-Time Control of Robot Manipulators Based on Visual Sensory Feedback*. World Scientific, Singapore.

Hashimoto, K., Kimoto, T., Ebine, T., and Kimura, H. (1991). Manipulator control with image-based visual servo. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 2267–2272.

Iwatsuki, M. and Okiyama, N. (2005). A new formulation of visual servoing based on cylindrical coordinate system. *IEEE Trans. on Robotics*, 21(2):266–273.

Kazemi, M., Gupta, K., and Mehran, M. (2009). Global path planning for robust visual servoing in complex environments. In *Proc. IEEE Int. Conf. on Robotics and Automation*, Kobe, Japan.

Lopez-Nicolas, G., Bhattacharya, S., Guerrero, J. J., Sagues, C., and Hutchinson, S. (2007). Switched homography-based visual control of differential drive vehicles with field-of-view constraints. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 4238–4244, Rome, Italy.

Malis, E. (2004). Visual servoing invariant to changes in camera-intrinsic parameters. *IEEE Trans. on Robotics and Automation*, 20(1):72–81.

Malis, E. and Chaumette, F. (2000). 2 1/2 D visual servoing with respect to unknown objects through a new estimation scheme of camera displacement. *Int. Journal of Computer Vision*, 37(1):79–97.

Malis, E., Chesi, G., and Cipolla, R. (2003). 2 $1/2$ D visual servoing with respect to planar contours having complex and unknown shapes. *Int. Journal of Robotics Research*, 22(10):841–853.

Mezouar, Y. and Chaumette, F. (2002). Path planning for robust image-based control. *IEEE Trans. on Robotics and Automation*, 18(4):534–549.

Miura, K., Hashimoto, K., Inooka, H., Gangloff, J. A., and de Mathelin, M. F. (2006). Model-less visual servoing using modified simplex optimization. *Journal Artificial Life and Robotics*, 10(2):131–135.

Park, J. and Chung, M. (2003). Path planning with uncalibrated stereo rig for image-based visual servoing under large pose discrepancy. *IEEE Trans. on Robotics and Automation*, 19(2):250–258.

Tahri, O. and Chaumette, F. (2005). Point-based and region-based image moments for visual servoing of planar objects. *IEEE Trans. on Robotics*, 21(6):1116–1127.

Tarbouriech, S., Soueres, P., and Gao, B. (2005). A multi-criteria image-based controller based on a mixed polytopic and norm-bounded representation of uncertainties. In *44th IEEE Conf. on Decision and Control and European Control Conf.*, pages 5385–5390, Seville, Spain.

Thuilot, B., Martinet, P., Cordesses, L., and Gallice, J. (2002). Position based visual servoing: keeping the object in the field of vision. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 1624–1629, Washington, D.C.

Yao, Z. and Gupta, K. (2007). Path planning with general end-effector constraints. *Robotics and Autonomous Systems*, 55(4):316–327.

VARIABLE GEOMETRY TRACKED UNMANNED GROUNDED VEHICLE
Model, Stability and Experiments

Jean-Luc Paillat, Phillipe Lucidarme and Laurent Hardouin
Laboratoire d'Ingénierie des Systèmes Automatisés (LISA), 62 Avenue Notre Dame du Lac, Angers, France
jlpaillat@gmail.com, {philippe.lucidarme,laurent.hardouin}@istia.univ-angers.fr

Keywords: UGV (Unmanned ground vehicle), Variable Geometry Single Tracked Vehicle (VGSTV), Obstacle clearing capabilities, Teleoperation, Stability, Geometric model, Dynamic model.

Abstract: This paper introduces an originally designed tracked robot. This robot belongs to the VGSTV (Variable Geometry Single Tracked Vehicle) category. It is equipped with two tracks mounted on an actuated chassis. The first joint is used to articulate the chassis and the second one to keep the tracks tightened. By controlling the chassis joint it becomes possible to adapt the shape of the tracks to the ground and even to release the tracks in order to pass through specific obstacles. A prototype has been build ; technical specifications and dynamic model are presented in this paper. Tele-operated experiments have been conducted and have shown that the stability of the robot have to be addressed. We experimented and compared two classical criterions based on the center of mass and on the zero moment point technique. Experimental results are discussed in the case of a staircase clearing.

1 INTRODUCTION

UNMANNED GROUNDED VEHICLE (UGV) is a topically research field applied to a wide range of applications like for example exploration or missions in hostile environments. Research laboratories and robotics companies are currently working on the design of tele-operated and autonomous robots. According to (Casper and Murphy, 2003) and (Carlson and Murphy, 2005) UGVs can be classified into three categories :

- Man-packable delineates the robots that can be carried by one man in backpacks.
- Man-portable delineates the robots that are too heavy to be easily carried by men, but small enough to be transported in a car or a HUMMV.
- Not man-portable delineates the robots that must be carried by a truck, a trailer or a crane.

The robots presented in this paper are classified in the man-packable and man-portable categories. In this class of robots, designers have to face the following dilemma: on one hand, build a small robot that can be easily carried and move into narrow environments. Unfortunately, it will generally result in poor obstacle clearing capability. On the other hand, build a bigger robot will increase its ability to surmount obstacles but will not enable the robot to go through narrow openings. The challenge is then to build a robot as small as possible with the higher obstacle clearing capability. Based on this observation the first part of this paper introduces the existing experimental and commercial robots and discusses about their clearing capabilities. The following of the paper describes an originally designed UGV (Fig. 2b). This robot can be classified into the Variable Geometry Single Tracked Vehicle (VGSTV) category, i.e. it has the mechanical ability to modify its own shape according to the ground configuration. The design of our prototype is described in the third part with a short discussion about the technical choices (information can be found on the project website: http://www.istia.univ-angers.fr/LISA/B2P2/b2p2.html). The next section introduces the dynamic model of the robot. The first tele-operated experiments have shown that the mass distribution is crucial to pass through large obstacles. This is why the last section discusses about the stability of the robot. The presented survey has been conducted for three reasons. First of all, the information computed about the stability of the robot may be useful to the operator during tricky operations. Secondly, this information may also be used to automatically disable clumsy commands and prevent the robot from toppling over. And finally, the stability cannot be ignored during autonomous motion which is our long term goal. The survey has been conducted on two criterions : static (the center of mass) and dynamic (based on the well known zero moment point technique). Experimental results are compared and discussed in the case of a staircase clearing. A general conclusion ends the paper.

2 EXISTING UGVs

2.1 Wheeled and Tracked Vehicles with Fixed Shape

This category gathers non variable geometry robots. Theoretically, this kind of vehicles are able to climb a maximum step twice less high than their wheel diameter. Therefore their dimensions are quite important to ensure a large clearing capability. This conception probably presents a high reliability (Carlson and Murphy, 2003) but those robots cannot be easily used in unstructured environments like after an earthquake (Casper and Murphy, 2003).

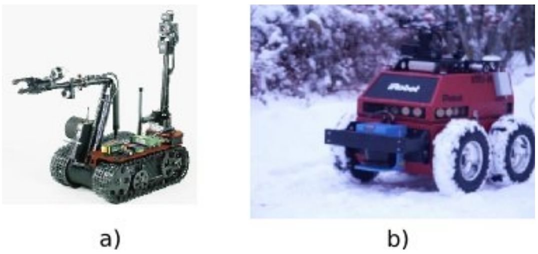


Figure 1: a) Talon-Hazmat robot (Manufacturer: Foster-Miller) b)ATRV-Jr robot. Photo Courtesy of AASS, Örebro University.

2.2 Variable Geometry Vehicle

A solution to ensure a large clearing capability and to reduce the dimensions consists in developing tracked vehicles which are able to modify their geometry in order to move their center of mass and climb higher obstacles than their wheel's diameters.

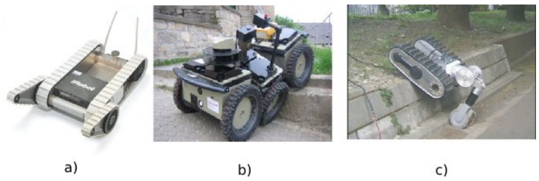


Figure 2: a)Packbot (manufacturer: IRobot), b)RobuROC 6(Manufacturer: Robosoft) c)Helios VII.

The Packbot robot (Fig. 2a) is probably one of the most famous commercial VGTV (Variable Geometry Tracked Vehicle). This robot is equipped with tracks and two actuated tracked flippers (372 mm long). The flippers are used to step over the obstacles. The obstacle clearing capability of this kind of VGTV depends on the size of the flippers. For more information and a detailed survey on clearance capability of the Packbot the reader can consult (Frost et al., 2002).

The robuROC6 (Fig. 2b) is equipped with 46.8 cm diameters wheels and can clear steps until 25cm (more than half the diameter of the wheels). Joints between the axles make this performance possible. An other original system called Helios VII (Fig 2c) (Guarnieri et al., 2004) is equipped with an arm ended by a passive wheel which is able to elevate the chassis along a curb.

2.3 Variable Geometry Single Tracked Robots

Actually, there is a subgroup in VGTV called Variable Geometry Single-Tracked Vehicles (VGSTV) (Kyun et al., 2005). It gathers robots equiped with as tracks as propulsion motors. In most cases those robots are equipped with one or two tracks (one for each side). It can be divided into two groups :

- robots with deformable tracks,
- robots with non deformable tracks.

2.3.1 Non-deformable Tracks VGSTV

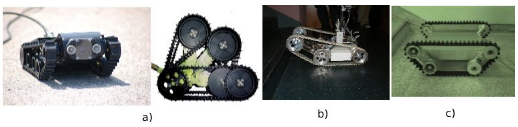


Figure 3: a) Micro VGTV (manufacturer: Inuktun Ltd) b)B2P2 prototype c)VGSTV mechanism.

The most famous example is the Micro VGTV. Illustrations of a prototypes manufactured by the company Inuktun are presented on Fig. 3a. This robot is based on an actuated chassis used to modify the shape of the robot. The right picture of Fig. 3a shows the superimposing configurations. The tracks are kept tightened by a passive mechanism. The robot is thus equipped with three motors: two for the propulsion and one for the chassis joint.

Non commercial vehicles exists in the literature as the VGSTV mechanism (Fig. 3c) which is dedicated to staircase clearing. It is composed of two tracks and two articulations which allow it to have many symmetrical configurations such as a rectangle, trapezoids, inverse trapezoids etc.

Many other VGTV architecture exist, for further information reader can consult (Vincent and Trentini, 2007), (Misawa, 1997), (Clement and Villedieu, 1987), (Guarnieri et al., 2004) and (Kyun et al., 2005).

2.3.2 Deformable Tracks VGSTV

Some single tracked robots have the ability to modify the flexing of their tracks. Two examples presented on Fig. 4, are able to adapt their shape to obstacles (Kinugasa et al., 2008). However, even if the control of the robot seems easier with a flexible track than with a non flexible one, the mechanical conception could be more complicated.

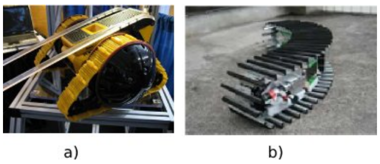


Figure 4: a) Viper robot (Manufacturer: Galileo) b)Rescue mobile track WORMY.

According to the presented state of the art, for general purpose missions we beleive that the best compromise between design complexity, reliability, cost and clearing capabilities is the Variable geometry single tracked robots category. The next section will introduce and describe our prototype of VGSTV.

3 PROTOTYPE DESCRIPTION

The main interest of VGSTV (equipped with deformable tracks or not) is that it is practicable to overcome unexpected obstacles (Kyun et al., 2005). Indeed, thanks to the elastic property of the tracks the clearance of a rock in rough terrain will be more smoothly with a VGSTV (e. g. Fig 3 and 4) than with a VGTV (Fig. 2). On the Micro VGTV presented on Fig. 3, the tension of the tracks is mechanically linked with the chassis joint so it is constant during the movement. Nevertheless, in some cases, less tense tracks could increase the clearance capability by increasing the adherence. An interesting study about this point was developed by (Iwamoto and Yamamoto, 1983) giving a VGSTV able to climb staircases where the tension of the tracks was mechanically managed as on the MicroVGTV (Fig. 3a). However, this system was equipped with a spring to allow the tracks to adapt their shape to the ground (depending on the strength of the spring).

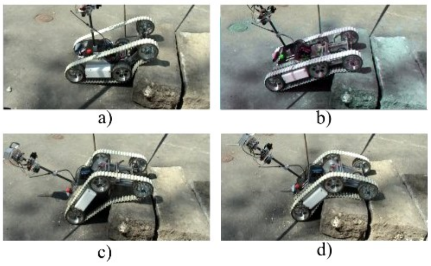


Figure 5: B2P2: clearing of a curb.

The conception of our prototype is based on this previous work, but we decided to actuate the tension of the tracks. Indeed, by using two motors instead of one (Fig. 6) it is possible to increase the tracks adaptation to the ground developed by (Iwamoto and Yamamoto, 1983) and reach new configurations for the robot. As example, the solution proposed in this paper allows our robot to adopt classical postures of VGSTV (Fig. 5(a), 5(b) and 5(c)), but also other interesting positions. On Fig. 5 B2P2 is clearing a curb of 30 cm height with tense tracks. The position of the robot on Fig 5(c) can also be obtained with the Micro VGTV, but it is a non-safety position and B2P2 is close to topple over. On Fig. 5(d) the tracks have just been released. They take the shape of the curb and it can be cleared safely. This last configuration outlines the interest of using an active system instead of a passive one. Consequently, although our prototype (Fig. 3b) belong to the VGSTV category and have not deformable tracks, it has the ability to adapt them to the ground (as deformable ones).

Besides, even if there is a risk of the tracks coming off, loosening the tracks may be an efficient mean of increase the surface in contact with the floor in rough terrain and then to improve the clearing capability of the structure. By the way, the risk could decreased by using sensor based systems to control the tension of the tracks or by modifying the mechanical structure of the robot (adding some kind of cramps on the tracks or using a guide to get back the tracks before it comes off).

3.1 Mechanical Description

This UGV is equipped with four motors. Fig. 6 presents the integration of the motors in the robot. Motors 1 and 2 are dedicated to the propulsion (tracks).

The actuated front part is composed of motors 3 and 4 :

- Motor 3 actuates the rotational joint, it allows the rotation of the front part around the second axle.
- Motor 4 actuates a driving screw, it controls the

distance between the second axle and the third one.

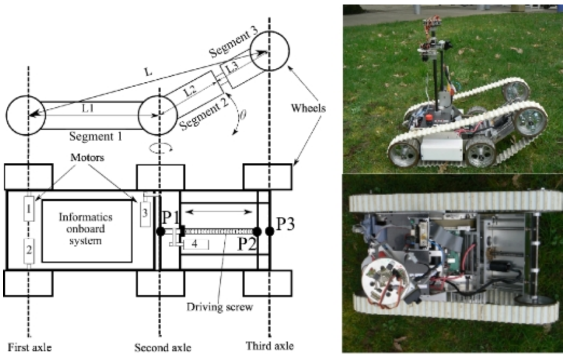


Figure 6: Overview of the mechanical structure, side and top view of the real robot.

To keep the tension of the tracks the trajectory of the third axle is given by an ellipse defined by two seats located on the first and the second axle.

$$L + (L_2 + L_3) = K \qquad (1)$$

where the lengths L, L_2 and L_3 are referenced on Fig. 6. K is a constant parameter depending on the length of the tracks, L_3 evolves in order to achieve equality (1) and is linked to the angle θ in the following manner:

$$L_3 = \frac{L_1^2 - K^2}{2(L_1 cos(\pi - \mid \theta \mid) - K)} - L_2 \qquad (2)$$

3.2 Sensors and Command Systems

The robot is equipped with multiple sensors, onboard/command systems and wireless communication systems.

- Onboard command systems :
  - PC104 equipped with a Linux OS compiled specifically for the robot needs based on a LFS (Home made light linux distribution).
  - An home-made I2C/PC104 interface.
  - Four integrated motor command systems running with RS232 serial ports.
  - Four polymer batteries which allow more than one hour of autonomy.
- Sensors :
  - An analog camera for tele-operation.
  - A GPS to locate the robot in outdoor environments.
  - A compass to know the orientation of the robot.
  - A two axis inclination sensor (roll and pitch).
- Wireless communication systems :
  - A 2.4 GHz analog video transmitter.
  - A bidirectional 152 MHz data transmitter.

4 DYNAMIC MODEL

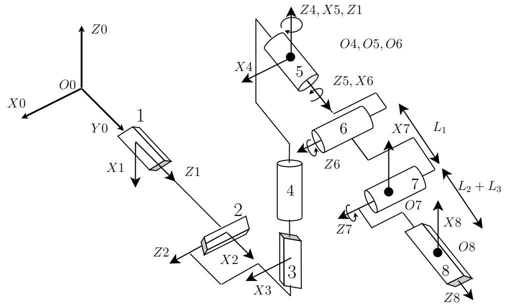


Figure 7: B2P2's geometric model. Joints 1,2 and 3 represents the robot position. Joints 4, 5 and 6 symbolize respectively the yaw, roll and pitch, 7 and 8 are the actuated joints.

This section deals with the dynamic model of the robot which is based on the geometric model (Fig. 7) detailed on (Paillat et al., 2008). According to this model, the robot motion in a 3D frame (R_0) is described by the vector q of the 8 joints variables :

$$q = [q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8]^T$$

The dynamic model of a mechanical system establishes a relation between the effort applied on the system and its coordinates, generalized speeds and accelerations ((Craig, 1989) and (Khalil and Dombre, 2004)). In this section, the following notations are used:

- j describes the joints from 1 to 8,
- i describes the segments from 1 to 3 (referenced on Fig 6),
- n and m describes indexes from 1 to 8.

4.1 The Dynamic Equations

The general dynamic equations of a mechanical system is:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_j} - \frac{\partial L}{\partial q_j} = Q_j + T_j \qquad (3)$$

- L is the Lagrangien of the system. It is composed of rigid segments, so there is no potential energy. Although the Lagrangien corresponds to the kinetic energy.
- q_j is the j^{th} joint variable of the system.
- Q_j is the gravity's torque applied to the j^{th} joint of the system.
- T_j is the external force's torque applied to the j^{th} joint of the system.

The kinetic energy is given by:

$$K = \sum_{i=1}^{n} \frac{1}{2} m_i v_i^T v_i + \frac{1}{2} w_i^T I_i w_i. \tag{4}$$

- m_i is the mass of the i^{th} element of the model,
- v_i is the linear speed of the i^{th} element's center of gravity,
- w_i is the angular speed of the i^{th} element's center of gravity,
- I_i is the matrix of inertia of the i^{th} element of the system.

In order to have homogeneous equations, w_i is defined in the same frame as I_i ; it allows to formulate v_i and w_i according to q :

$$v_i = J_{v_i}(q)\dot{q} \tag{5}$$

$$w_i = R_{0j}^T J_{w_i}(q)\dot{q} \tag{6}$$

where J_{v_i} and J_{w_i} are two matrices and R_{0j} is the transport matrix between the frame R_0 and the frame j linked to the segment i.

The kinetic energy formula is:

$$K = \frac{1}{2}\dot{q}^T \sum_i [m_i J_{v_i}(q)^T J_{v_i}(q) + J_{w_i}^T(q) R_{0j} I_i R_{0j}^T J_{w_i}(q)]\dot{q} \tag{7}$$

which can be rewritten as :

$$K = \frac{1}{2}\dot{q}^T D(q)\dot{q} \tag{8}$$

by developing the previous formula, we obtain :

$$K = \frac{1}{2}\sum_{m,n} d_{m,n}(q)\dot{q}_m\dot{q}_n \tag{9}$$

where $d_{m,n}(q)$ is the m,n^{th} element of the matrix $D(q)$.

The gravity's torque is given by:

$$Q_j = \sum_i g m_i \frac{\partial G_{zi}^0}{\partial q_j}. \tag{10}$$

- G_{zi}^0 is the z coordinate of the CoG of the i^{th} segment's computed in the base frame (R_0),
- g is the gravity acceleration.

Vector T (defined in (3)) is composed of the external forces' torque. For the robot presented here, there is no consideration of external forces, so the T vector only describes the motorized torques. Joints 1, 4, 7 and 8 are motorized, so the vector T is given by those four parameters. T_1 and T_4 are computed from the torques of motors 1 and 2 while T_7 and T_8 are deduced from motors 3 and 4.

The Euler-Lagrange equations can be written as:

$$\sum_m d_{jm}(q)\ddot{q}_m + \sum_{n,m} c_{nmj}(q)\dot{q}_n\dot{q}_m = Q_j + T_j \tag{11}$$

$$c_{nmj} = \frac{1}{2}[\frac{\partial d_{jm}}{\partial q_n} + \frac{\partial d_{jn}}{\partial q_m} - \frac{\partial d_{nm}}{\partial q_j}] \tag{12}$$

which is classically written as:

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} = Q + T \tag{13}$$

where $D(q)$ represents the matrix of inertia and $C(q,\dot{q})$ the centrifuge-coriolis matrix where X_{jm}, the jm^{th} element of this matrix, is defined as :

$$X_{jm} = \sum_n c_{nmj}\dot{q}_n.$$

Finally, the J_{vi} and J_{wi} matrix considered in (5) and (6) have to be computed.

4.2 J_{vi} and J_{wi} Matrix Formulation

The matrix which links articular speed and general speed of a segment is computed from the linear and angular speeds formulas. The goal is to find a matrix for each segment. They are composed of 8 vectors (one for each joint of the model).

The computation consists in formulating in the base frame, the speed $(V_{P_i}(j-1,j)^{R_0})$ of a point P_i given by a motion of the joint q_j attached to the frame j according to the frame $j-1$. Those parameters can be deduced from the law of composition speeds and the Denavitt Hartenberg (DH) formalism used for the geometric model (Paillat et al., 2008). Indeed, the general formulation is simplified by the geometric model. Only one degree of freedom (DoF) links two frames using the DH model and this DoF is a revolute or a prismatic joint. Moreover, the Z axis is always the rotation or translation axis, so the angular and linear speeds are given by four cases:

- The angular speed of a point for a revolute joint:

$$w_P(j-1,j)^{R_0} = R_{0,j}\begin{bmatrix}0\\0\\1\end{bmatrix}\dot{q}_j. \tag{14}$$

- The linear speed of a point for a revolute joint:

$$\begin{aligned} v_P(j-1,j)^{R_0} &= V_{O_j}^{R_{j-1}} + V_P^{R_j} + w_j \wedge O_jP^{R_j} \\ &= \dot{q}_j R_{0,j}\begin{bmatrix}0\\0\\1\end{bmatrix} \wedge R_{0,j}P_j. \end{aligned} \tag{15}$$

- The angular speed of a point for a prismatic joint:

$$w_P(j-1,j) = \begin{bmatrix}0\\0\\0\end{bmatrix}. \tag{16}$$

- The linear speed of a point for a prismatic joint:

$$v_P(j-1,j)^{R_0} = R_{0,j}\begin{bmatrix}0\\0\\1\end{bmatrix}\dot{q}_j \tag{17}$$

where P_j is the P point's coordinates in R_j.
Thus, the matrix of a segment i is formulated by computing speeds for each joints as :

$$\begin{bmatrix} v_i \\ w_i \end{bmatrix} = J(q)\dot{q} = [J_{1,i}(q), J_{2,i}(q), \ldots J_{8,i}(q)]\,\dot{q} \tag{18}$$

where $J_{j,i}(q)$ is a vector which links the speed of the i^{th} segment according to the j^{th} joint. The first segment is not affected by the motion of joints 7 and 8 while the second is not affected by joint 8, therefore $J_{7,1}(q)$, $J_{8,1}(q)$ and $J_{8,2}(q)$ are represented by a null vector.

5 BALANCE CRITERION

The balance criterion used here are the ZMP (Zero Moment Point), widely used for the stability of humanoid robots and the Center of Gravity (CoG). Previous theoretical works and experiments have proved the ZMP efficiency (Vukobratovic and Borovac, 2004). It consists in keeping the point on the ground at which the moment generated by the reaction forces has no component around x and y axis ((Kim et al., 2002) and (Kajita et al., 2003)) in the support polygon of the robot. When the ZMP is at the border of the support polygon the robot is teetering. Unlike the ground projection of the center of gravity, it takes into account the robot's inertia.

The purpose of the following is to defined the coordinates of this point in any frame of the model according to the configuration of the robot. The definition can be implemented into the Newton equations to obtain those coordinates. In any point of the model : $M_0 = M_z + OZ \wedge R$ (M_0 and M_z define respectively the moment generated by the reaction force R at the points 0 and z).

According to the previous definition, there is no moment generated by reaction forces at the Zero Moment Point. Consequently, if Z defines the ZMP coordinates $M_0 = OZ \wedge R$. This formulation can be implemented into the Newton equations as:

$$\delta_0 = M_0 + \vec{OG} \wedge \vec{P} + \vec{OG} \wedge \vec{F_i} \tag{19}$$

where P is the gravity force, G is the robot's center of gravity and F_i is the inertial force (the first Newton's law gives $F_i = -m\ddot{G}$). According to the ZMP definition, the equation (19) can be formulated as :

$$\delta_0 = \vec{OZ} \wedge \vec{R} + \vec{OG} \wedge \vec{P} + \vec{OG} \wedge \vec{F_i} \tag{20}$$

$$\begin{cases} \delta_{0x} = Z_yR_z + G_yP_z - G_zP_y - G_zFi_y \\ \delta_{0y} = -Z_xR_z + G_zP_x - G_xP_z \end{cases} \tag{21}$$

$$\begin{cases} Z_y = \frac{\delta_{0x} - G_yP_z + G_zP_y + G_zFi_y}{R_z} \\ Z_x = \frac{-\delta_{0y} + G_zP_x - G_xP_z}{R_z}. \end{cases} \tag{22}$$

Also, it is possible to compute the position of the ZMP as a function of q (δ_0 depends on the matrix $D(q)$).

Assuming the ground knowledge, the ZMP computation gives a criterion to determinate the stability of the platform.

6 RESULTS

This section presents the numerical computation of the criterion in the case of the clearance of a staircase (staircase set of 15 cm risers and 28 cm runs) with an average speed of 0.13 $m.s^{-1}$ (Fig. 9). The robot is equipped with a 2-axis inclination sensor that provides rolling and pitching. Vector q entries are measured using encoders on each actuated axis of the robot. Data have been stored during the experiments and the models (CoG and ZMP) have been computed off-line. This computation does not take into account the tracks' weight which is negligible in regard to the robot's weight. Fig. 8 presents the evolution of the ZMP (left) and the difference between those two criterion (right) during all the clearance. P_1 P_2 and P_3

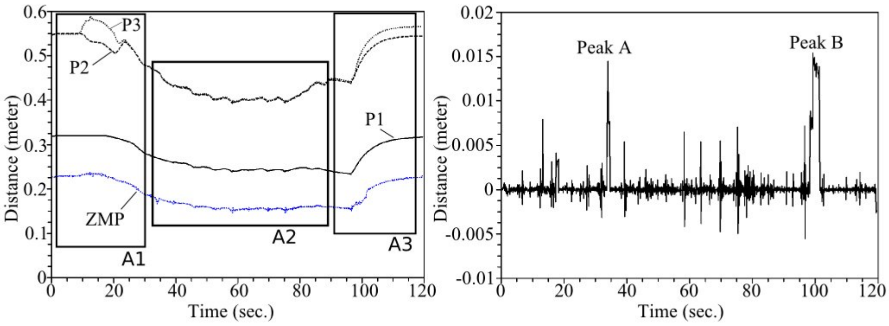


Figure 8: Experiment's results. The left chart represents the evolution of the ZMP and the right one, the difference between CoG and ZMP.

represents the z-coordinates in the frame R_5 (Fig. 7) of three points of the robot which localization are noticed on Fig. 6.

The following section finely details the results of the experiment and the correlation with the sensors data is described. This analyze is divided into three parts: the approach of the first step, the clearance of the middle steps and the clearance of the final step.

6.1 The Clearance of the First Step

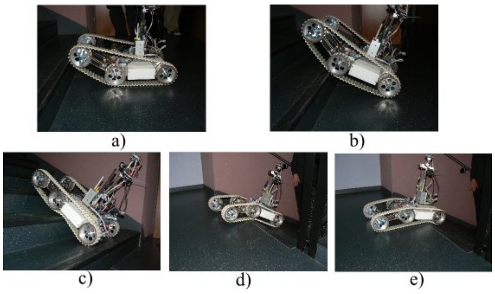


Figure 9: Clearance of a staircase.

First of all, the robot is approaching while moving up the front part (Fig. 9(a)) in order to go onto the first step. Then, it has to move forward and move down the elevation articulation in order to keep the stability (Fig. 9(b)). Once the robot is step onto the first stair, the operator have to switch in the next configuration. The area noted $A1$ on Fig. 8 shows the evolution of the ZMP projection during the clearance of the first step. Note that the tracks are tense because when the front part is rising up, there is a large difference between P_2 and P_3.

6.2 The Clearance of the Middle Steps

This stage starts in the position noticed on Fig. 9(c). By moving forward, the robot naturally climbs the stairs. At each step, the robot is gently swaying when the ZMP is passing over the step. This phenomenon is illustrated by the oscillation of the ZMP which are visible on the area noted A2 on Fig. 8. Note that, this oscillation is dependent on the ratio between the size of the robot and the size of the steps (”size-step” ratio). It fully disappears when the length of the robot is superior to the size of three steps. On the other hand, oscillations may be more important until reaching a ”size-step” ratio where the robot cannot climb the step.

6.3 The Clearance of the Final Step

The robot is moving forward while moving down its front part (Fig. 9(d)). This operation brings the ZMP closer to the limits of the support polygon, i.e. the corner of the last step. This operation allows a smooth swing of the ZMP. Area A3 on Fig. 8 shows the evolution of the ZMP during the clearing of the last step.

Note that the tracks releasing was not used during this experiment because it was not necessary to overcome this staircase. It could become essential for bigger obstacles.

This experiment allows us to validate the presented model and confirms the computation of the ZMP criterion. However, as it is shown on Fig. 8, the average difference between the ZMP and the COG is insignificant (about 0.21%). Moreover, the two peaks (A and B) on the Fig. 8 are not due to the dynamics of the system but to measurement errors. As the acceleration is measured with the encoders (linked to the motor shaft), when the tracks slip, the measurement is erroneous. The ZMP is computationally more expensive, needs more sensor measurements and the difference with the CoG is negligible. For these reasons, we conclude that the CoG seems well suited for this kind of experiments. Anyway, in the case where fast obstacle clearance may be necessary, the CoG may not longer be considered and the ZMP must be used instead.

7 CONCLUSIONS

In this paper, an original prototype of VGSTV has been presented. The dynamic model have been introduced. From this model, a stability criterion based on the ZMP technique has been computed. This model have been validated on the real robot in the context of a staircase clearance.

Future works will focus on the autonomy of the robot. In a short term work, a real time stability assistance will be provided to the operator based on the model (CoG or ZMP) presented in this paper. The presented results are also a preliminary survey for a long term work that will be oriented on the autonomous obstacle clearance. Actual work (based on the model and results presented here) is focused on the automatic control of the robot.

REFERENCES

Carlson, J. and Murphy, R. R. (2003). Reliability analysis of mobile robots. *International Conferenee on Robotics and Automation.*

Carlson, J. and Murphy, R. R. (2005). How ugvs physically fail in the field. *IEEE Transactions on robotics*, 21(3):423–437.

Casper, J. and Murphy, R. R. (2003). Human-robot interactions during the robot-assisted urban search and rescue response at the world trade center. *IEEE Transactions on systems, man, and cybernetics*, 33(3):367–384.

Clement, G. and Villedieu, E. (1987). Variable geometry track vehicle. Us patent.

Craig, J. J. (1989). *Introduction to Robotics Mechanics and control*. Silma, 2 edition.

Frost, T., Norman, C., Pratt, S., and Yamauchi, B. (2002). Derived performance metrics and measurements compared to field experience for the packbot. *Proceedings of the 2002 PerMIS Workshop.*

Guarnieri, M., Debenest, P., Inoh, T., Fukushima, E., and Hirose, S. (2004). Development of helios vii : an arm-equipped tracked vehicle for search and rescue operations. *IEEE/RSJ Int. Conference on Intelligent Robots.*

Iwamoto, T. and Yamamoto, H. (1983). Variable configuration track laying vehicle (us patent).

Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Harada, K., Yokoi, K., and Hirukawa, H. (2003). Biped walking pattern generation by using preview control of zero moment point. Proceedings of the 2002 IEEE Internationnal Conference on Robotics ans Automation.

Khalil, W. and Dombre, E. (2004). *Modeling, Identification and Control of Robots*. Kogan Page Science, 2 edition.

Kim, J., Chung, W. K., Youm, Y., and Lee, B. H. (2002). Real time zmp compensation method using null motion for mobile manipulators. Proceedings of the 2002 IEEE Internationnal Conference on Robotics ans Automation.

Kinugasa, T., Otani, Y., Haji, T., Yoshida, K., Osuka, K., and Amano, H. (2008). A proposal of flexible monotread mobile track. Internationnal Conference on Intelligent Robots and Systems.

Kyun, L. S., Il, P. D., Keun, K. Y., Byung-Soo, K., and Sang-Won, J. (2005). Variable geometry single-tracked mechanism for a rescue robot. *IEEE International Workshop on Safety, Security and Rescue Robotics*, pages 111–115.

Misawa, R. (1997). Stair-climbing crawler transporter. Us patent.

Paillat, J. L., Lucidarme, P., and Hardouin, L. (2008). Variable geometry tracked vehicle, description, model and behavior. Conference Mecatronics.

Vincent, I. and Trentini, M. (2007). Shape-shifting tracked robotic vehicle for complex terrain navigation: Characteristics and architecture. Technical memorandum, Defence R and D Canada.

Vukobratovic, M. and Borovac, B. (2004). Zero-momment point - thirty five years of its life. *Internationnal journal of humanoid Robotics*, 1(1):157–173.

PASSIVITY-BASED DYNAMIC BIPEDAL WALKING WITH TERRAIN ADAPTABILITY
Dynamics, Control and Robotic Applications

Qining Wang, Long Wang, Jinying Zhu, Yan Huang and Guangming Xie
Intelligent Control Laboratory, College of Engineering, Peking University, Beijing 100871, China
qiningwang@pku.edu.cn

Keywords: Passive dynamic walking, Bipedal robots, Terrain adaptability, Modeling.

Abstract: This paper presents an approach for passivity-based bipedal robots to achieve stable walking on uneven terrain. A powered two-dimensional seven-link walking model with flat feet and compliant ankles has been proposed to analyze and simulate the walking dynamics. We further describe an optimization based control method, which uses optimized hip actuation and ankle compliance as control parameters of bipedal walking. Satisfactory results of simulations and real robot experiments show that the passivity-based walker can achieve stable bipedal walking with larger ground disturbance by the proposed method in view of stability and efficiency.

1 INTRODUCTION

Stability guaranteed dynamic bipedal walking is one of the keys but also one of the more challenging components of humanoid robot design. Several actively controlled bipedal robots are able to deal with such dynamic stability when walk on irregular surface (Yamaguchi and Takanishi, 1997), step over obstacles (Guan et al., 2006), climb stairs (Michel et al., 2007), even if achieve bipedal running (Honda Motor, 2005). However, to increase the autonomy of the robot, the locomotion efficiency, which is far from that of human motion, has to be improved.

As one of the possible explanations for the efficiency of the human gait, passive dynamic walking (McGeer, 1990) showed that a mechanism with two legs can be constructed so as to descend a gentle slope with no actuation and no active control. Several studies reported that these kinds of walking machines work with reasonable stability over a range of slopes (e.g. (McGeer, 1990), (Collins et al. 2001)) and on level ground with kinds of actuation added (e.g. (Collins et al. 2005), (Wisse et al., 2007)) show a remarkable resemblance to the human gait. In spite of having high energetic efficiency, passivity-based walkers have limits to achieve adaptive locomotion on rough terrain, which is one of the most important advantages of the legged robots. In addition, these walkers are sensitive to the initial conditions of walking.

To overcome such disadvantages, several studies proposed quasi-passive dynamic walking methods, which implement simple control schemes and actuators supplementarily to handle ground disturbances. For example, (Schuitema et al., 2005) described a reinforcement learning based controller and showed that the walker with such controller can maximally overcome $-10mm$ ground disturbance. (Tedrake et al., 2004) presented a robot with a minimal number of degrees of freedom which is still capable of stable dynamic walking even on level ground and even up a small slope. (Wang et al., 2006) designed a learning controller for a two-dimensional biped model with two rigid legs and curved feet to walk on uneven surface that monotonically increases from $12mm$ to $40mm$ with a $2mm$ interval. In these studies, passivity-based walkers are often modeled with point feet or round feet. And the control parameter only includes hip actuation. None of them analyzed the stability or adaptability in quasi-passive dynamic walking with flat feet and ankle joint compliance which is more close to human motion. In fact, flat feet can offer the advantage of distributing the energy loss per step over two collisions, at the heel and at the toe (Kwan and Hubbard, 2007), (Ruina et al., 2005). Moreover, experiments on human subjects and robot prototypes revealed that the tendon of the muscle in ankle joint is one mechanism that favors locomotor economy (Fukunaga et al., 2001), (Wang et al., 2007). It is predictable that by controlling and optimizing the hip actuation and ankle compliance, the passivity-based bipedal walker may achieve adaptive bipedal locomotion with larger disturbance on uneven terrain.

In this paper, we investigate how to control passivity-based bipedal walkers to achieve stable walking with terrain adaptability. A powered two-dimensional seven-link walking model with flat feet and ankle compliance has been proposed to analyze and simulate the walking dynamics. We hypothesized that the nervous system that controls human locomotion may use optimized hip actuation and joint compliance to achieve stable bipedal walking on irregular terrain. Actually, hip actuation will cause the walker to change locomotive patterns. Furthermore, we use both simulations and real robot prototype to explore the performance of the passivity-based walkers when walk on uneven terrain by utilizing a biologically inspired optimization based controller, which is adaptive and capable of selecting proper hip actuation and ankle compliance in view of walking stability and efficiency.

This paper is organized as follows. Section II describes the walking dynamics of the biped with flat feet and ankle compliance. In Section III, we present the optimization based walking control method. In section IV, experimental results of stable locomotion with terrain adaptability are shown. We conclude in Section V.

2 PASSIVITY-BASED BIPEDAL WALKER

Our model extended the Simplest Walking Model (Garcia et al., 1998) with the addition of hip joint actuation, upper body, flat feet and linear spring based compliance on ankle joints, aiming for adaptive bipedal locomotion with optimization based controller. The model consists of an upper body (point mass added stick) that rotates around the hip joint, a point mass representing the pelvis, two legs with knee joints and ankle joints, and two mass added flat feet (see Fig. 1).

The mass of each leg is simplified as point masses added on the Center of Mass (CoM) of the shank and the thigh respectively. Similar to (Wisse et al., 2004), a kinematic coupling has been used in the proposed model to keep the body midway between the two legs. In addition, our model added compliance in ankle joints. Specifically, the ankle joints are modeled as passive joints that are constrained by linear springs. The model is implemented in MATLAB, using the parameter values shown in Table 1, which are derived from the mechanical prototype.

The passive walker travels forward on level ground. The stance leg keeps contact with the ground while the swing leg pivots about the constraint hip.

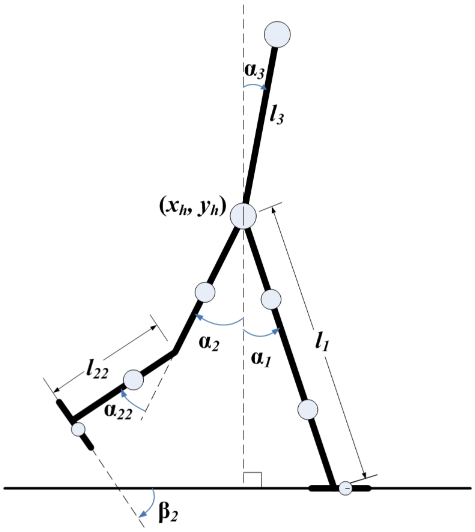


Figure 1: Two-dimensional seven-link passive dynamic walking model. Mechanical energy consumption of level ground walking is compensated by applying a hip torque. The global coordinates of the hip joint is notated as (x_h, y_h). α_1 and α_2 are the angles between each leg and the vertical axis in sagittal plane respectively. The knee joints and ankle joints are all passive joints. To simulate human ankle compliance, the ankle joints of the model are constrained by linear springs.

Table 1: Default Parameter Values for the dynamic walking model and the following simulations.

Parameter	Description	Value
m_1, m_2	leg mass	$1.12kg$
m_3	upper body mass	$0.81kg$
m_4	hip mass	$15.03kg$
m_{1t}, m_{2t}	thigh mass	$0.56kg$
m_{1s}, m_{2s}	shank mass	$0.56kg$
m_{f1}, m_{f2}	foot mass	$2.05kg$
l_1, l_2	leg length	$0.7m$
l_{11}, l_{22}	shank length	$0.35m$
l_{f1}, l_{f2}	foot length	$0.15m$
l_b	upper body length	$0.5m$
l_w	body width	$0.15m$
c_b	CoM of upper body	$0.2m$
c_1, c_2	CoM of leg	$0.2m$
c_{11}, c_{22}	CoM of shank	$0.2m$
k	ankle stiffness	$8.65Nm/rad$
P	hip torque	$0.38Nm$

To compensate the mechanical energy consumption, similar to (Kuo, 2002), we added a hip torque P between the swing leg and the stance leg. When the flat foot strikes the ground, there are two impulses, ”heel-strike” and ”foot-strike”, representative of the initial impact of the heel and the following impact as the whole foot contacts the ground.

The walking model can be defined by the rectangular coordinates x which can be described by the x-coordinates and y-coordinates of the mass points and the corresponding angles. The walker can also be described by the generalized coordinates q. The springs on the ankles constrain the foot vertical to the shank when no heel-strike or foot-strike has occurred.

2.1 Constraint Equations

The constraint equations ξ_1 and ξ_2 used to detect heel contact with ground are defined as follows:

$$\begin{aligned} \xi_1 &= \begin{bmatrix} x_h + l_1 \sin\alpha_1 - l_{f1}\cos\alpha_1 - x_{f1} \\ y_h - l_1\cos\alpha_1 - l_{f1}\sin\alpha_1 \end{bmatrix} \\ \xi_2 &= \begin{bmatrix} x_h + l_2 \sin\alpha_2 - l_{f2}\cos\alpha_2 - x_{f2} \\ y_h - l_2\cos\alpha_2 - l_{f2}\sin\alpha_2 \end{bmatrix} \end{aligned} \tag{1}$$

where x_{f1} and x_{f1} are the global x-coordinates of the latest strike. When the flat foot completely contacts the ground, the constraint equations ξ_3 and ξ_4 used to maintain foot contact with ground are defined as follows:

$$\begin{aligned} \xi_3 &= \begin{bmatrix} x_h + l_1 \sin\alpha_1 - x_{f1} \\ y_h - l_1\cos\alpha_1 \end{bmatrix} \\ \xi_4 &= \begin{bmatrix} x_h + l_2 \sin\alpha_2 - x_{f2} \\ y_h - l_2\cos\alpha_2 \end{bmatrix} \end{aligned} \tag{2}$$

If only the heel contacts the ground, the constraint equations ξ_5 and ξ_6 during the period before foot-strike are defined as follows:

$$\begin{aligned} \xi_5 &= (x_h + l_1\sin\alpha_1 - x_{f1})^2 + (y_h - l_1\cos\alpha_1)^2 - l_{f1}^2 \\ \xi_6 &= (x_h + l_2\sin\alpha_2 - x_2)^2 + (y_h - l_2\cos\alpha_2)^2 - l_{f2}^2 \end{aligned} \tag{3}$$

Note that ξ_5 and ξ_6 are similar to constraining the ankle joint that connects shank and foot to move in a circular orbit with heel as the center and distance between heel and ankle joint as the radius.

Similar to (Wisse et al., 2004), a reduced mass matrix M_r is introduced, which is defined as follows:

$$[M_r] = [T]^T [M_g] [T] \tag{4}$$

where the Jacobian $T = \frac{\partial x}{\partial q}$. Here x is the global coordinates of the six pointmasses (stance shank with foot, swing shank with foot, hip, stance thigh, swing thigh, body), while M_g is the mass matrix in global coordinates. Matrix Ξ_i transfers the independent generalized coordinates q into the constraint equation ξ_i, where $i = 1, 2, \ldots, 6$

$$\Xi_i = \frac{d\xi_i}{dq} \tag{5}$$

Consequently, matrix $\tilde{\Xi}_i$ is defined as follows:

$$\tilde{\Xi}_i = \frac{\partial(\Xi_i \dot{q})}{\partial q}\dot{q} \tag{6}$$

2.2 Single Support Phase

Suppose that leg 1 (l_1) is the stance leg, while leg 2 (l_2) is the swing leg. In the beginning of the single support phase, the knee joint is locked (keep the shank and the thigh in a straight line). The Equation of Motion (EoM) is described as follows:

$$\begin{bmatrix} M_r & \Xi_3^T \\ \Xi_3 & 0 \end{bmatrix} \begin{bmatrix} \ddot{q} \\ F_c \end{bmatrix} = \begin{bmatrix} F_r \\ -\tilde{\Xi}_3 \end{bmatrix} \tag{7}$$

where F_r is the external force, while F_c is the foot contact force. Here, the external force F_r is used to compensate the mechanical energy consumption of level ground walking, which defined as follows:

$$\{F_r\} = [T]^T (\{F\} - [M_g]\{\ddot{x}\})$$

where F is the external force in global coordinates, including gravity, hip actuation, and torque in compliant ankle joints. Then when the swing leg is swung forward, the knee joint releases the shank.

2.3 Heel-strike Phase

In this phase, leg 1 (l_1) is still the stance leg, while leg 2 (l_2) is the swing leg. The heel contacts the ground (heel-strike occurs). The EoM of the model changes to:

$$\begin{bmatrix} M_r & \Xi_6^T \\ \Xi_6 & 0 \end{bmatrix} \begin{bmatrix} \dot{q}^+ \\ F_c \end{bmatrix} = \begin{bmatrix} M_r\dot{q}^- \\ -e\Xi_6\dot{q}^- \end{bmatrix} \tag{8}$$

After the heel-strike, the foot rotates around the ankle joint. The EoM of the model is:

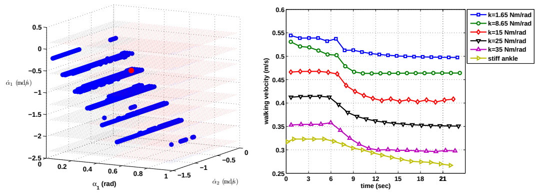
$$\begin{bmatrix} M_r & \Xi_3^T & \Xi_6^T \\ \Xi_3 & 0 & 0 \\ \Xi_6 & 0 & 0 \end{bmatrix} \begin{bmatrix} \ddot{q} \\ F_{c1} \\ F_{c2} \end{bmatrix} = \begin{bmatrix} F_r \\ -\tilde{\Xi}_3 \\ -\tilde{\Xi}_6 \end{bmatrix} \tag{9}$$

Note that the constraint equations guarantee that the stance leg maintains foot contact with ground and the swing leg maintains heel contact with ground. In addition, the spring constant k in the compliant ankle joints should not be too big. Otherwise, the stance leg will lose contact with ground. In this phase, since the foot rotates around the ankle joint, the force generated by the springs on the swing leg should be considered as external force. Thus, in this event, the mass matrix M would not include the point mass of the swing foot.

2.4 Toe-strike Impulse

The proposed walking model with flat feet introduces a toe-strike impulse in addition to the heel-strike collision. The EoM of the model is:

$$\begin{bmatrix} M_r & \Xi_4^T \\ \Xi_4 & 0 \end{bmatrix} \begin{bmatrix} \dot{q}^+ \\ F_c \end{bmatrix} = \begin{bmatrix} M_r\dot{q}^- \\ -e\Xi_4\dot{q}^- \end{bmatrix} \tag{10}$$



(a) (b)

Figure 2: (a) Basin of attraction of the passive dynamic walking model with flat feet and compliant ankles. The blue layers of points represent horizontal slices of a three-dimensional region of initial conditions that eventually result in the cyclic walking motion. The fixed point is indicated with a red point, which is above one of the sample slices. (b) Results of actively changing walking speed with the same hip actuation and different ankle compliance (k varies).

Note that in this phase, we consider that the ankle joint of the swing leg is constrained to move in a circular orbit with toe as the center and distance between toe and ankle joint as the radius. The force generated by the spring on the swing leg should be considered as external force, which can also be considered as the constraint force of the circular orbit. After the toe-strike, one step ends.

2.5 Effects of Hip Actuation and Ankle Compliance

By application of the cell mapping method that has been used in the several studies of passive dynamic walking (e.g. (Wisse et al., 2004), (Wisse et al., 2007)), we found that the model performs well in the concept of global stability. The allowable errors can be much larger than the results obtained in (Wisse et al., 2004). This can be inspected by the evaluation of the basin of attraction as shown in Fig. 2(a), which is the complete set of initial conditions that eventually result in cyclic walking motion. One can find that cyclic walking with initial conditions in Table 1, emerges even if the initial step is nearly fourfold as large, e.g. $\{\alpha_1(0) = 0.8, \dot{\alpha}_1(0) = -2.3, \dot{\alpha}_2(0) = -0.8\}$. It indicates that passive dynamic walking with flat feet and ankle compliance may play better on rough terrain with ground disturbance.

It has been examined that optimized hip actuation can improve the stability of the passive walker (Kurz and Stergiou, 2005). In addition, one can use the hip actuation as the control parameter to achieve stable walking on uneven terrain (Schuitema et al., 2005)-(Ueno et al., 2006). In our model mentioned above, ankle compliance k can also be used as a control parameter to affect the walking gait. As shown in Fig. 2(b), under the same change of hip actuation, different ankle compliance reveals different responses in walking velocity transition. It indicates that more compliance results in more visible sensitivity to the change of hip torque. According to the analysis of basins of attraction with different k (Wang et al., 2007), we find that a relatively small k will lead to more stable points. However, more compliance in ankle joints may result in often falling backward during walking. Thus, optimized ankle compliance may result in a more stable bipedal walking that allows larger disturbance.

3 OPTIMIZATION BASED WALKING CONTROL

In order to optimize the hip actuation and ankle compliance which affect walking gait as analyzed above, Particle Swarm Optimization (PSO) has been chosen with a focus lying on quickly finding suitable results, in view of time-consuming and adaptivity of the gait. In the realization of the PSO algorithm, a swarm of N particles is constructed inside a D-dimensional real valued solution space, where each position can be a potential solution for the optimization problem. The position of each particle is denoted as X_i $(0 < i < N)$. Each particle has a velocity parameter V_i $(0 < i < N)$. It specifies that the length and the direction of X_i should be modified during iteration. A fitness value attached to each location represents how well the location suits the optimization problem. The fitness value can be calculated by a fitness function of the optimization.

In this study, we used adaptive PSO with changing inertia weight. The update equation for velocity with inertial weight is described as follows:

$$v_{id}^{k+1} = wv_{id}^{k} + c_1 r_{1d}^{k}(pbest_{id}^{k} - x_{id}^{k}) + c_2 r_{2d}^{k}(gbest_{d}^{k} - x_{id}^{k}) \tag{11}$$

where w is the inertia weight. v_{id}^k is one component of V_i (d donates the component number) at iteration k. Similarly, x_{id}^k is one component of X_i at iteration k. $pbest_i$ $(0 < i < N)$ and $gbest$ are the personal best position and the global best position at each iteration respectively. c_1 and c_2 are acceleration factors. r_1 and r_2 are random numbers uniformly distributed between 0 and 1. Note that each component of the velocity has new random numbers. In order to prevent particles from flying outside the searching space, the amplitude of the velocity is constrained inside a spectrum $[-v_d^{max}, +v_d^{max}]$.

3.1 Fitness Function and Optimization Process

For a specific passive walker, the mechanical parameters (length and mass distribution) are constant. To control passive dynamic walking on uneven terrain, we focus the control parameters on hip actuation P and ankle compliance k. Then the two-dimensional parameter space is $(P,\ k)$. Here a set of parameters stands for a particle of PSO. Since the walker will be optimized with integration of stability and efficiency, the fitness function is defined as follows:

$$\sigma = \sigma_s + \gamma\sigma_e \tag{12}$$

where σ_s and σ_e are the fitness value to assess the stability and efficiency of each set of parameters respectively. γ is the tuning factor to change the importance of the two characteristics.

There are several methods to evaluate the stability of the passive dynamic walking. In this study, the stability will be quantified by the modulus of the Jacobian matrix of the mapping function as defined in (Wisse et al., 2004). Here, we notate the maximal eigenvalue as λ_m, which represents the decreasing speed of the deviation. The stability grades varies for different sets of parameters even they all have a stable fixed point. The smaller the λ_m is, the faster the deviation decreases and the more stable the walker is. The similar conclusion can be obtained when all sets of parameters only have an unstable fixed point. The larger the λ_m is, the more far from the stable state the walker is. Then we define the fitness function of stability as the follows:

$$\sigma_s = \frac{1}{\lambda_m} \tag{13}$$

Similar to (Collins et al. 2005) and (Wisse et al., 2004), the energetic efficiency of walking can be evaluated by the specific resistance as follows:

$$\Theta = \frac{E}{MgL} \tag{14}$$

where E is the cost of energy. In this study, the energy cost is generated only by the hip torque. M is the total mass of the model. g is the acceleration of gravity. L is the length of distance the robot passed. Then the fitness function can be defined as follows:

$$\sigma_e = \frac{1}{\Theta} = \frac{Mgl}{E} \tag{15}$$

From (12), (13) and (15), we can obtain the whole expression of the fitness function:

$$\sigma = \frac{1}{\lambda_m} + \gamma\frac{Mgl}{E} \tag{16}$$

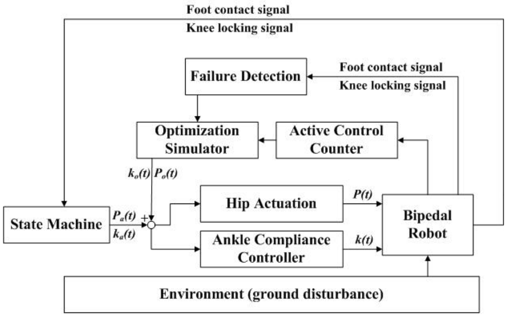


Figure 3: The control scheme to overcome ground disturbance with optimized hip torque and ankle compliance.

Additionally, to evaluate the walking motion after overcoming ground disturbance, we introduce another expression of fitness function. We use further walking distance d instead of the fitness efficiency σ_e. Then the fitness function can be rewritten as:

$$\sigma = \frac{1}{\lambda_m} + \gamma d \tag{17}$$

3.2 Gait Controller with Optimized Parameters

After analyzing the effects of hip actuation and ankle compliance in the stability and adaptability of the passive dynamic walker, we select P and k as the gait control input. The output of the optimization simulator $P_o(t)$ and $k_o(t)$ are added to the current actual hip actuation $P_a(t)$ and ankle compliance $k_a(t)$ (see Fig. 3).

This results in extra hip torque to move the swing leg more forward and prevent tripping. The purpose of the failure-detection block in Fig. 3 is to monitor in simulations the foot contact and the knee locking in order to detect whether walking failed. A failure means that the robot fell either backward or forward or that it started running (both feet leave contact with ground). There is a active control counter module in the diagram. It is used to count the times of applying active control (increasing or decreasing P or k) during one continuous walking. The output of this module make the simulator change P or k. With the dynamic model that adequately describing the real robot, an adaptive optimizing control scheme can be done without manually set to teach the robot and without the robot damaging itself.

4 EXPERIMENTAL RESULTS

All the simulation experiments used the dynamic model mentioned in Section II which was imple-

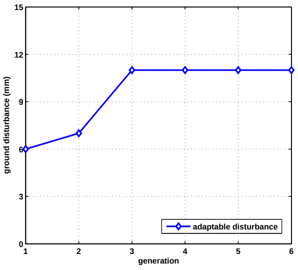


Figure 4: Maximum ground disturbance with optimized hip torque and ankle compliance that keep unchanged during stable walking.

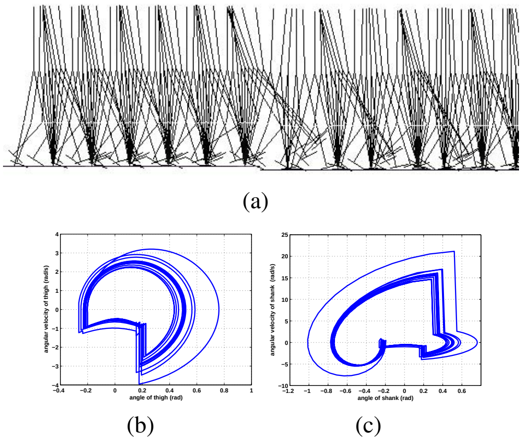
mented in MATLAB, using the parameter values shown in Table 1. The numerical integration of the second order differential EoMs uses the Runge-Kutta method, which is similar to the simulation methods mentioned in (Wisse et al., 2004).

4.1 Parameter Optimization

Based on the adaptive PSO with proper inertial weight mentioned above, we optimized the hip torque (hip actuation) and ankle compliance to achieve adaptive walking with maximal allowable ground disturbance of the model with parameter values in Table 1. The testing scenario is a floor with one step down. The height of the step is gradually chosen from the range from $1mm$ to $20mm$. The initial particles which represent the parameter set of P and k are randomly selected from the corresponding points in 2(a) that will finally achieve stable walking. During the walking simulation of the dynamic model, the ground disturbance gradually increase. The optimization process evaluates the maximum ground disturbance of the dynamic model with certain P and k. During the walking, the selected P and k keep constant to overcome gradually increased ground disturbance. The optimization finally record the maximum ground disturbance in each generation. The fitness function is (16). Fig. 12 shows the results. It is clear that by applying the adaptive PSO, the optimization process can quickly find the optimal parameter set. It also indicated that with optimized hip actuation and ankle torque, the passive dynamic walker can achieve stable walking with no active control even if there is $-11mm$ ground disturbance.

4.2 Walking on Uneven Terrain with Control

Starting from the optimized P and k with no active control during walking simulations, we add the control scheme shown in Fig. 3 to the walking motion.



(a)

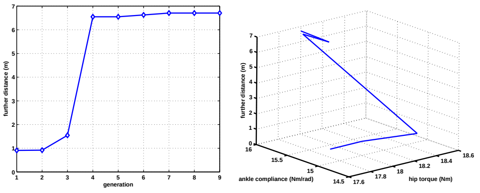
(b) (c)

Figure 5: Adaptive locomotion with active control on uneven terrain. This result is obtained every 10 frames during a continuous walking. (a) is the stick diagram. (b) and (c) are the angular trajectories of the thigh and shank respectively.

A preset ground disturbance occurs at known time during the level walking. The active control counter determines the times of active control to change P and k. In this simulation, the ground disturbance varies from $-15mm$ to $-25mm$. The fitness function of the optimization is (17). The counter first makes the optimization simulator to actively change P and k once when ground disturbance occurs. There was no optimized set of P and k that can overcome the $-25mm$ step. Then the counter makes the simulator to actively change P and k twice when ground disturbance occurs. The walker successfully achieved stable walking with $-25mm$ disturbance (see stick diagram shown in Fig. 5(a)). Fig. 5(b) and (c) show the trajectories of hip and knee during the adaptive walking.

Note that the cyclic walking was initially actuated by a relatively small hip torque. After one time of active control (varying P and k), the hip torque increased to move the swing leg more forward and prevent tripping. Since there was a second time of active control, the trajectories of the swing thigh and the swing shank transited to bigger limit cycles. Such optimized P and k finally stabilized the walking motion after a step down occurred.

Fig. 6 shows the optimization process of the hip actuation and ankle compliance during the walking with two times of active control. We set that if the walker can walk stably for enough time after the ground disturbance, the walking motion is adaptive on the uneven ground. Specifically, the distance is the product of walking speed times 25 seconds. Fig. 6(a) demonstrates the results of further distance after ground disturbance each generation. After four gen-



(a) (b)

Figure 6: Optimization of hip actuation and ankle compliance. Both *P* and *k* vary during the continuous walking. (a) is the results of further distance after ground disturbance each generation. (b) is the results of selecting hip actuation and ankle compliance during optimization.

erations, the walker can find optimized parameters to overcome the step. Fig. 6(b) shows the process of selecting *P* and *k* during the optimization. The initial hip torque is $5.5130Nm$ and ankle compliance is $12.2218Nm/rad$. The two times of variance of *P* and *k* are ($18.0000Nm$, $16.0000Nm/rad$ and ($5.5000Nm$, $15.4228Nm/rad$) respectively.

Though there is no complex learning algorithm in the control scheme, the walker can perform better terrain adaptability comparing with other simulation results (e.g. (Schuitema et al., 2005), (Wang et al., 2006)).

4.3 Real Robot Experiments

To study natural and energy-efficient bipedal walking, we designed and constructed a bipedal robot prototype, $1.2m$ in height and $20kg$ in weight. With the bisecting hip mechanism similar to (Wisse et al., 2007), the prototype has five Degrees of Freedom (DOFs). Two commercial motors are used in the hip joints to perform hip actuation. Each leg consists of a thigh and a shank interconnected through a passive knee joint that has a locking mechanism. Springs are installed between the foot and the plate that is pushing the leg up while it is rotated around the ankle. To prevent foot-scuffing, we modified the foot with arc in the front and the back-end. Specific mechanical parameters are shown in Table 1. In the experiment, by using the proposed method, the robot tried to walk on natural ground outdoor. The natural ground is not a strict continuous level floor, where random irregularity of the ground and slight slippery occurred. Fig. 7 shows the result.

The robot can achieve three-dimensional stable walking on natural ground with more than 10 steps. Comparing to the results of terrain adaptability of other real robot experiments (e.g. two-dimensional walkers (Wisse et al., 2005), (Ueno et al., 2006)), the successful three-dimensional walking of the robot prototype shows that the quasi-passive dynamic walker with optimized hip actuation and ankle compliance can perform stable walking with larger ground disturbance.

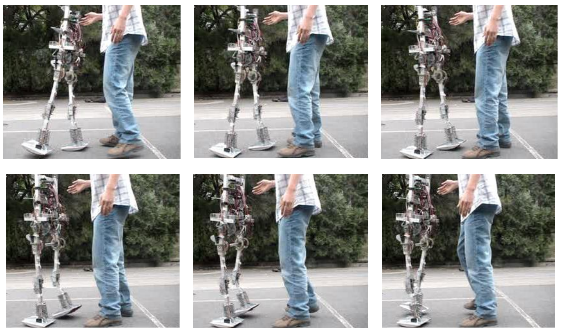


Figure 7: A sequence of photos captured during autonomous walking of the robot prototype on natural ground.

5 CONCLUSIONS

In this paper, we have investigated how to control passivity-based bipedal walkers to achieve stable locomotion with terrain adaptability. Satisfactory results of simulations and real robot experiments indicated that having the fixed mechanical parameters during walking, the passivity-based walker can walk on uneven terrain with larger ground disturbance by optimized hip actuation and ankle compliance in view of walking stability and efficiency. In the future, more real robot experiments will be continued to overcome more complex ground disturbance.

ACKNOWLEDGEMENTS

The authors would like to thank M. Wisse for sharing the simulation files of the simplest walking model. This work was supported by the National Natural Science Foundation of China (No. 60774089), National High Technology Research and Development Program of China (863 Program) (No. 2006AA04Z258) and 985 Project of Peking University.

REFERENCES

S. Collins, M. Wisse, A. Ruina, A three-dimensional passive-dynamic walking robot with two legs and knees, *International Journal of Robotics Research*, vol. 20, pp. 607-615, 2001.

S. Collins, A. Ruina, R. Tedrake, M. Wisse, Efficient bipedal robots based on passive-dynamic walkers, *Science*, vol. 307, pp. 1082-1085, 2005.

R. Eberhart, J. Kennedy, Particle swarm optimization, *Proc. of the IEEE Conf. on Neural Network*, 1995, pp. 1942-1948.

T. Fukunaga, K. Kubo, Y. Kawakami, S. Fukashiro, H. Kanehisa, C. N. Maganaris, In vivo behaviour of human muscle tendon during walking, *Proc. Biol. Sci.*, vol. 268, pp. 229-233, 2001.

M. Garcia, A. Chatterjee, A. Ruina, M. Coleman, The simplest walking model: stability, complexity, and scaling, *ASME Journal Biomechanical Engineering*, vol. 120: pp. 281–288, 1998.

Y. Guan, E. S. Neo, K. Yokoi, and K. Tanie, Stepping over obstacles with humanoid robots, *IEEE Transactions on Robotics*, **22**(5), pp. 958-973, 2006.

Honda Motor Co., Ltd. New asimo - running at 6km/h, http://world.honda.com/HDTV/ASIMO/New-ASIMO-run-6kmh/, 2005.

M. Kwan, M. Hubbard, Optimal foot shape for a passive dynamic biped, *Journal of Theoretical Biology*, vol. 248, pp. 331-339, 2007.

A. D. Kuo, Energetics of actively powered locomotion using the simplest walking model, *ASME Journal Biomechanical Engineering*, vol. 124, pp. 113-120, 2002.

M. J. Kurz, N. Stergiou, An artificial neural network that utilizes hip joint actuations to control bifurcations and chaos in a passive dynamic bipedal walking model, *Biol. Cybern.*, vol. 93, pp. 213-221, 2005.

T. McGeer, Passive dynamic walking, *International Journal of Robotics Research*, vol. 9, pp. 68-82, 1990.

P. Michel, J. Chestnutt, S. Kagami, K. Nishiwaki, J. Kuffner, and T. Kanade, GPU-accelerated real-time 3D tracking for humanoid locomotion and stair climbing, *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2007, pp. 463–469.

A. Ruina, J. E. A. Bertram, M. Srinivasan, A collisional model of the energetic cost of support work qualitatively explains leg sequencing in walking and galloping, pseudo-elastic leg behavior in running and the walk-to-run transition, *Journal of Theoretical Biology*, vol. 237, no. 2, pp. 170-192, 2005.

E. Schuitema, D. Hobbelen, P. Jonker, M. Wisse, J. Karssen, Using a controller based on reinforcement learning for a passive dynamic walking robot, *Proc. of the IEEE-RAS Int. Conf. on Humanoid Robots*, 2005, pp. 232-237.

R. Smith, U. Rattanaprasert, and N. O'Dwyer, Coordination of the ankle joint complex during walking, *Huamn Movement Science*, vol. 20, pp. 447-460, 2001.

R. Tedrake, T. W. Zhang, M. F. Fong, H. S. Seung, Actuating a Simple 3D Passive Dynamic Walker. In *Proc. IEEE Int. Conf. Robotics and Automation*, 2004, pp. 4656-4661.

T. Ueno, Y. Nakamura, T. Takuma, K. Hosoda, T. Shibata and S. Ishii, Fast and stable learning of quasi-passive dynamic walking by an unstable biped robot based on off-policy natural actor-critic, *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 5226-5231.

S. Wang, J. Braaksma, R. Babuška, D. Hobbelen, Reinforcement learning control for biped robot walking on uneven surfaces, *Proc. of the 2006 International Joint Conference on Neural Networks*, 2006, pp. 4173-4178.

Q. Wang, J. Zhu, L. Wang, Passivity-based three-dimensional bipedal robot with compliant legs, *Proc. of the SICE Annual Conference*, 2008.

M. Wisse, A. L. Schwab, F. C. T. Van der Helm, Passive dynamic walking model with upper body, *Robotica*, vol. 22, pp. 681-688, 2004.

M. Wisse, A. L. Schwab, R. Q. van der Linde, F. C. T. van der Helm, How to keep from falling forward: elementary swing leg action for passive dynamic walkers, *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 393-401, 2005.

M. Wisse, G. Feliksdal, J. van Frankenhuyzen, B. Moyer, Passive-based walking robot - Denise, a simple, efficient, and lightweight biped, *IEEE Robotics and Automation Magazine*, vol. 14, no. 2, pp. 52-62, 2007.

J. Yamaguchi and A. Takanishi, Development of a leg part of a humanoid robot - development of a biped walking robot adapting to the humans' normal living floor, *Autonomous Robots* 4(4), pp. 369-385, 1997.

AUTOMATIC APPROACH FOR RECTIFYING BUILDING FACADES FROM A SINGLE UNCALIBRATED IMAGE

Wenting Duan and Nigel M. Allinson
The Department of Electronic and Electrical Engineering, The University of Sheffield
Mappin Street, Sheffield, U.K.
{elq06wd, n.allinson}@sheffield.ac.uk

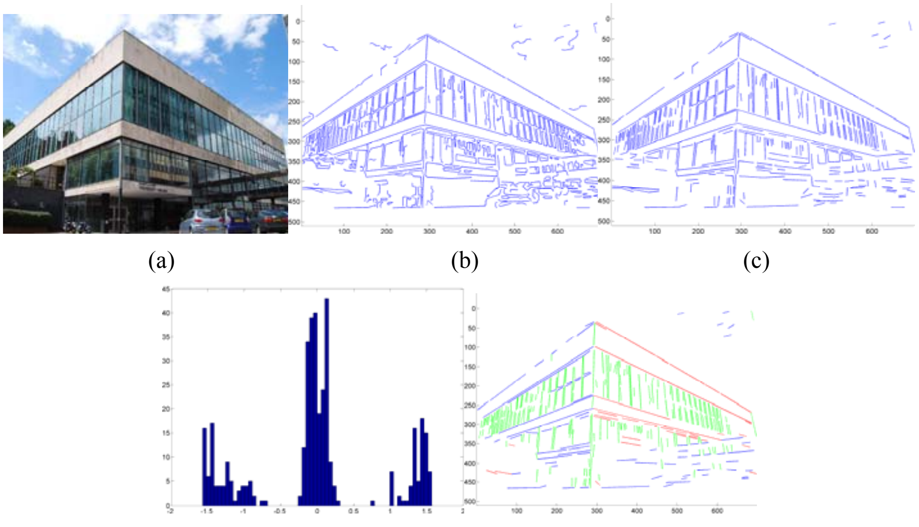
Keywords: Facade rectification, Vanishing point estimation, Line grouping, Building recognition.

Abstract: We describe a robust method for automatically rectifying the main facades of buildings from single images taken from short to medium distances. This utility is an important step in building recognition, photogrammetry and other 3D reconstruction applications. Our main contribution lies in a refinement technique for vanishing point estimation and building line grouping, since both significantly affect the location and warping of building facades. The method has been shown to work successfully on 96% of images from the Zubud-Zurich building database where images frequently contain occlusions, different illumination conditions and wide variations in viewpoint.

1 INTRODUCTION

The rectification of main building facades to their fronto-parallel view is of importance in building recognition, photogrammetry and other 3D reconstruction applications (Wang et al., 2005). It can simplify the extraction of metric information and recover the canonical shape of a building because the metric rectification allows the scene to be warped-back using a similarity transformation. In other words, the rectified view is almost free from perspective distortion. It should be noted that the rectification problem addressed here is different from image rectification for stereo vision, where the purpose is to match the epipolar projections of image pairs (Hartley, 1999). How to rectify a single uncalibrated image is a different challenge; and various approaches having been proposed and studied.

As pointed out by Menudet et al. (2008), “camera self-calibration is intrinsically related to metric reconstruction”. Therefore, an important factor for rectification lies in obtaining accurate calibration parameters and inclusion of appropriate scene constraints. Menudet et al. (2008) described a new way of decomposing the scene-to-image homography, which allows a cost function to assess how close the rectification is to similarity. However, to obtain the calibration parameters, at least four images of the same scene were required. Using only a single image of a particular scene, Liebowits and Zisserman (1998) utilised some geometric constraints such as equal angles for rectification. Chen and Ip (2005) achieved rectification by using the vanishing line and an arbitrary circle extracted from the image to estimate the image of the absolute conic (IAC). In the context of rectifying building images, reliable geometric features such as parallel lines and orthogonal angles can be used as scene constraints (Hu, Sawyer and Herve, 2006; Robertson and Cipolla, 2004; David, 2008; Košecká and Zhang, 2005). The estimation of the vanishing line is a major technique to recover images from perspective distortion. Hence, improving the accuracy and efficiency of computing these vanishing points is of foremost interest. Košecká and Zhang (2002) proposed a technique of applying the EM algorithm to detect vanishing points for images taken in man-made environments. The method achieved good accuracy with vanishing points being detected, on average, within 5 pixels of their true position. However, for building facade rectification, the following factors can adversely affect the success rate of detecting vanishing points. Firstly the images of the building can be taken in different illumination conditions and from different viewpoints. Secondly occlusion and scene clutter can obscure the building image. Finally, not all buildings have facades that are orthogonal to each other. These issues have not



(a) (b) (c)

(d) (e)

Figure 1: (a) Original Image (700×468 pixels) – Main Library of Sheffield University; (b) Line segments fitted to the connected edge points; (c) Line segments with length > 15 selected; (d) Histogram of line segment directions; (e) Separated groups of line segments associated with the three principal directions.

been considered by existing methods for building rectification, and for this reason it is desirable to develop more robust methods that can handle these potential problems.

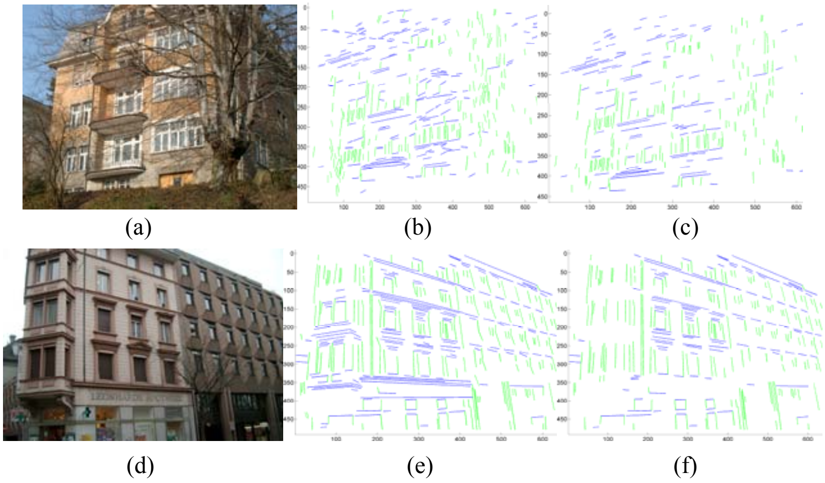
In this paper, we first present a method based on Expectation-Maximisation (EM) algorithm for estimating the vanishing points of building images. Then, we show how to use the appropriate scene constraints appeared in the image to enable automatic rectification of the main building facades. The approach is described in Section 2. In Section 3, the results are presented and compared to Košecká and Zhang's work (2002). Finally, we draw some conclusions.

2 METHOD

2.1 Line Segments Detection and Initial Grouping

Lines, derived from local intensity edges, in building images contain significant and stable geometric information because the majority are aligned to the three principal axes. These three axes are associated with the 3D orthogonal real-world axes. Under perspective transformation, the parallel lines of buildings intersect at vanishing points in the image (though the actual vanishing points may be outside the area of the captured scene). Hence, first of all, we need to find those groups of lines that are associated with these vanishing directions. A conventional Canny edge detector was used to find edge strength and orientation followed by non-maximum suppression. Hysteresis thresholding was then used to further refine the recovered edges. We applied the edge-linking function (Kovesi, 2000-2006) to the detected edges to label connected edge points. The linked points with a length under 15 pixels were discarded since these lengths were determined experimentally to be inconsequential for our image sizes (typically 700×468 pixels). A line-fitting scheme was then utilised to form straight line segments from these linked edges. At this stage, a line segment list was produced, which contains the end point coordinates of all the computed line segments in the image coordinate frame. A typical example is shown in Fig. 1(b).

In Fig. 1(b), we can easily see that most short line segments belong to the background or general scene clutter. The many short ones belonging to the building are also not reliable. Hence, the length of each segment was calculated and again ones longer than 15 pixels were selected. This small step also enables us to roughly segment the building region from the whole scene (Fig. 1(c)). The directions of all the lines were calculated in order to compute the histogram shown in Fig. 1(d). The top peaks which are at least five bins apart were selected after curve fitting to the histogram. The lines which have orientation within the range of +/-π/8 around a particular peak were included in the same group.


(a) (b) (c)

(d) (e) (f)

Figure 2: (a) Building with occlusion; (b) Detected and initially grouped two sets of lines for image (a); (c) Refined line groups for (a); (d) Building with confusing line directions; (e) Two initial groups of lines for image (d) – left side of the image contain lines that do not belong to the expected vanishing direction; (f) Refined line groups of (d).

The resultant three main groups are shown in Fig. 1(e) as separate colours.

For each group of line segments, we can now compute their initial vanishing point. In Fig. 1(e), each line segment is plotted by connecting their two end points $\mathbf{x_1}$ and $\mathbf{x_2}$**.** In a homogeneous form,
$\mathbf{x_1}$ = (x_1,y_1,1) and $\mathbf{x_2}$ = (x_2,y_2,1). Under the 2D projective plane, the homogeneous line representation is obtained by**:**

$$\mathbf{l} = \mathbf{x_1} \times \mathbf{x_2} \tag{1}$$

As mentioned above, under perspective transformations, parallel lines in the real-world coordinate frame intersect at vanishing points in the image plane. The two lines $\mathbf{l_1}$ and $\mathbf{l_2}$ intersect at the point $\mathbf{v} = \mathbf{l_1} \times \mathbf{l_2}$. Alternatively, the relationship between vanishing points and their associated lines can be expressed as $\mathbf{v^T l} = 0$. However, with so many pairs of lines available in each principal axis, we can produce many differing vanishing points. This requires us to solve the linear least square estimation problem:

$$\min_v \sum_{i=1}^{n} \left(l_i^T v\right)^2 \tag{2}$$

where n is the number of lines. This formula (2) can be written as

$$\min_{\mathbf{v}} \|\mathbf{Av}\|^2 \tag{3}$$

The rows of matrix **A** are the grouped lines with the same vanishing direction.

Before solving the linear least square estimation problem, we need to normalise the image end-point coordinates since we are dealing with the case of an uncalibrated camera. More detailed information of normalisation can be found in Košecká and Zhang (2002). The initial vanishing point for each group was calculated by the closed form solution of (3), where the estimation of **v** was the eigenvector associated with the smallest eigenvalue of $\mathbf{A^T A}$. The initial grouping of lines and estimated vanishing points are accurate enough for the example image in Fig. 1(a). However, for images with occlusions or false groupings such as in Fig. 2(a) and (b), further refinement is necessary. For example, Fig. 2(a) shows a building with some occlusions. Its initial grouped lines (as shown in Fig. 2(b)) causes large errors in vanishing point detection. The building in Fig. 2(d) also contains lines that do not belong to the dominant vanishing directions but are still grouped.

2.2 Further Refinement of Vanishing Points Locations based on EM Algorithm

The refinement method is based on the Expectation Maximisation (EM) algorithm. We first compute the likelihood of line segments $\boldsymbol{l_i}$ belonging to each of the initially estimated vanishing points $\boldsymbol{v_k}$ by the formula:

$$p(l_i \mid v_k) \propto \exp\left(\frac{-\left(l_i^T v_k\right)^2}{2\sigma_1^2}\right) \tag{4}$$

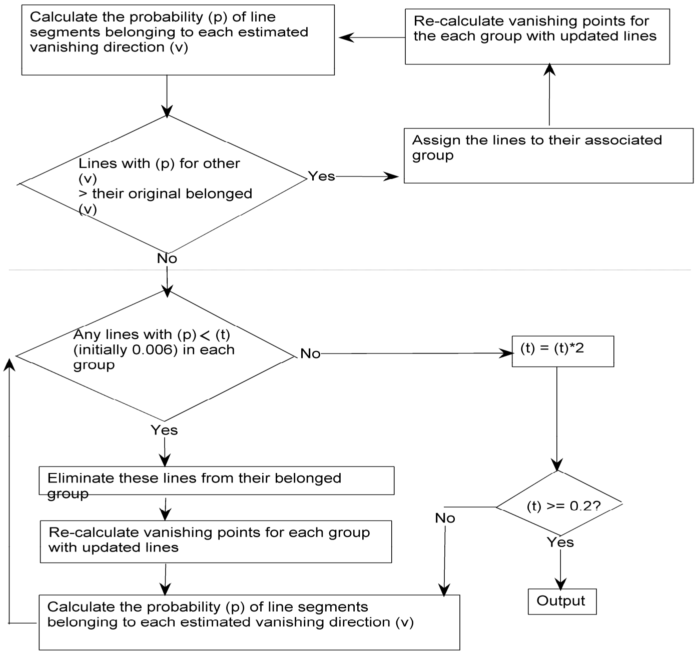

Calculate the probability (p) of line segments belonging to each estimated vanishing direction (v)
Re-calculate vanishing points for the each group with updated lines
Lines with (p) for other (v) > their original belonged (v)
Yes
Assign the lines to their associated group
No
Any lines with (p) < (t) (initially 0.006) in each group
No
(t) = (t)*2
Yes
Eliminate these lines from their belonged group
Re-calculate vanishing points for each group with updated lines
(t) >= 0.2?
No
Yes
Output
Calculate the probability (p) of line segments belonging to each estimated vanishing direction (v)

Figure 3: Flow chart of the refining process.

The upper half of the flow chart - Fig. 3 is mainly used for re-grouping lines and combining similar vanishing directions. The probabilities for each line corresponded to every vanishing direction were compared. The updated line groups were passed to the lower half of the algorithm when no lines are found to belong to other directions. The lower half eliminates lines with low probability for the direction they belong to so producing more accurate estimates of vanishing point locations. The iteration stops when line probabilities for each group are all above 0.1. In our experiments, t = 0.1 normally is sufficient to give an accurate vanishing point. The effect of this refining process for Fig. 2(a) and (d) is shown in Fig.2(c) and (f).

2.3 Automatic Rectification of Main Building Facades

To automatically warp an image's main building facades to the fronto-parallel view, we have to use the geometric information provided by the image, no external interaction should be required during the processing. We followed the approach described by Liebowitz and Zisserman (1998) as well as Hartley and Zisserman （2003). Here, we briefly summarize the method. The homography $\boldsymbol{H}$ which relates the points $\boldsymbol{x}$ in the image plane to $\boldsymbol{x'}$ （homogeneous 3-vector ） in the real-world plane can be decomposed into three transformation matrices: **H = Similarity * Affine * Projective**.

The first transformation is a pure projective transformation obtained with the vanishing line of the plane:

$$P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l_1 & l_2 & l_3 \end{bmatrix} \tag{5}$$

where l_1, l_2 and l_3 are the vector elements of the vanishing line $\boldsymbol{l}_\infty$. Since the vanishing line is computed by the two vanishing points from their corresponding facade, the vanishing point corresponds to the vertical lines of the building normally need to be used twice. For images with three vanishing points, we decided the vertical vanishing point by exploiting the location of the coordinates of all vanishing points with respect to the image's principal point.

The affine transformation which enables the recovery of metric geometry is expressed as:

$$A = \begin{pmatrix} \frac{1}{\beta} & -\frac{\alpha}{\beta} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{6}$$

where α and β are the two parameters involved in the transformation of circular points from the metric plane to the affine plane. When the metric geometry is restored it includes angles and length ratios of

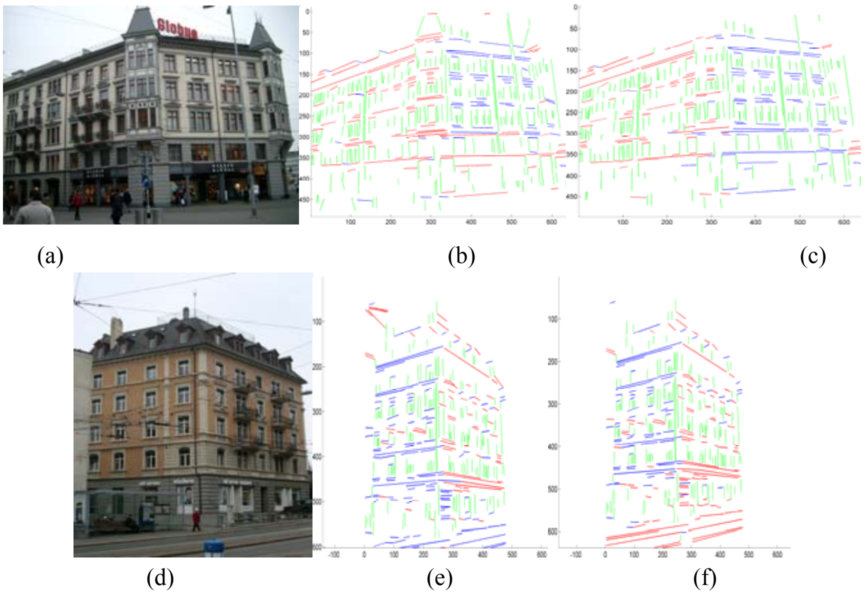


Figure 4: (a) Building with facades not aligned with the three orthogonal axes; (b) Initial grouping for image (a); (c) Refined line groups for (b); (d) Image taken from a critical viewpoint; (e) Plenty of lines grouped falsely; (f) Refined line groups for (e).

non-parallel line segments. In order to determine *α* and *β*, two constraints for non-parallel line sets must be supplied. The constraints can be obtained from "a known angle between lines; Equality of two (unknown) angles; and a known length ratio" (Liebowitz and Zisserman, 1998). Each constraint produces a circle in the complex plane with α and β indeed its real and imaginary components. The value of α and β can be found at the intersection points of two circles. However, for the problem of building facade rectification, there is only one constraint that the detected line segments provide with high confidence — the right angle between the two sets of lines. Therefore, the parameter *β* is assumed to be 1. This assumption is based on the fact that under affine transformations, the circular points with coordinates $(1,\pm i,0)^T$ in the metric plane are mapped to $(\alpha \mp i\beta,1,0)^T$. If no affine distortion occurred, the value of *α* and *β* is $(0,\ 1)^T$.

The last similarity transformation matrix is in the form of:

$$s = \begin{pmatrix} sR & t \\ 0^T & 1 \end{pmatrix} \text{ and } R = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \quad (7)$$

This final stage is used to: (i) adjust the image centre so that no coordinates of the image points has a negative value; (ii) rotate the line sets in order to make sure the majority of the lines in each group are aligned with the x and y axis directions; and (iii) scale the image up or down if the warped image size exceeds our desired value.

3 DISCUSSION

The approach proposed for building facade rectification was tested on the buildings images from the Zubud-Zurich buildings database (Shao and Gool, 2003). The method managed to rectify 96% of all the images tested. From this test, we found that the key to properly warp the building facades lies in the accuracy of the vanishing points and grouping of real-world parallel lines. The proposed refinement method was compared with Košecká and Zhang's (2002) work on vanishing points detection. Our experiment shows following improvement in the context of rectifying building images:

(1) Accuracy of Estimated Vanishing Points. Instead of assigning probability weights to each line for the Maximisation step described in [11], the lines with very low probability to the vanishing point of the associated group are eliminated or assigned to other groups. Therefore, lines which could degrade the estimate of vanishing point are reduced. From our experiments, the average deviation of vanishing points from its manually measured true position is five pixels, (the true position was decided by using

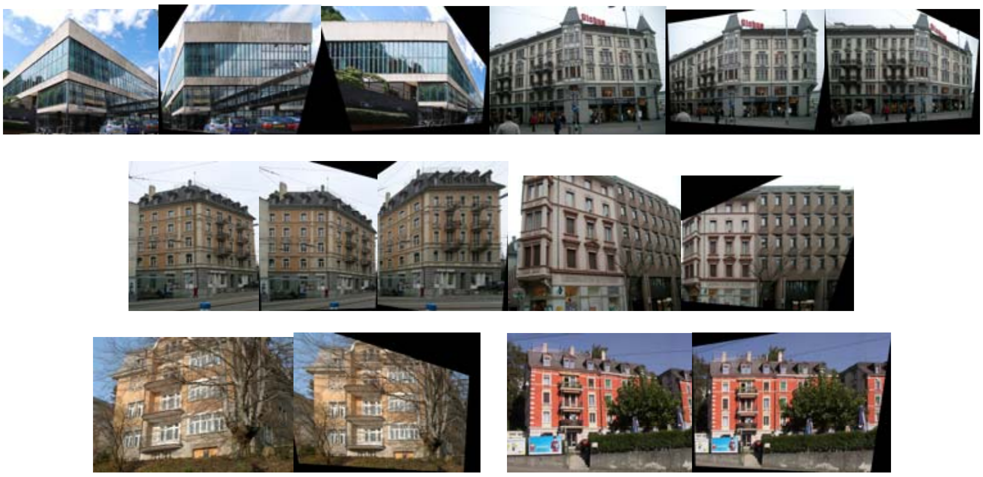


Figure 5: Some example images with their associated warped facades.

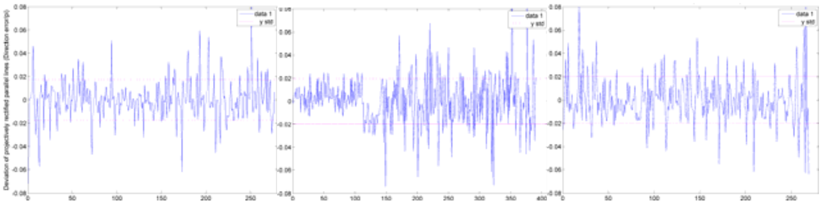


Figure 6: Deviation errors from being parallel of the projective warped lines (for first three building examples in Fig.5) – assessing vanishing point accuracy in each line set.

ruler to extend major building lines and locating the intersection).

(2) Better Grouping of Lines. Buildings can have facades which do not necessarily align with the three orthogonal axes as shown in Fig. 4(a). Images of buildings can also be taken from a critical viewpoint where false initial groupings occur (Fig. 4(d)). The problem of false grouping can also easily occur when the vanishing points' initial position is decided by the intersection of lines with similar orientations. The refinement method solved this by iteratively adjusting the position of the vanishing points and line groupings.

(3) Adaptable to Occlusion, Illumination and Viewpoint Change. These factors need to be considered when dealing with building images. Fig. 5 shows some of the rectified building facades using the proposed method that have been successfully adapted. At the rectification stage, the accuracy of each vanishing point obtained with the final grouped line segments in each vanishing direction set can be also assessed by investigating parallel lines after applying the projective transformation **P**. In theory, the projective matrix **P** can recover the lines to affinity. This means that the line direction in each group should be the same. From the experiment, the average deviation error was 1.8%. Fig. 6 shows the plots of parallel deviation errors for the first three building example of Fig. 5. In addition, instead of directly applying computed projective, affine and similarity transformations to the original image, the three-stage transformations were only applied to grouped line sets. After the final transformation, three least-deviating (from parallel to x or y axis) line segments were selected for final image registration in order to reduce rectification distortion introduced by lines with large deviation errors.

4 CONCLUSIONS

In conclusion, our approach for building facade rectification is generally robust to occlusions, different illuminations, wide changes in viewpoint and different camera settings. The method could be improved further by analysing the peaks detected at the stage of curve fitting. For example, instead of selecting the highest two or three peaks for grouping, minor peaks could also be included. Groups with similar initial estimates could be combined at the refinement stage. This kind of improvement can also enable rectification of a collection of buildings appeared in a single image.

REFERENCES

Chen, Y & Ip, H, H, S 2005, ‘Planar metric rectification by algebraically estimating the image of the absolute conic’, In *Proc IEEE conf. on Pattern Recognition*, vol. 38, pp. 1117-1120.

David, P 2008, ‘Detection of building facades in urban environments’, In *Proc. SPIE conf. on Visual Information Processing XV!!,* vol. 6978, pp. 9780-9780.

Hartley, R, I 1999. ‘Theory and practice of projective rectification’, *International Journal of Computer Vision*, vol. 35, pp. 115-127.

Hu, J, Sawyer, J & Herve, J, Y 2006, ‘Building detection and recognition for an automated tour guide’, In *Proc. IEEE Conf. on Systems, Man and Cybernetics*, vol. 1, pp. 283-289.

Košecká, J & Zhang, W 2002, ‘Video compass’, In *Computer Vision — ECCV,* vol. 2353, pp. 29-32.

Košecká, J & Zhang, W 2005, ‘Extraction, matching, and pose recovery based on dominant rectangular structures’, *Computer Vision and Image Understanding*, vol. 100, pp. 274-293.

Liebowitz, D & Zisserman, A 1998, ‘Metric rectification for perspective images of planes’, In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 482-488.

Menudet, J, F, Becker, J, M, Fournel, T & Mennessier, C 2008, ‘Plane-based camera self-calibration by metric rectification of images’, *Image and Vision Computing*, vol. 26, pp. 913-934.

Robertson, D & Cipolla, R 2004. ‘An image-based system for urban navigation’, In *Proc. British Machine Vision Conference*.

Wang, G, Hu, Z, Wu, F & Tsui, H, T 2005, ‘Single view metrology from scene constraints’, *Image and Vision Computing*, vol. 23, pp. 831-840.

Kovesi, P 2000-2006, http://www.csse.uwa.edu.au/~pk/research/matlabfns/

Shao, T, S, H & Gool, L, V (2003), ‘Zubud-zurich buildings database for image based recognition’, *Technical report No. 260,* Swiss Federal Institute of Technology, http://www.vision.ee.ethz.ch/showroom/zubud/

Hartley, R, & Zisserman, A 2003, *Multiple View Geometry in Computer Vision*, Cambridge University Press, Second Edition, Chapter 8.

A DECENTRALIZED COLLISION AVOIDANCE ALGORITHM FOR MULTI-ROBOTS NAVIGATION

Michael Defoort, Arnaud Doniec and Noury Bouraqadi
Ecole de Mines de Douai, 941 rue Charles Bourseul, BP 10838, 59508 Douai, France
defoort@ensm-douai.fr

Keywords: Decentralized intelligence, Real-time path planning, Collision avoidance, Receding horizon, Nonholonomic mobile robots.

Abstract: This paper presents a distributed strategy for the navigation of multiple autonomous robots. The proposed planning scheme combines a decentralized receding horizon motion planner, in which each robot computes its own planned trajectory locally, with a fast navigation controller based on artificial potential fields and sliding mode control technique. This algorithm solves the collision avoidance problem. It explicitly accounts for computation time and is decentralized, making it suited for real time applications. Simulation studies are provided in order to show the effectiveness of the proposed approach.

1 INTRODUCTION

The research effort in multi-robots systems (MRS) relies on the fact that multiple robots have the possibility to perform a mission more efficiently than a single robot. Among all the topics of study in this field, the issue of conflict resolution becomes an increasingly important point. Many cooperative tasks such as surveillance, search, rescue or area data acquisition need the robots to autonomously navigate without collision.

Solving conflicts in MRS consists in introducing some coordination mechanisms in order to give a coherence between the robot acts (Kuchar and Yang, 2000). For motion planning, three coordination mechanisms are identified:

- *The Coordination by Adjustment*, where each robot adapts its behavior to achieve a common objective (Tomlin et al., 1998). However, most of the planning algorithms are centralized, which often limit their applicability in real systems.
- *The Coordination by Leadership (or supervision)* where a hierarchical relationship exists between robots (Das et al., 2002). Such an approach is easy to implement. However, due to the lack of an explicit feedback from the followers to the leader, the collision avoidance cannot be guaranteed if followers are perturbed (during obstacle avoidance for instance). Another disadvantage is that the leader is a single point of failure.
- *The Standardization* where procedures are predefined to solve some particular interaction cases (Pallatino et al., 2007). While this approach can lead to straightforward proofs, it also tends to be less flexible with respect to changing conditions.

Here, the problem of interest is the decentralized navigation for autonomous robots through a coordination by adjustment. Each vehicle is modeled as an unicycle with a limited sensing range in order to capture the essential properties of a wide range of vehicles. They are dynamically decoupled but have common constraints that make some conflicts. Indeed, each robot has to avoid collision with the other entities. Furthermore, the proposed framework allows moving (and static) obstacles to be avoided since they can be modeled as non cooperative entities.

Motion planning consists in generating a collision-free trajectory from the initial to the final desired positions for a robot. Since the environment is partially known and further explored in real time, the computation of complete trajectories from start until finish must be avoided. Therefore, the trajectories have to be computed gradually over time while the mission unfolds. It can be accomplished using an online receding horizon planner (Mayne et al., 2000), in which partial trajectories from an initial state toward the goal are computed by solving optimal control problems over a limited horizon.

Two strategies for motion planning in MRS are the centralized and decentralized (distributed) approaches. Although the centralized one has been used in different studies (see (Dunbar and Murray, 2002) for instance), its computation time which scales exponentially with the number of robots, its communication requirement and its lack of security make it prohibitive. To overcome these limitations, one can use a distributed strategy which results in behaviors closed to what is obtained with a centralized approach. Recently, some decentralized receding horizon planners have been proposed. In (Dunbar and Murray, 2006), a distributed solution is provided for the rigid formation stabilization problem.

In (Kuwata et al., 2006), the navigation problem is solved through a coordination by leadership. Indeed, the robots update their trajectory sequentially. In (Defoort et al., 2007), a decentralized algorithm based on a coordination by adjustment is proposed to solve the navigation problem for MRS. However, the large amount of information exchanged between robots and the addition of several constraints make this strategy prohibitive when the number of vehicles increases.

One of other collision avoidance algorithm is potential field method, where an artificial potential function treats each robot as a charged particle that repels all the other entities (Latombe, 1991; De-Gennaro and Jadbabaie, 2006). However, most of them are only based on relative position information and do not consider coordination between cooperative robots.

In this paper, we proposed a practical decentralized scheme, based on a coordination by adjustment, for real time navigation of large-scale MRS. As illustrated in Fig. 1, the scheme consists of two parallel processes:

- a distributed receding horizon planner, in which each robot computes its own planned trajectory locally, for the coordination between cooperative robots,
- a reactive approach, which combines artificial potential fields and sliding mode control technique, for simultaneously tracking the planned trajectory while avoiding collision with unexpected entities (i.e. non cooperative entities).

The main advantages of the proposed strategy, especially for large-scale MRS, are the small amount of information exchanged between cooperative robots and the robustness.

The outline of this paper is as follows. In Section 2, the problem setup is described. In Section 3, the navigation algorithm is presented. Finally, numerical results illustrate the effectiveness of the strategy.

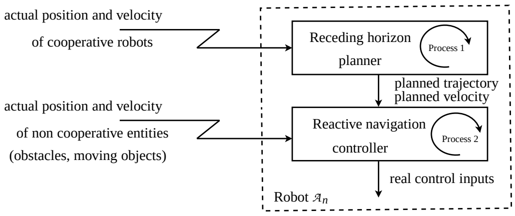


Figure 1: Proposed navigation algorithm.

2 PROBLEM STATEMENT

2.1 Dynamic Model of the Robots

Each robot $\mathcal{A}_n$ ($n \in (1,\ldots,N)$ with $N \in \mathbb{N}$), shown in Fig. 2, is of unicycle-type. Its two fixed driving wheels of radius r_n, separated by $2\rho_n$, are independently controlled by two actuators (DC motors) and the passive wheel prevents the robot from tipping over as it moves on a plane. Its configuration is given by:

$$\boldsymbol{\eta}_n = [x_n, y_n, \theta_n]^T$$

where (x_n, y_n) is the position of its mass center C_n and θ_n is its orientation in the global frame.

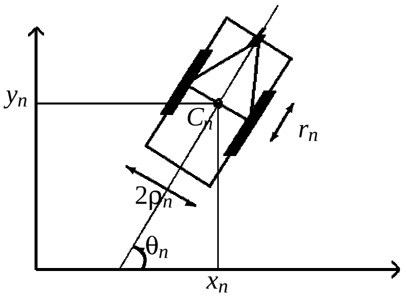


Figure 2: Unicycle-type robot.

The dynamic model of robot $\mathcal{A}_n$ is given as in (Do et al., 2004):

$$\begin{aligned} \dot{\boldsymbol{\eta}}_n &= J(\boldsymbol{\eta}_n)\mathbf{z}_n && (1)\\ M_n\dot{\mathbf{z}}_n + D_n\mathbf{z}_n &= \boldsymbol{\tau}_n && (2) \end{aligned}$$

where

- M_n is a symmetric positive definite inertia matrix
- D_n is a symmetric damping matrix
- the transformation matrix $J(\boldsymbol{\eta}_n)$ is

$$J(\boldsymbol{\eta}_n) = \frac{r_n}{2}\begin{bmatrix} \cos\theta_n & \cos\theta_n \\ \sin\theta_n & \sin\theta_n \\ \rho_n^{-1} & -\rho_n^{-1} \end{bmatrix} \qquad (3)$$

- $\mathbf{z}_n = [z_n^r, z_n^l]^T$ where z_n^r, z_n^l are the angular velocities of the right and left wheels. The relationship between $\mathbf{z}_n$ and the linear and angular velocities, denoted v_n, w_n, is

$$\begin{bmatrix} z_n^r \\ z_n^l \end{bmatrix} = B_n \begin{bmatrix} v_n \\ w_n \end{bmatrix} \text{ with } B_n = \frac{1}{r_n} \begin{bmatrix} 1 & \rho_n \\ 1 & -\rho_n \end{bmatrix} \quad (4)$$

- $\boldsymbol{\tau}_n = [\tau_n^r, \tau_n^l]^T$ where τ_n^r, τ_n^l are the control torques applied to the wheels of the robot

Remark 1. *System (1)-(2) is flat (see (Fliess et al., 1995) for details about flatness) since all system variables can be differentially parameterized by x_n, y_n as well as a finite number of their time derivatives. For instance, θ_n, v_n and w_n can be expressed as*

$$\theta_n = \arctan \frac{\dot{y}_n}{\dot{x}_n}, \; v_n = \sqrt{\dot{x}_n^2 + \dot{y}_n^2}, \; w_n = \frac{\ddot{y}_n \dot{x}_n - \ddot{x}_n \dot{y}_n}{\dot{x}_n^2 + \dot{y}_n^2}$$

Remark 2. *Speed $\mathbf{u}_n = [\dot{x}_n, \dot{y}_n]^T$ of $\mathcal{A}_n$ is restricted to lie in a closed interval $\mathcal{S}_n$*

$$\mathcal{S}_n = \left\{ \; \mathbf{u}_n \in \mathbb{R}^2 \quad | \quad \|\mathbf{u}_n\| \leq u_{n,max} \; \right\} \quad (5)$$

2.2 Assumptions and Control Objective

Assumption 1. *The following assumptions are made:*

- *$\mathcal{A}_n$ knows its position $\mathbf{p}_n = [x_n, y_n]^T$ and its velocity $\mathbf{u}_n = [\dot{x}_n, \dot{y}_n]^T$*
- *$\mathcal{A}_n$ has a physical safety area, which is centered at C_n with a radius R_n, and has a circular communication area which is also centered at C_n with a radius $\bar{R}_n$. Note that $\bar{R}_n$ is strictly larger than $R_n + R_j$, $j \in (1, \ldots, N)$, $j \neq n$*
- *$\mathcal{A}_n$ broadcasts $(\mathbf{p}_n, \mathbf{u}_n)$ and receives $(\mathbf{p}_j, \mathbf{u}_j)$ broadcasted by other cooperative robots $\mathcal{A}_j$, in its communication area*
- *$\mathcal{A}_n$ can compute the relative position and velocity $(\mathbf{p}_{obs_i}, \mathbf{u}_{obs_i})$ of non cooperative entities within a given sensing range*
- *At the initial time $t_{ini} \geq 0$, each robot starts at a location outside of the safety areas of other entities*

The objective is to find the control inputs $\boldsymbol{\tau}_n$ for each robot $\mathcal{A}_n$ such that, under Assumption 1,

- $\mathcal{A}_n$ is stabilized toward its desired point $\mathbf{p}_{n,des}$, i.e.

$$\lim_{t \to \infty} \|\mathbf{p}_n(t) - \mathbf{p}_{n,des}\| = 0 \quad (6)$$

- collisions are avoided
- all computations are done on board in a decentralized cooperative way

Remark 3. *It should be noted that for collision avoidance, one can distinguish two kinds of entities, i.e.*

- *cooperative robots which are involved in a detected potential collision*
- *non cooperative entities which cannot cooperate in the collision avoidance process. They represent the moving objects and static obstacles.*

3 DISTRIBUTED ALGORITHM

In order to solve the multi-robots navigation problem, a decentralized algorithm combining two parallel processes is proposed. First, a receding horizon planner, in which each robot computes its own planned trajectory locally, achieves middle-term objectives, i.e. coordination between cooperative robots which are involved in a detected potential collision. Then, a reactive navigation controller is proposed to fulfill short-term objectives, i.e. trajectory tracking while taking into account non cooperative entities.

3.1 Conflicts and Collisions

Definition 1. *(conflict) A conflict occurs between two cooperative robots $\mathcal{A}_n$ and $\mathcal{A}_j$ at time $t_k \in \mathbb{R}^+$ if they are not in collision at t_k, but at some future time, a collision may occur.*

The following proposition, based on the well-known concept of velocity obstacle (Fiorini and Shiller, 1998), is useful to check the presence of conflicts.

Proposition 1. *Let us define for each pair $(\mathcal{A}_n, \mathcal{A}_j)$, the following variables depicted in Fig. 3:*

$$\begin{array}{rcl} \beta_{nj}(t_k) & = & \arg(\mathbf{u}_n(t_k) - \mathbf{u}_j(t_k)) - \arg(\mathbf{p}_j(t_k) - \mathbf{p}_n(t_k)) \\ \alpha_{nj}(t_k) & = & \arcsin\left(\frac{R_n + R_j}{\|\mathbf{p}_j(t_k) - \mathbf{p}_n(t_k)\|}\right) \end{array} \quad (7)$$

A necessary and sufficient condition for no conflict between $\mathcal{A}_n$ and $\mathcal{A}_j$ at t_k is:

$$|\beta_{nj}(t_k)| \geq \alpha_{nj}(t_k) \quad (8)$$

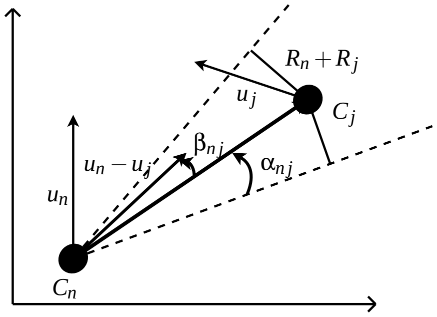


Figure 3: Velocity obstacle concept.

Definition 2. *(conflict subset) For each $\mathcal{A}_n$, the conflict subset $\mathcal{N}_n(t_k)$ at time $t_k \in \mathbb{R}^+$ is the set of all cooperative robots which are in the communication area of $\mathcal{A}_n$ and in conflict with $\mathcal{A}_n$.*

3.2 Receding Horizon Planner

The purpose of the distributed receding horizon planner is to decompose the overall problem into a family of simple receding horizon planning problems which are implemented in each robot $\mathcal{A}_n$.

In every problem, the same planning horizon $T_p \in \mathbb{R}^+$ and update period $T_c \in \mathbb{R}^+$ ($T_c < T_p$) are used. The receding horizon updates are

$$t_k = t_{ini} + (k-1)T_c, \;\; k \in \mathbb{N}^* \tag{9}$$

Remark 4. *During the initialization step, that is to say before robots move, we denote $t_0 = t_{ini}$.*

At each update t_k, robots in conflict exchange information about each others (position, velocity, ...). Then, in parallel, every robot $\mathcal{A}_n$ computes an anticipated trajectory, denoted $\widehat{\boldsymbol{p}}_j(t,t_k)$ and an anticipated velocity $\widehat{\boldsymbol{u}}_j(t,t_k)$, over the overall horizon, for all $\mathcal{A}_j$ belonging to $\mathcal{N}_n(t_k)$. These trajectories are obtained without taking the collision avoidance constraint into account. Therefore, by design, the anticipated trajectory is the same in every receding horizon planning problem in which it occurs. At last, in parallel, every robot $\mathcal{A}_n$ computes only its own planned trajectory $\boldsymbol{p}^*_n(t,t_k)$ and planned velocity $\boldsymbol{u}^*_n(t,t_k)$, over the planning horizon T_p, in order to integrate the collision avoidance between cooperative robots. From the planned trajectory and velocity associated to the planning horizon T_p, only the part which corresponds to the update horizon T_c is stored.

Remark 5. *Note that the first argument of $\boldsymbol{p}^*_n$, $\widehat{\boldsymbol{p}}_n$, $\boldsymbol{u}^*_n$ and $\widehat{\boldsymbol{u}}_n$ denotes time. The second argument is only added to distinguish at which receding horizon update the trajectory and velocity are computed.*

The collection of distributed receding horizon planning problems is formally defined as Problems 1-2 for each robot $\mathcal{A}_n$.

Problem 1. For each robot $\mathcal{A}_n$ and at any update t_k, $k \in \mathbb{N}$:
<u>Given</u>: the actual positions $\boldsymbol{p}_n(t_k)$, $\boldsymbol{p}_j(t_k)$ and the actual velocities $\boldsymbol{u}_n(t_k)$, $\boldsymbol{u}_j(t_k)$ of robot $\mathcal{A}_n$ and robots $\mathcal{A}_j$ belonging to $\mathcal{N}_n(t_k)$, respectively.
<u>Find</u>: the anticipated trajectory and velocity pairs $(\widehat{\boldsymbol{p}}_i(t,t_k), \widehat{\boldsymbol{u}}_i(t,t_k))$, $\forall i \in \{i \in \mathbb{N} \mid \mathcal{A}_i \in \mathcal{N}_n(t_k) \cup \{\mathcal{A}_n\}\}$ subject to the following constraints:

$$\begin{cases} \widehat{\boldsymbol{p}}_i(t_k,t_k) & = & \boldsymbol{p}_i(t_k) \\ \widehat{\boldsymbol{u}}_i(t_k,t_k) & = & \boldsymbol{u}_i(t_k) \\ \widehat{\boldsymbol{u}}_i(t,t_k) & \in & \mathcal{S}_i, \;\; \forall t \geq t_k \end{cases} \tag{10}$$

The anticipated trajectories are computed without taking the collision avoidance constraint into account. That is why, to integrate the path planning with local collision avoidance, the following problem is solved.

Problem 2. For each robot $\mathcal{A}_n$ and at any update t_k, $k \in \mathbb{N}$:
<u>Given</u>: the anticipated pairs $(\widehat{\boldsymbol{p}}_i(t,t_k), \widehat{\boldsymbol{u}}_i(t,t_k))$, $\forall i \in \{i \in \mathbb{N} \mid \mathcal{A}_i \in \mathcal{N}_n(t_k) \cup \{\mathcal{A}_n\}\}$.
<u>Find</u>: the planned trajectory and velocity pairs $(\boldsymbol{p}^*_n(t,t_k), \boldsymbol{u}^*_n(t,t_k))$ that minimizes

$$\int_{t_{k+1}}^{t_{k+1}+T_p} \left(a_n \sum_j \widehat{U}_{nj,rep}(t) + \|\boldsymbol{p}^*_n(t,t_k) - \widehat{\boldsymbol{p}}_n(t,t_k)\| \right) dt \tag{11}$$

subject to the following constraints:

$$\begin{cases} \boldsymbol{p}^*_n(t_{k+1},t_k) & = & \boldsymbol{p}^*_n(t_{k+1},t_{k-1}) \\ \boldsymbol{u}^*_n(t_{k+1},t_k) & = & \boldsymbol{u}^*_n(t_{k+1},t_{k-1}) \\ \boldsymbol{u}^*_n(t,t_k) & \in & \mathcal{S}_n, \; \forall t \in [t_{k+1}, t_{k+1}+T_p] \end{cases} \tag{12}$$

where

$$\begin{array}{rcl} \widehat{U}_{nj,rep}(t) & = & \begin{cases} 0 & \text{if } \widehat{\rho}_{nj}(t) \geq b_n \\ \frac{1}{2}\left(\frac{1}{\widehat{\rho}_{nj}(t)} - \frac{1}{b_n}\right)^2 & \text{else} \end{cases} \\ \widehat{\rho}_{nj}(t) & = & \|\boldsymbol{p}^*_n(t,t_k) - \widehat{\boldsymbol{p}}_j(t,t_k)\| - (R_n + R_j) \end{array} \tag{13}$$

a_n and b_n are strictly positive factors which can vary among robots to reflect differences in aggressiveness ($a_n < 1$, $b_n \ll 1$) and shyness ($a_n > 1$, $b_n \gg 1$).

One can note that the first part of cost (11) is designed to enforce the collision avoidance between cooperative robots. The cost term $\|\boldsymbol{p}^*_n(t,t_k) - \widehat{\boldsymbol{p}}_n(t,t_k)\|$ in (11) is a way of penalizing the deviation of the planned trajectory $\boldsymbol{p}^*_n(t,t_k)$ from the anticipated trajectory $\widehat{\boldsymbol{p}}_n(t,t_k)$, which is the trajectory that other robots rely on. In previous work, this term was incorporated into the decentralized receding horizon planner as a constraint (Defoort et al., 2007). The formulation presented here is an improvement over this past formulation, since the penalty yields an optimization problem that is much easier to solve.

Remark 6. *One can note that constraints (12) which guarantee the continuity of the planned trajectory and velocity need $\boldsymbol{p}^*_n(t_{k+1},t_{k-1})$ and $\boldsymbol{u}^*_n(t_{k+1},t_{k-1})$ computed in the previous step. Therefore, the proposed planner is not able to reject external disturbances or inherent discrepancies between the model and the real process. However, it takes the real time constraint into account. Indeed, each robot has a limited time to plan its trajectory. The time allocated to make its decision depends on its perception sensors, its computation delays and is less than the update period T_c (see Fig. 4).*
The discussed claim for robustness in trajectory tracking will be achieved hereafter.

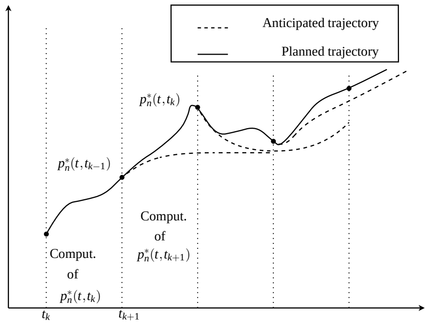


Figure 4: Implementation of the receding horizon planner.

Remark 7. *A compromise must be done between reactivity and computation time. Indeed, the planning horizon must be sufficiently small in order to have good enough results in terms of computation time. However, it must be higher than the update period to guarantee enough reactivity.*

Remark 8. *To numerically solve Problems 1-2, a nonlinear trajectory generation algorithm (Defoort et al., 2009) is applied. It is based on finding trajectory curves in a lower dimensional space and parameterizing these curves by B-splines. A constrained feasible sequential quadratic optimization algorithm is used to find the B-splines coefficients that optimize the performance objective while respecting the constraints.*

3.3 Reactive Navigation Controller

Hereafter, a reactive approach, which combines artificial potential fields and sliding mode control technique, for simultaneously tracking the planned trajectory while avoiding collision with unexpected entities (i.e. non cooperative entities), is proposed.

Since the robot dynamics (1)-(2) is of strict feedback systems (see (Krstic et al., 1995) for details about strict feedback systems) with respect to the robot linear and angular velocities (i.e. v_n and w_n), a backstepping procedure is used to design the control input $\boldsymbol{\tau}_n$. That is why the control design is divided into two main steps.

3.3.1 Step 1 based on Artificial Potential Fields

Let us introduce the following notations:

$$\left\{\begin{array}{lcl}\theta_{ne} & = & \theta_n - \gamma_{\theta_n}\\ v_{ne} & = & v_n - \gamma_{v_n}\end{array}\right. \tag{14}$$

where γ_{θ_n} and γ_{v_n} are auxiliary variables used to avoid collisions. Replacing expressions (14) into the first two equations of (1) and using (4) yield:

$$\dot{\boldsymbol{p}}_n = \begin{bmatrix}\cos\gamma_{\theta_n}\\ \sin\gamma_{\theta_n}\end{bmatrix}\gamma_{v_n} + \Delta_{1n} + \Delta_{2n} \tag{15}$$

with $\Delta_{1n} = \gamma_{v_n}\begin{bmatrix}(\cos\theta_{ne}-1)\cos\gamma_{\theta_n} - \sin\theta_{ne}\sin\gamma_{\theta_n}\\ \sin\theta_{ne}\cos\gamma_{\theta_n} + (\cos\theta_{ne}-1)\sin\gamma_{\theta_n}\end{bmatrix}$ and $\Delta_{2n} = v_{ne}\begin{bmatrix}\cos\theta_n\\ \sin\theta_n\end{bmatrix}$.

The objective is to design the auxiliary variables γ_{v_n} and γ_{θ_n} such that robot $\mathcal{A}_n$ robustly tracks its planned trajectory $\boldsymbol{p}_n^*$ while avoiding unexpected collisions. Here, artificial potential functions are used in order to design an attractive force between the robot and its planned trajectory and a repulsive force to avoid collisions.

In conventional potential field method (Latombe, 1991), the planned robot velocity $\boldsymbol{u}_n^*$ is assumed to be zero and the obstacle velocity $\boldsymbol{u}_{obs_i}$ is not considered. However, to make robot $\mathcal{A}_n$ track the planned trajectory among moving obstacles, velocities $\boldsymbol{u}_n^*$ and $\boldsymbol{u}_{obs_i}$ play key roles. This issue will be addressed by extending the results given in (Huang, 2009). Let us consider the conventional potential function (Latombe, 1991):

$$U_n = U_{n,att} + U_{n,rep} \tag{16}$$

where $U_{n,att}$ and $U_{n,rep}$ are, respectively, the attractive potential defined to track the planned trajectory $\boldsymbol{p}_n^*$ and the repulsive potential related to collision avoidance, specified as follows:

- The attractive potential is designed such that it puts penalty on the tracking error and is equal to zero when the robot is at its desired position, i.e.

$$U_{n,att} = \frac{1}{2}\|\boldsymbol{p}_n - \boldsymbol{p}_n^*\|^2 \tag{17}$$

- The repulsive potential is designed such that it equals to infinity when a collision occurs with $\mathcal{A}_n$ and decreases according to the relative distance between $\mathcal{A}_n$ and an obstacle, i.e.

$$U_{n,rep} = c_n \sum_i U_{ni,rep} \tag{18}$$

with

$$U_{ni,rep} = \left\{\begin{array}{ll} 0 & \text{if } \rho_{ni} \geq d_n\\ \frac{1}{2}\left(\frac{1}{\rho_{ni}} - \frac{1}{d_n}\right)^2 & \text{else}\end{array}\right. \tag{19}$$

where

ρ_{ni} is the minimum distance between robot $\mathcal{A}_n$ and the obstacle i. c_n and d_n are strictly positive factors which have similar properties as a_n and b_n.

Proposition 2. *If the errors θ_{ne} and v_{ne} are asymptotically stable, $\mathcal{A}_n$ robustly tracks its planned trajectory $\boldsymbol{p}_n^*$ while avoiding collisions using the auxiliary variables:*

$$\begin{array}{rl}\gamma_{v_n} = & [(\|\boldsymbol{u}_n^*\|\cos(\theta_n^* - \psi_n) - c_n\sum_i \xi_{ni}\|\boldsymbol{u}_{obs_i}\|\cos(\theta_{obs_i} - \psi_{ni})\\ & +\|\boldsymbol{p}_n - \boldsymbol{p}_n^*\|)^2 + \|\boldsymbol{u}_n^*\|^2\sin^2(\theta_n^* - \bar{\psi}_n)]^{0.5}\\ \gamma_{\theta_n} = & \bar{\psi}_n + \arcsin\left(\frac{\|\boldsymbol{u}_n^*\|\sin(\theta_n^* - \bar{\psi}_n)}{\gamma_{v_n}}\right)\end{array} \tag{20}$$

with

$$\begin{array}{lcl} \theta_n^* & = & \arg(\boldsymbol{u}_n^*) \\ \theta_{obs_i} & = & \arg(\boldsymbol{u}_{obs_i}) \\ \psi_n & = & \arg(\boldsymbol{p}_n^* - \boldsymbol{p}_n) \\ \psi_{ni} & = & \arg(\boldsymbol{p}_{obs_i} - \boldsymbol{p}_n) \\ \bar{\psi}_n & = & \arctan\left(\frac{\sin\psi_n - c_n\sum_i \xi_{ni}\sin\psi_{ni}}{\cos\psi_n - c_n\sum_i \xi_{ni}\cos\psi_{ni}}\right) \\ \xi_{ni} & = & \left\{\begin{array}{ll} 0 & if\, \rho_{ni} \geq d_n \\ \left(\frac{1}{\rho_{ni}} - \frac{1}{d_n}\right)\frac{1}{(\rho_{ni})^2}\frac{1}{\|\boldsymbol{p}_n - \boldsymbol{p}_n^*\|} & else \end{array}\right. \end{array}$$

Proof. Let us differentiate U_n with respect to time in equation (16), i.e.:

$$\dot{U}_n = \dot{U}_{n,att} + \dot{U}_{n,rep} \tag{21}$$

Substituting (20) into (21) yields after some geometric manipulations:

$$\begin{array}{lcl} \dot{U}_n & = & \|\boldsymbol{p}_n^* - \boldsymbol{p}_n\|(\|\boldsymbol{u}_n^*\|\cos(\theta_n^* - \psi_n) - \gamma_{v_n}\cos(\gamma_{\theta_n} - \psi_n) \\ & & -c_n\sum_i \xi_{ni}\left(\|\boldsymbol{u}_{obs_i}\|\cos(\theta_{obs_i} - \psi_{ni}) + \gamma_{v_n}\cos(\gamma_{\theta_n} - \psi_{ni})\right)) \\ & & +\left((\boldsymbol{p}_n - \boldsymbol{p}_n^*)^T - c_n\sum_i \xi_{ni}\frac{\|\boldsymbol{p}_n - \boldsymbol{p}_n^*\|}{\|\boldsymbol{p}_n - \boldsymbol{p}_{obs_i}\|}(\boldsymbol{p}_n - \boldsymbol{p}_{obs_i})^T\right)(\Delta_{1n} + \Delta_{2n}) \\ & = & \|\boldsymbol{p}_n^* - \boldsymbol{p}_n\|(\|\boldsymbol{u}_n^*\|\cos(\theta_n^* - \psi_n) - \gamma_{v_n}\cos(\gamma_{\theta_n} - \bar{\psi}_n) \\ & & -c_n\sum_i \xi_{ni}\left(\|\boldsymbol{u}_{obs_i}\|\cos(\theta_{obs_i} - \psi_{ni})\right)) \\ & & +\left((\boldsymbol{p}_n - \boldsymbol{p}_n^*)^T - c_n\sum_i \xi_{ni}\frac{\|\boldsymbol{p}_n - \boldsymbol{p}_n^*\|}{\|\boldsymbol{p}_n - \boldsymbol{p}_{obs_i}\|}(\boldsymbol{p}_n - \boldsymbol{p}_{obs_i})^T\right)(\Delta_{1n} + \Delta_{2n}) \\ & = & \|\boldsymbol{p}_n^* - \boldsymbol{p}_n\|(\|\boldsymbol{u}_n^*\|\cos(\theta_n^* - \psi_n) - \sqrt{\gamma_{v_n}^2 - \|\boldsymbol{u}_n^*\|^2\sin^2(\theta_n^* - \bar{\psi}_n)} \\ & & -c_n\sum_i \xi_{ni}\left(\|\boldsymbol{u}_{obs_i}\|\cos(\theta_{obs_i} - \psi_{ni})\right)) \\ & & +\left((\boldsymbol{p}_n - \boldsymbol{p}_n^*)^T - c_n\sum_i \xi_{ni}\frac{\|\boldsymbol{p}_n - \boldsymbol{p}_n^*\|}{\|\boldsymbol{p}_n - \boldsymbol{p}_{obs_i}\|}(\boldsymbol{p}_n - \boldsymbol{p}_{obs_i})^T\right)(\Delta_{1n} + \Delta_{2n}) \end{array}$$

Assuming that the errors θ_{ne} and v_{ne} are asymptotically stable (i.e. $\Delta_{1n} = \Delta_{2n} = 0$), one can get from (20):

$$\dot{U}_n \leq -\|\boldsymbol{p}_n^* - \boldsymbol{p}_n\|^2 \tag{22}$$

Since $U_n \geq 0$ and $\dot{U}_n \leq 0$, U_n is bounded. That is why $\mathcal{A}_n$ robustly tracks its planned trajectory $\boldsymbol{p}_n^*$ while avoiding collisions. □

3.3.2 Step 2 based on Sliding Mode Technique

Now, the objective is to force the motion of robot $\mathcal{A}_n$ such that the errors θ_{ne} and v_{ne} are asymptotically stable. The proposed strategy is based on the so-called second order sliding mode control (SMC) approach. The SMC methodology (Utkin et al., 1999) is chosen because it is a robust technique to control nonlinear systems operating under uncertainty conditions (Fridman and Levant, 2002). Furthermore, second order SMC can reduce the chattering phenomenon (high frequency vibrations of the controlled system which degrade the performances). Indeed, instead of influencing the first sliding variable time derivative, the signum function acts on its second time derivative. This method can also achieve a better convergence accuracy with respect to discrete sampling time than conventional SMC (see (Fridman and Levant, 2002) for a survey).

Let us apply to system (1)-(2) the following preliminary feedback:

$$\bar{\boldsymbol{\tau}}_n = (M_n B_n)^{-1}(\boldsymbol{\tau}_n - D_n \boldsymbol{z}_n) \tag{23}$$

where $\bar{\boldsymbol{\tau}}_n = [\bar{\tau}_{1n}, \bar{\tau}_{2n}]^T$ is the auxiliary control input. Thus, system (1)-(2) can be expressed as follows:

$$\dot{\boldsymbol{\eta}}_n = J(\boldsymbol{\eta}_n)B_n\begin{bmatrix} v_n \\ w_n \end{bmatrix} \tag{24}$$

$$\begin{bmatrix} \dot{v}_n \\ \dot{w}_n \end{bmatrix} = \bar{\boldsymbol{\tau}}_n \tag{25}$$

Since the relative degree of system (24)-(25) with respect to the sliding variable v_{ne} is only one, a dynamic extension is done before designing the control (see (Isidori, 1989) for further details). Thus, an integrator chain is added on the input variable $\bar{\tau}_{1n}$.

There are several algorithms to ensure the finite time stabilization of the sliding variables θ_{ne} and v_{ne} towards the origin. Among them, the sampled twisting algorithm (Fridman and Levant, 2002) has been developed for systems with relative degree two. This algorithm provides good convergence accuracy and robustness properties. It does not require the knowledge of the time derivative of the sliding variables and takes into account some practical constraints such as the sampling of the measurement and the control.

Proposition 3. *Consider system (1)-(2). The errors θ_{ne} and v_{ne} are stable in finite time under the nonlinear controller defined in (23) where*

$$\begin{array}{l} \bar{\tau}_{1n} = \left\{\begin{array}{l} -\lambda_{1,M}\,\mathrm{sign}(\theta_{ne})\ \ if\ \ \theta_{ne}\Delta_{\theta_{ne}} > 0 \\ -\lambda_{1,m}\,\mathrm{sign}(\theta_{ne})\ \ if\ \ \theta_{ne}\Delta_{\theta_{ne}} \leq 0 \end{array}\right. \\ \bar{\tau}_{2n} = \left\{\begin{array}{l} -\lambda_{2,M}\,\mathrm{sign}(v_{ne})\ \ if\ \ v_{ne}\Delta_{v_{ne}} > 0 \\ -\lambda_{2,m}\,\mathrm{sign}(v_{ne})\ \ if\ \ v_{ne}\Delta_{v_{ne}} \leq 0 \end{array}\right. \end{array} \tag{26}$$

with

$$\begin{array}{l} \Delta_{\theta_{ne}} = \left\{\begin{array}{ll} 0 & if\, k = 0 \\ \theta_{ne}(kT_s) - \theta_{ne}((k-1)T_s) & else \end{array}\right. \\ \Delta_{v_{ne}} = \left\{\begin{array}{ll} 0 & if\, k = 0 \\ v_{ne}(kT_s) - v_{ne}((k-1)T_s) & else \end{array}\right. \end{array} \tag{27}$$

T_s is the sampling period, $k \in \mathbb{N}$ is related to the time of the process and $\lambda_{i,m}$, $\lambda_{i,M}$, $i = 1,2$ are positive constants high enough to enforce the sliding motion.

Proof. It can be shown that this controller ensures a finite time convergence of the trajectories onto the manifold $\{v_{ne} = \dot{v}_{ne} = 0\}$ and $\{\theta_{ne} = \dot{\theta}_{ne} = 0\}$ (see (Fridman and Levant, 2002) for further details). Hence, the application of the control input (26) results in the robust finite time stabilization of θ_{ne} and v_{ne}. □

Remark 9. *We would like to emphasize that although not explicitly considered here the procedure based on sliding mode control guarantees proper behavior even in the presence of uncertainties in the mass and inertia of the robots and additive disturbances to the linear and angular velocities which constitute very realistic assumptions.*

Once the sliding mode occurs on all the surfaces (which happens in finite time), based on Proposition 2, the global control objectives, defined in Section 2.2, are fulfilled.

Some specific advantages of the proposed decentralized algorithm are enumerated below:

- robustness with respect to uncertainties and disturbances (sliding mode controller),
- reactivity (potential field functions),
- low communication bandwidth, i.e. small amount of information is locally exchanged,
- reduction of deadlocks due to local minima in potential field (anticipation and coordination mechanism through the receding horizon planner).

4 SIMULATION RESULTS

This section demonstrates the performance of the proposed decentralized algorithm. The following simulations showcase two different scenarios for which the environment is partially known (i.e. the range of sensors of each robot is of radius $1.5m$).

The main parameters of the robots are: $\forall n$, $R_n = 0.25m$, $\overline{R}_n = 4m$ and $u_{n,max} = 1m/s$. For the decentralized algorithm, the following parameters are used: $T_p = 3s$, $T_c = 0.5s$, $a_n = c_n = 1$, $b_n = 2$, $d_n = 0.5$, $\lambda_{1,M} = \lambda_{2,M} = 10$, $\lambda_{1,m} = \lambda_{2,m} = 1$ and $T_s = 0.01s$.

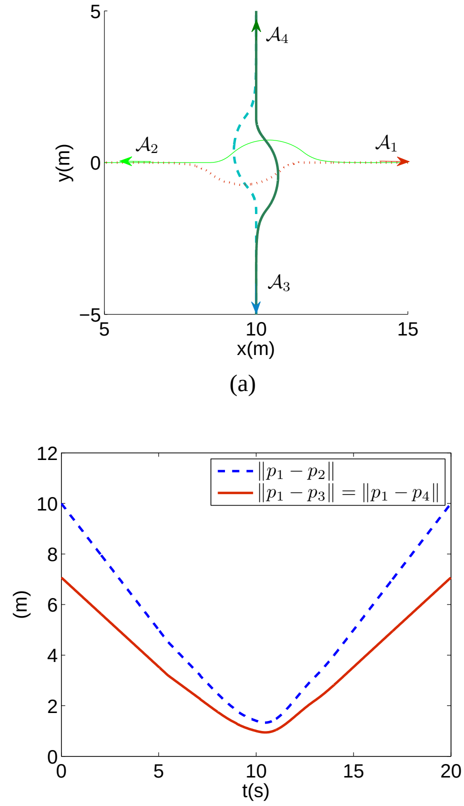
4.1 Scenario 1: Crossing

In this scenario, there are four robots ($N = 4$) starting at $\boldsymbol{p}_1(t_{ini}) = [5,0]^T$, $\boldsymbol{p}_2(t_{ini}) = [15,0]^T$, $\boldsymbol{p}_3(t_{ini}) = [10,5]^T$ and $\boldsymbol{p}_4(t_{ini}) = [10,-5]^T$ respectively, with velocities equal to zero. These robots must cross each other in order to reach their desired configuration $\boldsymbol{p}_{1,des} = [15,0]^T$, $\boldsymbol{p}_{2,des} = [5,0]^T$, $\boldsymbol{p}_{3,des} = [10,-5]^T$ and $\boldsymbol{p}_{4,des} = [10,5]^T$. One can note that this problem is not trivial due to its symmetry properties.

The simulation results are given in Fig. 5. One can see that each robot modifies its trajectory in order to avoid collision. Figure 5(b) depicts the evolution of the distance between robots. Since it is higher than $0.5m$, the collision avoidance is guaranteed.

4.2 Scenario 2: Reconfiguration with Collision Avoidance

In this scenario, a swarm of five robots ($N = 5$) reconfigures its geometric shape (from "linear" to "triangular") while avoiding collisions with obstacles. The proposed decentralized controller has only a limited knowledge of the obstacles (initially unknown). It simply keeps the robots spaced out using the proposed potential field technique. The five robots make decisions in order to avoid collisions. One can note that the number of potential conflicts is high.



(a)

(b)

Figure 5: Four vehicles simulation: (a) Robot trajectories. (b) Relative distances between $\mathcal{A}_1$ and other robots.

One can see in Fig. 6 that under the proposed decentralized algorithm, the robots meet the objective defined in Section 2.2. Note that the radius of obstacles is increased by 0.25m (dotted lines around obstacles) to take the size of robots into account.

5 CONCLUSIONS

A new distributed strategy for the navigation of multiple autonomous robots is presented. The proposed scheme combines a decentralized receding horizon motion planner to satisfy middle-term objectives (coordination between cooperative robots) with a fast navigation controller based on artificial potential fields and sliding mode control technique to satisfy short-term objectives (collision avoidance and

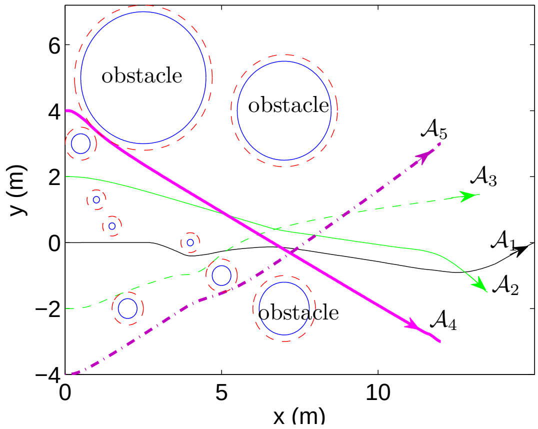


Figure 6: Collision avoidance of five robots.

trajectory tracking). The fact that there is no leader increases the security and the robustness of the missions. Simulation studies are provided in order to show the effectiveness of the proposed approach.

Experimental testing on WifiBot is under way. In the future, it is planned to design real time observers to estimate the relative velocities between robots.

REFERENCES

Das, A., Fierro, R., Kumar, V., Ostrowski, J., Spetzer, J., and Taylor, C. (2002). A vision-based formation control framework. *IEEE T. Robotic Autom.*, 18(5):pp. 813–825.

De-Gennaro, M. and Jadbabaie, A. (2006). Formation control for a cooperative multi-agent system using decentralized navigation functions. In *American Control Conf.*

Defoort, M., Floquet, T., Kokosy, A., and Perruquetti, W. (2007). Decentralized robust control for multi-vehicle navigation. In *European Control Conf.*

Defoort, M., Palos, J., Kokosy, A., Floquet, T., and Perruquetti, W. (2009). Performance based reactive navigation for nonholonomic mobile robots. *Robotica*, 27(2):pp. 281–290.

Do, K., Jiang, Z., and Pan, J. (2004). A global output-feedback controller for simultaneous tracking and stabilization of unicycle-type mobile robots. *IEEE T. Automat. Contr.*, 20(3):pp. 589–594.

Dunbar, W. and Murray, R. (2002). Model predictive control of coordinated multi-vehicle formation. In *IEEE Conf. on Decision and Control*.

Dunbar, W. and Murray, R. M. (2006). Distributed receding horizon control for multi-vehicle formation stabilization. *Automatica*, 42(4):pp. 549–558.

Fiorini, P. and Shiller, Z. (1998). Motion planning in dynamic environments using velocity obstacles. *Int. J. Robot. Res.*, 17(7):pp. 760–772.

Fliess, M., Levine, J., Martin, P., and Rouchon, P. (1995). Flatness and defect of nonlinear systems: introductory theory and examples. *Int. J. Control*, 61(6):pp. 1327–1361.

Fridman, L. and Levant, A. (2002). Higher order sliding mode modes. *Sliding mode control in Engineering, Ed W. Perruquetti, J. P. Barbot*, pages 53–101.

Huang, L. (2009). Velocity planning for a mobile robot to track a moving target - a potential field approach. *Robot. Auton. Syst.*, 57(1):pp. 55–63.

Isidori, A. (1989). *Nonlinear control systems*. Springer, New York, 2nd edition.

Krstic, M., Kanellakopoulos, I., and Kokotovic, P. (1995). *Nonlinear and adaptative control design*. Wiley, N. Y.

Kuchar, J. and Yang, L. (2000). A review of conflict detection and resolution modeling methods. *IEEE T. Intell. Transp.*, 1(4):pp. 179–189.

Kuwata, Y., Richards, A., Schouwenaraars, T., and How, J. (2006). Decentralized robust receding horizon control. In *American Control Conf.*

Latombe, J. (1991). *Robot Motion Planning*. Kluwer Academic Publishers, Norwell, MA.

Mayne, D., Rawlings, J., Rao, C., and Scokaert, P. (2000). Constrained model predicitive control: Stability and optimality. *Automatica*, 36(6):pp. 789–814.

Pallatino, L., Scordio, V., Bicchi, A., and Frazoli, E. (2007). Decentralized cooperative policy for conflict resolution in multivehicle systems. *IEEE T. Robot.*, 23(6):pp. 1170–1183.

Tomlin, C., Pappas, G., and Sastry, S. (1998). Conflict resolution for air traffic management: a study in multiagent hybrid systems. *IEEE T. Automat. Contr.*, 43(4):pp. 509–521.

Utkin, V., Guldner, J., and Shi, J. (1999). *Sliding Modes Control in Electromechanical Systems*. Taylor and Francis, New York.

INTERFERENCE REDUCTION THROUGH TASK PARTITIONING IN A ROBOTIC SWARM
Or: *"Don't you Step on My Blue Suede Shoes!"*

Giovanni Pini, Arne Brutschy, Mauro Birattari and Marco Dorigo
IRIDIA, CoDE, Université Libre de Bruxelles, Brussels, Belgium
{gpini, arne.brutschy, mbiro, mdorigo}@ulb.ac.be

Keywords: Swarm robotics, Foraging, Self-organized task allocation, Task partitioning, Swarm intelligence.

Abstract: This article studies the use of task partitioning as a way to reduce interference in a spatially constrained harvesting task. Interference is one of the key problems in large cooperating groups. We present a simple method to allocate individuals of a robotic swarm to a partitioned task, and show that task partitioning can increase system performance by reducing sources of interference. The method is experimentally studied, both in an environment with a narrow area and an environment without this constraint. The results are analyzed and compared to the case in which task partitioning is not employed.

1 INTRODUCTION

In collective robotics, interference is a critical problem limiting the growth of a group: the time each robot spends in non-task-relevant behaviors such as obstacle avoidance increases when the density of individuals rises—see e.g., (Lerman and Galstyan, 2002). The performance on tasks that suffer from physical interference can typically be improved by spatial partitioning; for example, by keeping each robot in its own "working area". A known approach that uses this rationale is the so called bucket-brigade (Fontán and Matarić, 1996; Shell and Matarić, 2006). In this approach, robots hand over objects to robots working in the following area, until the objects reach their destination. As tasks usually cannot be partitioned arbitrarily, this approach effectively limits the number of robots that can be employed in the task. A possible solution to this problem, treating working areas as non-exclusive, raises other problems: How should individuals be allocated to tasks? How can such an allocation help in limiting the amount of interference?

In this paper, we study how task partitioning can help in reducing sources of interference. Additionally, we show a simple way to achieve self-organized allocation to such a task partition when using a robotic swarm.

We use the foraging problem, one of the canonical testbeds for collective robotics, as a base for our studies. In our experiments, a swarm of homogeneous robots has to harvest prey objects from a source area and transport them to a home area. In this study, we limit ourselves to a harvesting task that is pre-partitioned by the designer into two subtasks with a sequential interdependency. We study a simple threshold-based model of self-organized allocation and focus on two questions: Under which environmental conditions is it advantageous to partition the task? Can this partition reduce interference between group members? These questions are studied in two experiments using a simulated robot swarm.

The paper is organized as follows. We first review related works in Section 2. In Section 3 we explain the task partitioning and the allocation method employed in this study. Section 4 gives the methods used in the experiments by describing the environments, the simulated robots, and the controller. In Section 5 the results of the experiments are given and discussed. Section 6 draws some conclusions and discusses future work.

2 RELATED WORK

Interference has long been acknowledged as being one of the key issues in multi-robot cooperation (Goldberg and Matarić, 2003). (Lerman and Galstyan, 2002) devised a mathematical model that allows a quantification of the interference and its effect on group performance. Probably, the most thorough study was published by (Goldberg, 2001), who identified several types of multi-robot interactions. Goldberg notes that one of the most common types of interference is physical interference in a central area, for example the nest. This kind of interference results from resource conflicts, in this case physical space, and can be arbitrated by either making sure that robots stay in different areas all the time or by employing a scheduling mechanism to ensure that robots use the same space only at different times.

A simple method for reducing interference by using the first arbitration method mentioned is the so-called bucket-brigade: robots are forced to stay in exclusive working areas and to pass objects to the following robot as soon as they cross the boundaries of their area (Fontán and Matarić, 1996; Shell and Matarić, 2006). Recently, this has been extended to work with adaptive working areas by (Lein and Vaughan, 2008). To the best of our knowledge, current works concerned with bucket brigading only studied the influence of interference due to obstacle avoidance. Other sources of interference (e.g., object manipulation) were never studied, although they might have a critical impact on the performance of any task partitioning approach. To quote (Shell and Matarić, 2006): *"If the cost of picking up or dropping pucks is significant [. . .], then bucket brigading may not be suitable."*

Task allocation for multi-robot systems is a wide field, which can be divided in intentional and self-organized task allocation. Intentional task allocation relies on negotiation and explicit communication to create global allocations, whereas in self-organized task allocation global allocations result from local, stochastic decisions. A formal analysis and taxonomy that covers intentional task allocation has been proposed by (Gerkey and Matarić, 2004). (Kalra and Martinoli, 2006) recently compared the two best-known approaches of intentional and self-organized task allocation.

The field of self-organized task allocation is in its early stages, as most studies tackle simple problems without task interdependencies. Studies in self-organized task allocation are mostly based on threshold-based approaches, taking inspiration from division of labor in social insects. (Krieger and Billeter, 2000) were among the first to propose threshold-based approaches in multi-robot task allocation. (Labella et al., 2006) used threshold-based task allocation in a multi-foraging task. Similarly, (Campo and Dorigo, 2007) used a notion of the group's internal energy to allocate individuals to a multi-foraging task. Finally, (Liu et al., 2007) studied a multi-foraging task while focusing on the influence of the use of different social cues on the overall group performance.

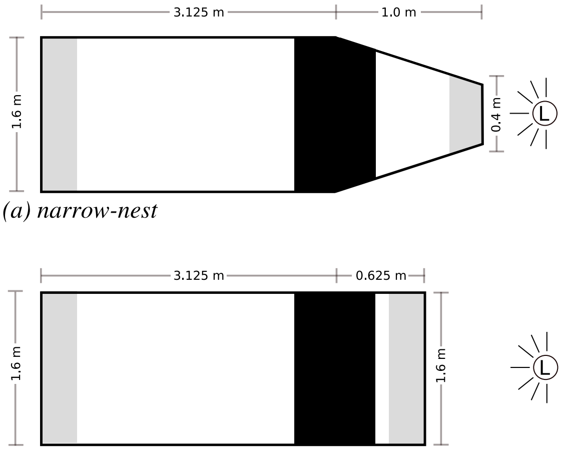
3 TASK PARTITIONING AND ALLOCATION

In this work, we study a collective foraging task. By spatially partitioning the environment, the global foraging task is automatically partitioned into two subtasks: 1) harvesting prey objects from a harvesting area (*source*) and 2) transporting them to a home area (*nest*). Robots working on the first subtask harvest prey objects from the source and pass them to the robots working on the second subtask, which store the objects in the nest. These subtasks have a sequential interdependency in the sense that they have to be performed one after the other in order to complete the global task once: delivering a prey object to the home area.

Robots can decide to switch from one subtask to the other, thus creating a task allocation problem: individual robots have to be allocated to subtasks and different allocations yield different performance. As a prey object has to be passed directly from one robot to the other, a robot usually has to wait some time before passing a prey object to or receiving a prey object from a robot working on the other subtask. This waiting time can therefore give an indication of the allocation quality for the respective subtask: if the waiting time is very long, there might not be enough robots allocated to the other subtask. Thus, the robots can use this waiting time to decide whether to switch subtask or not. Ideally, the waiting time should be the same for the two subtasks in order for the system to reach a stable state and deliver optimal performance.

Our robots exploit a simple threshold-based model to decide when to switch task: when the waiting time t_w is higher than a threshold Θ, a robot switches its subtask. The robot's waiting time is a function of the average time the robots working in the other subtask need to complete their task. The task-completion time of a robot depends on two factors: 1) round-trip-time (i.e., distance to travel) and 2) time lost due to interference. Thus, the robot's threshold Θ is a function of the round-trip-time and the interference of the robots in the other subtask. Therefore, the optimal task switching threshold depends on the task (i.e., time to harvest a prey object) and the environment (i.e., distance between the source and the nest). As the parameters of the environment are not pre-programmed into the robots, determining the optimal threshold can be a complex problem. In this work, we limit ourselves to a simple method for setting this threshold: at the start of the experiment, each robot draws a random threshold that is used as its task switching threshold throughout the experiment.

In the following, we study the properties of this simple self-organized task allocation strategy, compare this strategy to a strategy without task partitioning, and analyze how it can help to reduce interference. We refer to the two strategies as *partitioned* and *non-partitioned*, respectively.



(a) narrow-nest

(b) wide-nest

Figure 1: Depiction of (a) the *narrow-nest* environment used in the first experiment and (b) the *wide-nest* environment used in the second experiment. The gray stripes are the source (left), and the nest (right), each 0.25 m deep. The black stripe is the exchange zone, that is 0.5 m deep. The light source is marked with "L".

4 METHODS

This section describes the environments in which the experiments are carried out, the simulated robots, and the robot's controller. Additionally, we describe how we run the experiments and we introduce some metrics that we use to evaluate the properties of the system.

4.1 Environments

We study task allocation in two different environments. In these two environments, the nest is marked by a light source that can be perceived by all robots, thus providing directional information. The environment is spatially partitioned in two parts: the source is located on the left and the nest is located on the right side of the arena. We refer to the two sides of the arena as *harvest area* and *store area*, respectively. The *exchange zone* is located between these two areas. Robots working on the left side, called *harvesters*, gather prey objects in the source and move them to the exchange zone, where they pass them to the robots working on the other side. These are referred to as *storers*: their role is to transport prey objects to the nest and store them there. The nest, the source, and the exchange zone can be detected through environmental cues (ground color).

At time $t = 0$, the robots are randomly placed in the harvest area. The experiments run for $t_{max} = 18,000$ time steps (a simulated time of one hour, with a time step length of 200 ms). The experiments are run in two different arenas (see Figure 1). The first arena (Figure 1a) is 4.125 m long with a width of 1.6 m at the source and exchange zone, whereas the nest is 0.4 m wide. The exchange zone is located 3.125 m away from the source. This arena is characterized by the presence of an area, critical for the task, in which high interference between robots can be expected (the nest). Thus, this arena is referred to as the *narrow-nest* environment.

The second arena (Figure 1b) has a rectangular shape: it is 3.75 m long and 1.6 m wide. Here as well the exchange zone is located 3.125 m from the source. The arena shape does not suggest the presence of any zone where interference can be higher than in other places. This arena is referred to as the *wide-nest* environment.

The area of both arenas is $6\,\mathrm{m}^2$, $5\,\mathrm{m}^2$ for the harvest area and $1\,\mathrm{m}^2$ for the store area. The overall area is the same in the two arenas, so that the same group size results in the same robot density. Thus, results are comparable across the two environments.

4.2 Simulation

The experiments are carried out in a custom simulation environment that models geometries and functional properties of simple objects and robots. Our robots' model is purely kinematic. Prey objects are simulated as an attribute a robot can posses and not as physical entities. Although the experiments are conducted in simulation only, the simulated robots have a real counterpart: the swarm-bot robotic platform (Mondada et al., 2004). The platform consists of a number of mobile autonomous robots called s-bots, which have been used for several studies, mainly in swarm intelligence and collective robotics—see for instance (Groß et al., 2006) and (Nouyan et al., 2008). The simulated s-bots are of round shape, with a diameter of 0.116 m. Each of them is equipped with 16 infrared proximity sensors, used to perceive obstacles up to a distance of 0.15 m. Eight ambient light sensors can be used to perceive light gradients up to a distance of 5.0 m. The robots are equipped with 4 ground sensors used to perceive nest, source and exchange zone. A 8 LEDs ring is used to signal when a prey object is carried. An omnidirectional camera allows the perception of LEDs in a circle of radius 0.6 m surrounding the robot. A uniform noise of 10% is added to all sensor readings at each simulation step. The robots can move at a maximum speed of 0.1 m/s by means of a differential drive system.

4.3 Controller

All the robots share the same, hand coded, finite state machine controller depicted in Figure 2. The controller consists of two parts, each corresponding to a possible subtask a robot can perform. Gray states refer to the harvest subtask, white states to the store subtask. Since all the robots start in the harvest area, their controller is initially set to perform anti-phototaxis. In this way they will reach the source, where they can start retrieving prey objects. The behavior of each robot is a function of the task it is performing. Harvesters not carrying a prey object move towards the source, where they can find prey. Harvesters carrying a prey object, move to the exchange zone and wait for a free storer. Upon arrival of such a storer, the harvester passes the prey object to it. Storers carrying a prey object move towards the nest, where they can deposit the object. Storers not carrying a prey object head to the exchange zone and search for a harvester with a prey object. Robots can detect other robots carrying a prey on the basis of the color of their LED ring. While moving, each robot avoids obstacles (walls and other robots).

Task switches can occur: a harvester carrying a prey object can decide to become a storer, and a storer not carrying a prey object can decide to become a harvester. As mentioned before, robots switch task depending on an internal threshold Θ, representing the maximum amount of control cycles they can spend in the transfer zone waiting to pass (harvesters) or receive (storers) a prey object. If a robot remains in the transfer zone longer than its threshold without passing or receiving prey objects ($t_w > \Theta$), it switches its task. The optimal threshold value is not trivial to determine. In the work presented here, we use a simple method to set the threshold Θ: at the beginning of the experiment, each robot draws a random threshold, sampled uniformly in the interval $[0, 1000]$. We chose this method because it is independent of the environment and does not rely on complex approximation techniques. The threshold value does not change during the experiment. In case of the non-partitioned strategy, the threshold is set to $\Theta = 0$, causing the robots to switch subtask immediately as soon as they reach the exchange zone.

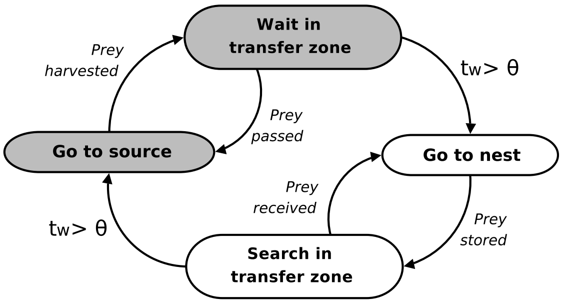


Figure 2: Simplified state diagram of the controller of the robots. Gray states belong to the harvest task, white states to the store task. The obstacle avoidance state has been omitted for clarity, as it is applicable in all states of the robot. t_w is the time spent in the exchange zone and Θ is the threshold.

4.4 Experiments

The goal of the experiments is to investigate whether task partitioning can reduce interference in task-critical zones, and how to allocate a robotic swarm to partitions. As pointed out by (Lerman and Galstyan, 2002), interference is related to the number of individuals in the system. Additionally, the physical interference between robots is also a function of the environment the robots act in. The higher the group size, the higher the density, resulting in a higher amount of physical interference. Thus, in order to study interference in our experiments, we increase the size of the group in each of the two environments shown in Figure 1, while using both strategies (non-partitioned and partitioned). We study the performance of the system when the group size N ranges in the interval $[1, 40]$. We run 50 repetitions for each value of N and each experimental settin.

4.5 Metrics

In order to quantify the influence of interference, we measure the *group performance P* by the number of prey objects collected by the swarm at the end of the experiment ($t_{max} = 1$ hour). From the group performance measure we can derive the *individual efficiency* as follows:

$$I_{eff} = P/N, \tag{1}$$

where N is the size of the group. Individual efficiency can help to understand the effect of interference on the performance.

In order to measure the influence of environmental features on the interference, we define an interference measure taking inspiration from (Rosenfeld et al., 2005). In their work, interference is measured

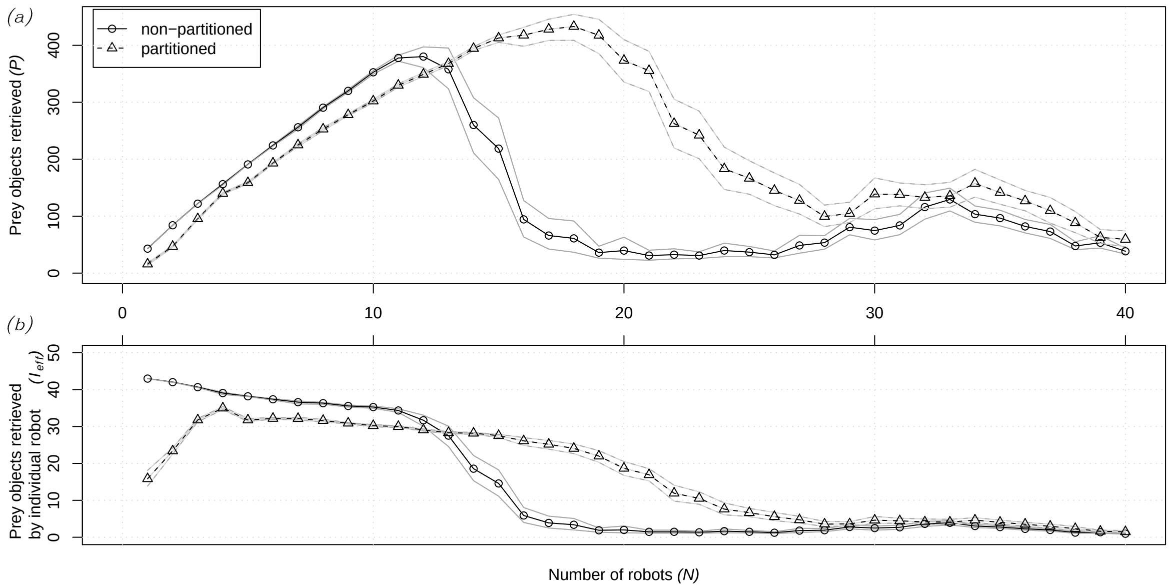


Figure 3: (a) Performance P and (b) individual efficiency I_{eff} for increasing number of robots in the narrow-nest environment. The black continuous line refers to the case of no task partitioning, the black dashed line to the case of partitioning. Gray lines indicate the 95% confidence interval on the expected value.

as the time spent performing actions not strictly related to the task, but rather lost due to negative interactions with the environment (e.g., obstacle avoidance maneuvers). By registering the number of collisions for each area of the arena, we can draw conclusions about where physical interferences happen most often. We measure interference through the state of the controller: in our case a robot is experiencing interference each time its controller perceives an obstacle.

In case of a partitioned task, there is another source of inefficiency that adds to interference: the time lost in the exchange zone. We define the *strategy cost C* as the sum of time lost because of physical interference and time lost in the exchange zone:

$$C = T_{int} + T_{part}, \qquad (2)$$

where T_{int} is the amount of time steps during which the controller perceives an obstacle, and T_{part} is the total amount of time steps spent in prey passing maneuvers. By using this metric, the cost of the non-partitioned strategy is purely due to interference ($T_{part} = 0$), while in case of the partitioned strategy, prey passing costs add to interference costs. In a way, passing a prey object produces another kind of interference in the system. The strategy cost captures this effect, thus allowing for a comparison of strategies.

5 RESULTS AND DISCUSSION

The graphs in Figures 3a and 4 show the performance P for different group sizes in the narrow-nest and wide-nest environment respectively. Figure 3b shows the individual efficiency I_{eff} of the robots in the narrow-nest environment. Black curves are the average computed over the 50 repetitions of each setting, gray curves indicate the 95% confidence interval on the expected value. The performance graph in Figure 3a shows that the partitioned strategy improves performance in the narrow-nest environment. The graph shows that the non-partitioned strategy performs better than the partitioned strategy for small group sizes (up to $N = 13$ robots). However, increasing the group size makes the non-partitioned strategy collapse: the number of gathered prey objects drops dramatically for groups larger than 13. The individual efficiency graph (Figure 3b) can explain the behavior of the system. The robots employing the partitioned strategy are less efficient, for small group sizes, than those performing the non-partitioned strategy. However, the addition of more individuals affects the efficiency of the non-partitioned group in a more dramatic way. At a certain point, the drop in efficiency becomes very steep for the non-partitioned strategy. On the other hand, the partitioned strategy scales better: individual efficiency drops smoothly. This explains why a group using the partitioned strategy performs better: it can benefit from the work of more individuals and therefore collects more prey objects.

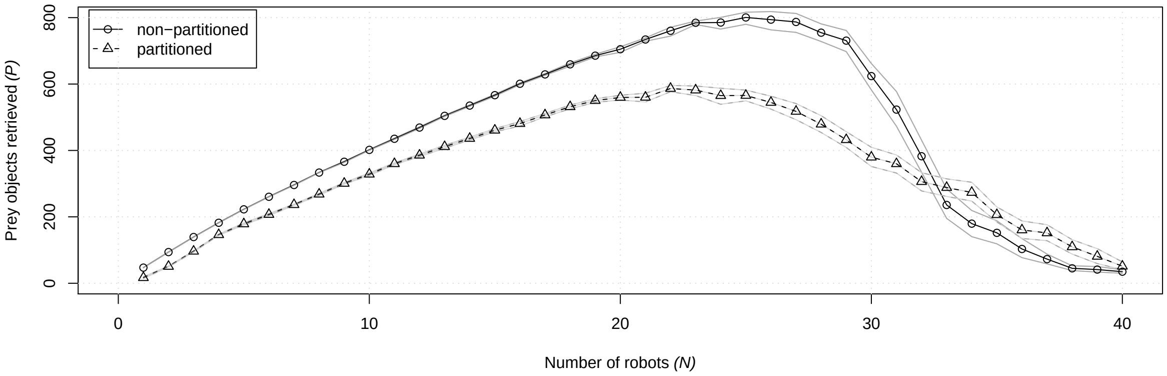


Figure 4: Performance P for increasing number of robots in the wide-nest environment. The black continuous line refers to the case of no task partitioning, the black dashed line to the case of partitioning. Gray lines indicate the 95% confidence interval on the expected value.

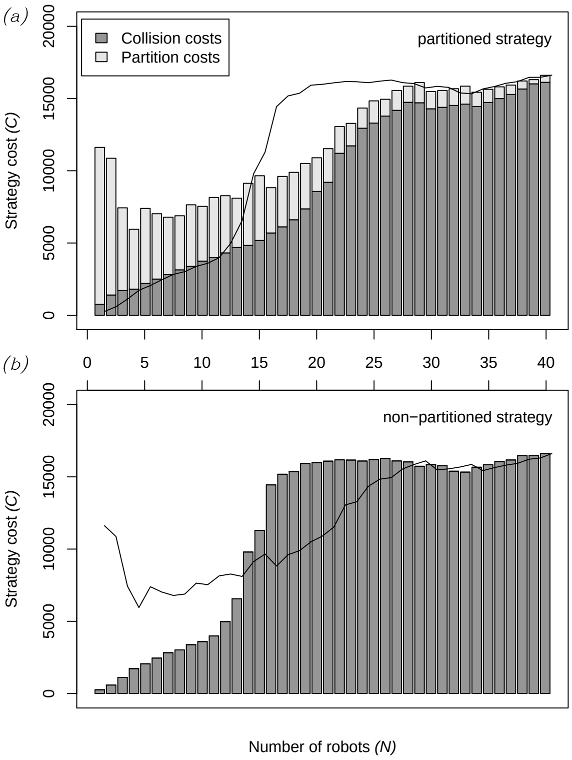


Figure 5: Cost of interference in the narrow-nest environment. Bars represent the cost C, sum of interference time T_{int} and partition time T_{part} (i.e., waiting times). For easy reference, the outline of the bars of the respective other graph has been added to each graph. (a) Costs for the partitioned strategy, where interference cost stem from waiting times and collisions. (b) Cost in case of the non-partitioned strategy, where only physical interference through collisions exists.

These considerations do not hold in the wide-nest environment. The performance graph in Figure 4 shows that the non-partitioned strategy performs better than the partitioned strategy for group sizes $N < 33$. In both the environments, independently of the strategy used to accomplish the task, the system collapses when the area is saturated by the swarm.

Figure 5 shows the effect on the cost of increasing the number of robots in the narrow-nest environment. The graph compares the cost C of each of the two strategies for different group sizes. In case of the partitioned strategy (Figure 5a), the graph shows each component of the cost (T_{int} and T_{part}). Clearly, task partitioning has the effect of reducing the cost due to interference but has the disadvantage of increasing the cost due to time lost. The probability of two or more robots encountering each other increases with the robot density. Although this determines a higher interference cost (i.e., T_{int}), it decreases the cost due to lower waiting time (i.e., T_{part}) in the case of the partitioned strategy. Partitioning performs better when the gain from interference reduction is greater than the loss of performance due to partitioning inefficiencies. These considerations hold in the narrow-nest environment, where the likelihood of physical interference in a task-critical zone is very high. In the wide-nest environment, interference in the nest is as likely as interference in the exchange zone. Thus, it is not beneficial to pay the cost of waiting and the non-partitioned strategy performs better for any group size.

The mechanism by which partitioning reduces interference costs can be deduced by comparing the interference graphs in Figure 6. The graphs show the number of times that physical interference (as defined in Section 4.5) was registered in each region of the narrow-nest environment. The total area was discretized in squares of $1\,\text{cm}^2$. Figure 6 shows the results obtained with 18 robots, in the case of the non-partitioned strategy (Figure 6a) and in the case of the partitioned strategy (Figure 6b). The graphs show that the use of the non-partitioned strategy leads to high interference in the nest, which becomes con-

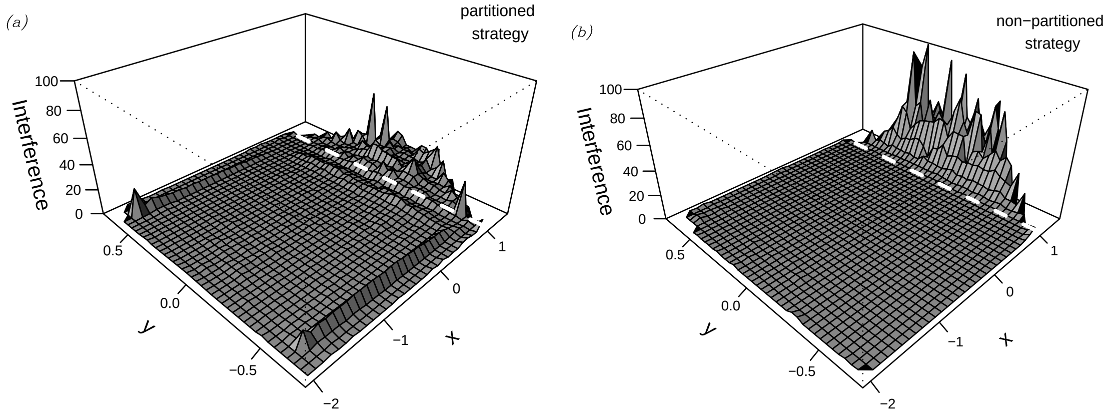


Figure 6: Mean interference values registered for (a) the partitioned strategy and (b) the non-partitioned strategy, both in the narrow-nest environment. Shown values are observation means of 50 repetitions with $N = 18$ robots. Coordinates on the x- and y-axis are given in meters. The arena is stretched along the y-axis for better visualization. The dashed white line marks the location of the exchange zone.

gested. Partitioning the task reduces the robot density in the nest, thus spreading the interference more uniformly across the arena. In addition, the overall interference diminishes because the exchange zone is wider: the robots have more freedom of movement and collide less often. Although the graphs show only data collected with 18 robots, experiments with different group sizes produced similar results.

6 CONCLUSIONS AND FUTURE WORK

Interference can be an issue when working with swarms of robots. In this work, we used task partitioning and allocation to reduce interference between robots sharing the same physical space. We manually partitioned the environment and employed a simple self-organized strategy for allocating individuals to subtasks. Results show that a partitioning strategy improves performance in a constrained environment. Additionally, we identified cases in which partitioning is not advantageous and a non-partitioned strategy should be used. The proposed strategy is fairly simple and far from being an optimal solution, nevertheless we improved the performance of the swarm when interference was costly.

Future work will concern the identification of the optimal allocation in the studied environments as well as the development and study of a strategy that can find this optimal allocation in a self-organized and adaptive way. In addition, the interference metric proposed in Section 4.5 could be used by the robots to decide whether to partition the task. In this way, we could achieve even better performance, since partitioning would be employed only when strictly needed. Finally, the goal is to validate the system using the real robots.

ACKNOWLEDGEMENTS

This work was supported by the SWARMANOID project, funded by the Future and Emerging Technologies programme (IST-FET) of the European Commission, under grant IST-022888 and by the VIRTUAL SWARMANOID project funded by the Fund for Scientific Research F.R.S.–FNRS of Belgium's French Community. The information provided is the sole responsibility of the authors and does not reflect the European Commission's opinion. The European Commission is not responsible for any use that might be made of data appearing in this publication. Marco Dorigo and Mauro Birattari acknowledge support from the Fund for Scientific Research F.R.S.–FNRS of Belgium's French Community, of which they are a research director and a research associate, respectively.

REFERENCES

Campo, A. and Dorigo, M. (2007). Efficient multi-foraging in swarm robotics. In *Advances in Artificial Life: Proceedings of the VIIIth European Conference on Artificial Life, Lecture Notes in Artificial Intelligence LNAI 4648*, pages 696–705, Berlin, Germany. Springer Verlag.

Fontán, M. S. and Matarić, M. J. (1996). A study of territoriality: The role of critical mass in adaptive task division. In *From Animals to Animats 4: Proceedings of* *the Fourth International Conference of Simulation of Adaptive Behavior*, pages 553–561, Cambridge, MA. MIT Press.

Gerkey, B. P. and Matarić, M. J. (2004). A formal analysis and taxonomy of task allocation in multi-robot systems. *The International Journal of Robotics Research*, 23(9):939–954.

Goldberg, D. (2001). *Evaluating the Dynamics of Agent-Environment Interaction*. PhD thesis, University of Southern California, Los Angeles, CA.

Goldberg, D. and Matarić, M. J. (2003). Maximizing reward in a non-stationary mobile robot environment. *Autonomous Agents and Multi-Agent Systems*, 6(3):287–316.

Groß, R., Bonani, M., Mondada, F., and Dorigo, M. (2006). Autonomous self-assembly in swarm-bots. *IEEE Transactions on Robotics*, 22(6):1115–1130.

Kalra, N. and Martinoli, A. (2006). A comparative study of market-based and threshold-based task allocation. In *Proceedings of the 8th International Symposium on Distributed Autonomous Robotic Systems (DARS)*, Minneapolis, Minnesota, USA.

Krieger, M. J. B. and Billeter, J.-B. (2000). The call of duty: Self-organised task allocation in a population of up to twelve mobile robots. *Journal of Robotics and Autonomous Systems*, 30:65–84.

Labella, T. H., Dorigo, M., and Deneubourg, J.-L. (2006). Division of labor in a group of robots inspired by ants' foraging behavior. *ACM Transactions on Autonomous and Adaptive Systems*, 1(1):4–25.

Lein, A. and Vaughan, R. (2008). Adaptive multi-robot bucket brigade foraging. In *Proceedings of the Eleventh International Conference on Artificial Life (ALife XI)*, pages 337–342, Cambridge, MA. MIT Press.

Lerman, K. and Galstyan, A. (2002). Mathematical model of foraging in a group of robots: Effect of interference. *Auton. Robots*, 13(2):127–141.

Liu, W., Winfield, A., Sa, J., Chen, J., and Dou, L. (2007). Towards energy optimization: Emergent task allocation in a swarm of foraging robots. *Adaptive Behavior*, 15(3):289–305.

Mondada, F., Pettinaro, G. C., Guignard, A., Kwee, I. V., Floreano, D., Deneubourg, J.-L., Nolfi, S., Gambardella, L. M., and Dorigo, M. (2004). SWARM-BOT: A new distributed robotic concept. *Autonomous Robots*, 17(2–3):193–221.

Nouyan, S., Campo, A., and Dorigo, M. (2008). Path formation in a robot swarm. Self-organized strategies to find your way home. *Swarm Intelligence*, 2(1):1–23.

Rosenfeld, A., Kaminka, G. A., and Kraus, S. (2005). A study of scalability properties in robotic teams. In *Coordination of Large-Scale Multiagent Systems*, pages 27–51, New York. Springer Verlag.

Shell, D. and Matarić, M. J. (2006). On foraging strategies for large-scale multi-robot systems. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 2717–2723, Beijing, China.

HIERARCHICAL PROBABILISTIC ESTIMATION OF ROBOT REACHABLE WORKSPACE

Jing Yang, Patrick Dymond and Michael Jenkin
Department of Computer Science and Engineering, York University, 4700 Keele Street, Toronto, Canada
(jyang, dymond, jenkin)@cse.yorku.ca

Keywords: Reachable workspace, Probabilistic motion planning, Mobile robotics.

Abstract: Estimating a robot's reachable workspace is a fundamental problem in robotics. For simple kinematic chains within an empty environment this computation can be relatively straightforward. For mobile kinematic structures and cluttered environments, the problem becomes more challenging. An efficient probabilistic method for workspace estimation is developed by applying a hierarchical strategy and developing extensions to a probabilistic motion planner. Rather than treating each of the degrees of freedom (DOFs) 'equally', a hierarchical representation is used to maximize the volume of the robot's workspace that is identified as reachable for each probe of the environment. Experiments with a simulated mobile manipulator demonstrate that the hierarchical approach is an effective alternative to the use of an estimation process based on the use of a traditional probabilistic planner.

1 INTRODUCTION

The reachable workspace $\mathcal{W}_{reach}$ is defined as the volume or space within which all the points can be reached by a reference point of the mechanism, for example, the centre of the end-effector (Kumar, 1980). Reachable workspace estimation is a fundamental problem in robotics as workspace properties can represent important criteria in the evaluation and design of mechanical manipulators (Lenarcic and Umek, 1994; Badescu and Mavroidis, 2004), robots (Zacharias et al., 2007) and environmental layouts (Yang et al., 2008). The determination of $\mathcal{W}_{reach}$ involves a considerable amount of numerical calculations, which increases with the number of degrees of freedom of the mechanism and the complexity of the environment.

For kinematic chains within an empty environment the computation of $\mathcal{W}_{reach}$ can be relatively straightforward. For mobile kinematic structures in the presence of obstacles, the problem becomes more challenging. Notably, estimating the $\mathcal{W}_{reach}$ for a mobile robot can be expressed in terms of the ability of the device to plan motions within its environment. Motion planning emerged as a crucial and productive research area in robotics in the late 1960's (Latombe, 1991) and its applications in real world problems continue to attract researchers from all over the world.

In basic motion planning (Latombe, 1991), given a robot $\mathcal{A}$ and a static workspace $\mathcal{W}$ containing a set of obstacles, the objective is to determine a collision-free motion between the specified start and goal for $\mathcal{A}$. A configuration c of $\mathcal{A}$ is a specification of the position and orientation of $\mathcal{A}$ in $\mathcal{W}$. A configuration c is said to be free if $\mathcal{A}$ positioned at c does not collide with any obstacles in $\mathcal{W}$. The free configuration space $\mathcal{C}_{free}$ is defined as the set of all free configurations of $\mathcal{A}$. The motion planning problem is therefore formulated as computing a path in $\mathcal{C}_{free}$ between two given configurations.

A complete solution to the motion planning problem is known to be exponential to the robot's degree of freedom (DOF) (Canny, 1988). As a consequence a number of heuristic approaches to path planning have been developed. The main difference between the probabilistic approaches and earlier complete approaches is that the probabilistic approaches do not attempt to construct an exact representation of $\mathcal{C}_{free}$. Rather they create a simplified graph that approximately "covers" $\mathcal{C}_{free}$ and captures its connectivity in reasonable time. The Probabilistic Roadmap Method (PRM) (Horsch et al., 1994; Kavraki et al., 1996) is a popular heuristic motion planner. The algorithm first constructs a roadmap by connecting randomly sampled collision free configurations and then answers multiple queries by attempting to connect them to the roadmap.

In estimating a robot's workspace it is important to

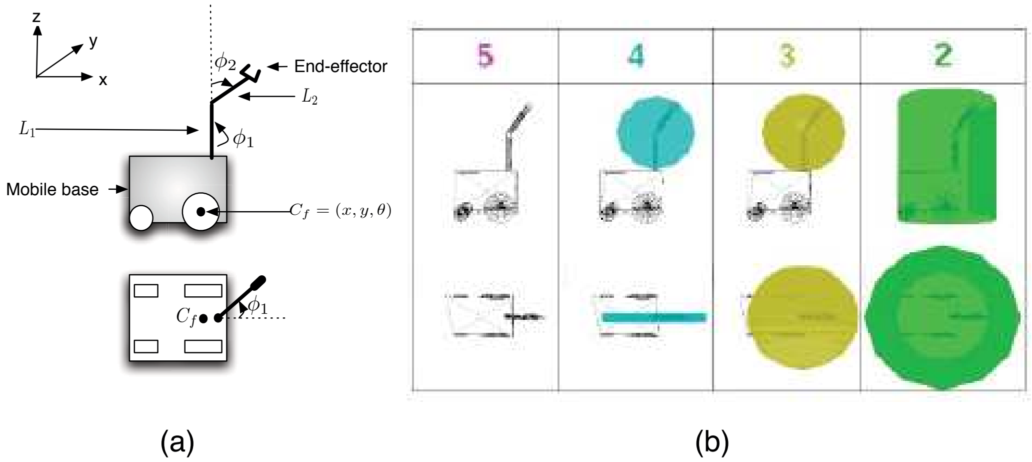


Figure 1: (a) $\mathcal{A}$ consists of $\mathcal{A}_{base}$ and $\mathcal{A}_{arm}$ that has two links L_1 and L_2 connected by revolute joints. The configuration of $\mathcal{A}$ is written as $(x, y, \theta, \phi_1, \phi_2)$. (b) Representations of the hierarchical occupancy of $\mathcal{A}$ are shown in different colors. The body itself also has occupancy constraints but these are not shown.

observe that some DOFs are likely to be more important than others in terms of their effect on how much of the workspace is reachable. In addition, it may be possible to construct subspaces of the workspace defined by arbitrary configurations of certain combinations of joints. Motivated by these observations we explore the use of a hierarchical structure of the DOFs of the kinematic device to establish the entire $\mathcal{W}_{reach}$. In this hierarchy we order the DOFs of the robot in terms of their predefined "importance". Then we consider corresponding sub-versions of the kinematic structure in which sub-versional joints are considered over their range of motion. Each of these sub-versions defines a reachability subspace that can be established as being reachable with a single probe into the robot's workspace. This hierarchical search mechanism can be used to enhance the probabilistic algorithms for reachable workspace estimation.

This paper is structured as follows. Section 2 reviews existing algorithms for reachable workspace estimation. Section 3 outlines our hierarchical approach. Section 4 includes comparison results from applying the hierarchical approach and basic PRM to a simulated mobile manipulator. Finally Section 5 summarizes the work and provides possible directions for future research.

2 RELATED WORK

Although a range of techniques exist for reachable workspace estimation (Badescu and Mavroidis, 2004; Hsu and Kohli, 1987; Zacharias et al., 2007) most existing approaches consider the problem for robotic manipulators and do not consider arbitrary obstacles in the environment. Robotic manipulators are fixed at one end and this assumption provides certain efficiencies for reachable workspace estimation. For example, one straightforward method to compute $\mathcal{W}_{reach}$ is to take planer sections of the workspace defined by the joint angles that make up the kinematic structure and determine the contour of the section in the plane. Rotating and translating this plane based on other joints in the chain yields the three-dimensional workspace (Morecki and Knapczyk, 1999).

A numerical approach calculates the exact $\mathcal{W}_{reach}$ by tracing boundary surfaces of a workspace (Kumar, 1980). In this approach, an imaginary force is applied to the reference point at the end-effector in order to achieve the maximum extension in the direction of the applied force. The manipulator reaches its maximum extension when the force's line of action intersects all joint axes of rotational joints and is perpendicular to all joint axes of prismatic joints. The drawback of this algorithm is its exponential time complexity and that it only deals with manipulators that have ideal joints (without limits). A more efficient system was later developed for computing $\mathcal{W}_{reach}$ for manipulators with joint limits (Alameldin et al., 1990). The system decomposed the problem into two subproblems: workspace point generation by direct kinematic based techniques and surface computation by extracting the workspace contours utilizing a subset of the points generated in the first module.

Much work has been done on capturing workspace properties for interesting kinematic structures such as human arms (Lenarcic and Umek, 1994; Zacharias et al., 2007) and reconfigurable robotic arms (Bade-

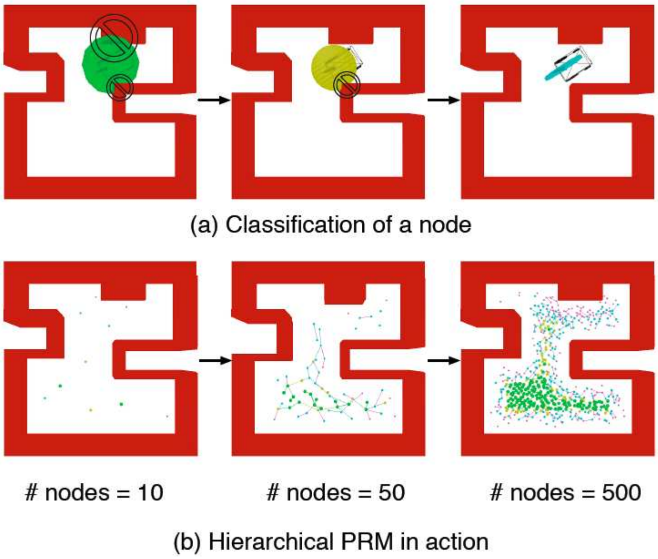


Figure 2: Construction of $\mathcal{R}$. Upper row shows the generation and classification of one node. The rank of a node c^r is calculated by checking the hierarchical occupancy representation of $\mathcal{A}$. The lower row shows the hierarchical PRM in operation. The coverage and connectivity of $\mathcal{R}$ increases as more nodes are added.

scu and Mavroidis, 2004). Lenaric and Umek developed a simplified kinematic model to represent human arm motion. $\mathcal{W}_{reach}$ was determined by calculating the reference point on the wrist for all combinations of values of joint coordinates inside the given ranges. Another approach presented by Zacharias et al. discretizes the workspace into equally sized small cubes. Into each cube a sphere is inscribed and sample points on the sphere are examined using inverse kinematics. The percentage of the points on the sphere that are reachable is used to represent its level of reachability.

The structure of $\mathcal{W}_{reach}$ for a given device can be very complex. Existing methods aim at capturing the exact shape and volume of $\mathcal{W}_{reach}$ and they involve a considerable amount of numerical, time-consuming calculations. It can be expensive to compute exact $\mathcal{W}_{reach}$ for a mobile robot in a cluttered environment. In this paper we develop a probabilistic algorithm to give a proper estimation for $\mathcal{W}_{reach}$ for an arbitrary mobile device operating in a known and cluttered environment.

3 HIERARCHICAL REACHABLE WORKSPACE ESTIMATION

For a robot $\mathcal{A}$ moving in the workspace $\mathcal{W}$, the robot's degrees of freedom (DOFs) are the minimum set of independent displacements/orientations that specify $\mathcal{A}$'s complete position and orientation in $\mathcal{W}$. Thus a configuration c of $\mathcal{A}$ with n DOFs can be specified as a set of n parameters, say $j_1, ..., j_n$, and theoretically there can be $O(n^2)$ different orderings of the joints. First we define an ordering of the DOFs such that more important DOFs have a lower index. Although there can be many definitions according to the nature of the problem, here we define the importance of a DOF by its effect on the volume of $\mathcal{A}$'s occupation in $\mathcal{W}$. This importance weight is expressed more formally below.

For a point a in $\mathcal{A}$, let $P_a : \mathcal{C} \rightarrow \mathcal{W}$ be the mapping that calculates the position of point a in $\mathcal{W}$ when $\mathcal{A}$ is placed at configuration c. Depending on the position of a in $\mathcal{A}$, P_a is a function of $c' \subseteq c$. We say that a is determined by a DOF j_i if $j_i \in c'$. Now we can define a weight function of a DOF j_i in terms of the volume of $\mathcal{A}$ in $\mathcal{W}$ determined by it:

Defn 1: $w(j_i) = \mid \{a \in A \mid a\ is\ determined\ by\ j_i\} \mid$.

A DOF j_x is more important than j_y if $w(j_x) > w(j_y)$. Therefore we can write the configuration c as a vector of length n in a decreasing order of their importance, say $c = (j_1, j_2, ... j_n)$, i.e. for $i = 1, 2, ... n - 1, w(j_i) \geq w(j_{i+1})$.

Take the 5-DOF mobile manipulator shown in Figure 1 as an example. x and y determines the entire robot, and θ determines all the portions except the rotation center of the mobile base (assume the mobile robot can rotate around its center). ϕ_1 determines links L_1 and L_2. ϕ_2 determines link L_2. Therefore, $w(x) = w(y) > w(\theta) > w(\phi_1) > w(\phi_2)$, and the ordered configuration can be written as $(x, y, \theta, \phi_1, \phi_2)$ (or $(y, x, \theta, \phi_1, \phi_2)$).

Given an ordering of the DOFs, we seek a hierarchical representation within which certain joint angles are 'free' and can assume arbitrary values within some previously defined domain. Let the domain of j_i be $\mathcal{D}_i$, $c^r = (j_1, j_2, ... j_r)$ is a subset of $\mathcal{D}_1 \times \mathcal{D}_2 \times ... \mathcal{D}_n$, given by $\{\forall x_{r+1} \in \mathcal{D}_{r+1}, x_{r+2} \in \mathcal{D}_{r+2}, ... x_n \in \mathcal{D}_n \mid (j_1, j_2, .. j_r, x_{r+1}, x_{r+2}, ... x_n)\}$. That is c^r is the set of possible configurations with joints $1...r$ having specific values but joints $r+1...n$ being free. This hierarchical concept applies to general kinematic structures in the domain of motion planning.

Algorithm 1: Node selection.

```
nodeFound ← false
while ¬nodeFound do
    c ← a randomly chosen configuration in $\mathcal{C}$
    for $k$ ← 1 to $n$ do
        if $V(c^k)$ then
            nodeFound ← true
            break
        end if
    end for
end while
$N \leftarrow N \cup \{c^k\}$
```

In order to integrate the above hierarchy into the workspace estimation process, we first perform two types of analysis to the hierarchical representations: occupation analysis and reachability analysis. Define the occupied area $OA_c(c^r)$ of c^r as the union of the occupied volume in $\mathcal{W}$ of every element in c^r, and the reachable area $RA_c(c^r)$ of c^r as the union of the reachable points of every element in c^r. Under the hierarchy nodes with lower r occupy and reach larger workspace than those with higher r. To be precise, we have these two lemmas:

Lemma 1. $\forall i, j \in [0, n], i < j \implies OA_c(c^i) \supseteq OA_c(c^j)$. For some configuration of $\mathcal{A}$, the occupied workspace of the lower hierarchy is the superset of that of the higher hierarchy.

Lemma 2. $\forall i, j \in [0, n], i < j \implies RA_c(c^i) \supseteq RA_c(c^j)$. For some configuration of $\mathcal{A}$, the reachable workspace of the lower hierarchy is the superset of that of the higher hierarchy.

In addition, let $V(c)$ be the function that returns true iff $\mathcal{A}$ is collision free when it is at configuration c. Similarly, $V(c^r)$ returns true iff every element in c^r is valid, i.e. $OA_c(c^r)$ does not collide with any obstacle in the environment. Therefore, we have the following lemma:

Lemma 3. $\forall i, j \in [0, n], i < j \wedge V(c^i) \implies V(c^j)$. For some configuration of $\mathcal{A}$, that its lower hierarchical representation is free implies the higher hierarchical representation is free, too.

Hierarchical representations of both the reachability and occupancy can be very complex shapes depending on the kinematics of the robot. In practice the computation of the exact hierarchical representations is unnecessary. Conservative representations of these complex shapes can provide significant computational savings and this computation can be done prior to the execution of the motion planner. This needs to be done only once for each DOF of the robot, independent of the robot's configuration.

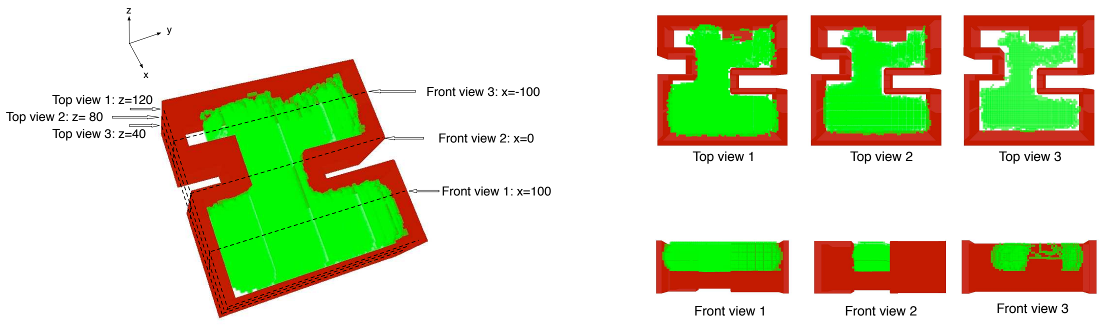
Algorithm 2: Connect(a^{r_1}, b^{r_2}).

```
$\tau$ ← the edge candidate returned by the local path locator
Discretize $\tau$ into a list of configurations $\tau' = (c_1, c_2, ..., c_m)$
$r_{current} \leftarrow MAX(r_1, r_2)$
for all $c_i \in \tau'$ do
    for $k \leftarrow r_{current}$ to $n$ do
        if $V(c_i^k)$ then
            $r_{current} \leftarrow k$
            break
        else
            exit and report failure
        end if
    end for
end for
$E \leftarrow E \cup \{(a, b)^{r_{current}}\}$
```

To establish the general representation of $OA(c^r)$ and $RA(c^r)$, we fix the first r DOFs and take all possible values of the remaining DOFs to construct the hierarchical body. Note that $RA(c^r)$ defines a subspace in the robot's workspace that is reachable by some unique configuration within c^r, and that for any two configurations within c^r there exists a continuous path between them.

As an example, consider the mobile manipulator $\mathcal{A}$ shown in Fig. 1(a), let $\mathcal{D}_1 = [x_{min}, x_{max}]$, $\mathcal{D}_2 = [y_{min}, y_{max}]$, $\mathcal{D}_3 = [-\pi, \pi)$, $\mathcal{D}_4 = [-\pi, \pi)$, and $\mathcal{D}_5 = [-\pi, \pi)$. Fig 1(b) shows the hierarchical representation of $\mathcal{A}$ from level 5 down to 2, indicated by color. For simplicity we used ideal ranges $[-\pi, \pi)$ for the rotational joints of $\mathcal{A}$.

The hierarchy can be integrated into workspace estimation using probabilistic planners such as PRM. We describe below the main steps involved in the construction of a hierarchical roadmap $\mathcal{R}$ for efficient reachable workspace estimation. Nodes with large reachable areas are preferred (they establish more of the environment as being reachable for each calculation). So for each configuration c, we look for the minimum value r_{min} such that $V(c^{r_{min}})$ is true, and we call r_{min} the rank of c. The procedure described in



(a) 3D workspace view

(b) Slices through the workspace

Figure 3: The estimation of the 3D reachable workspace $\mathcal{W}_{reach}$ (green area) shown in different vertical and horizontal layers of top and front views.

the pseudocode below finds a random configuration and establishes its most general representation in the hierarchy.

In the "for" loop from Line 4 to Line 9, the algorithm computes the rank of the node by checking collisions of the hierarchical representations of the robot's occupancy. Once the minimal valid hierarchical node is established the configuration together with the computed rank is added to the set of nodes N (Line 11).

Whenever a new hierarchical node is found, we select a number of candidate nodes from the current set N and try to connect the new node to each of them. In addition to the connection computation performed by the traditional local planner, the rank of the edge should be established. For an edge e we look for the minimum hierarchy r_{min} such that $V(e^{r_{min}})$ is true along the edge.

The hierarchical node interconnection is built upon a local path locator and a hierarchy establisher. The local path locator returns an edge candidate, i.e. a local path that $\mathcal{A}$ can follow from one configuration to another. Then the hierarchy establisher checks if the edge candidate is collision free and meanwhile establishes the edge's most general representation in the hierarchy. The process of establishing the hierarchical node interconnection is outlined in Algorithm 2.

In line 3, the hierarchy r is initialized to be the maximum value of the ends of the edge. There is an obvious lemma according to the definition of the hierarchical edge connecting two nodes a^{r_1} and b^{r_2}:

Lemma 4: $V(e^r) = true \implies r \geq r_1 \wedge r \geq r_2$, i.e. the rank of an edge is not less than the rank of either end node of the edge. Algorithm 2 searches over the sequence of configurations on the edge for verification and hierarchy establishment. This general approach is straightforward to implement. Note that the hierarchy is established through the validation of the configurations. Similarly, to apply this strategy to other probabilistic motion planners it can be done when samples are checked for collision.

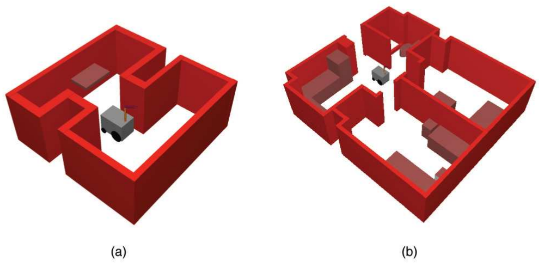


Figure 4: Experimental setup. (a) The mobile manipulator is placed in a simple 3D environment; (b) The mobile manipulator is placed in an apartment-like environment.

After the hierarchical roadmap $\mathcal{R}$ is built, $\mathcal{W}_{reach}$ can be computed from the connected component of $\mathcal{R}$ that contains the initial configuration of $\mathcal{A}$ through mapping function RA.

4 EXPERIMENTAL VALIDATION

We conducted experiments of our algorithm on the simulated 5-DOF mobile manipulator $\mathcal{A}$ shown in Fig. 1. First we provide an example that illustrates the hierarchical strategy described in the previous section. Fig. 2 provides details of the execution of the hierarchical PRM on this example. Initially $\mathcal{R}$ contains only one node that represents the initial configuration of $\mathcal{A}$. The rank of each randomly generated node is determined by looking for the most general occupancy representation of $\mathcal{A}$ that does not collide with any obstacles. Similarly the rank of each edge is determined by looking for the most general occupancy representation of $\mathcal{A}$ along the edge that does not collide with any obstacles. The top row of Fig. 2 shows tests for a randomly generated node. c^2 and c^3 generate collisions while c^4 does not, so this specific

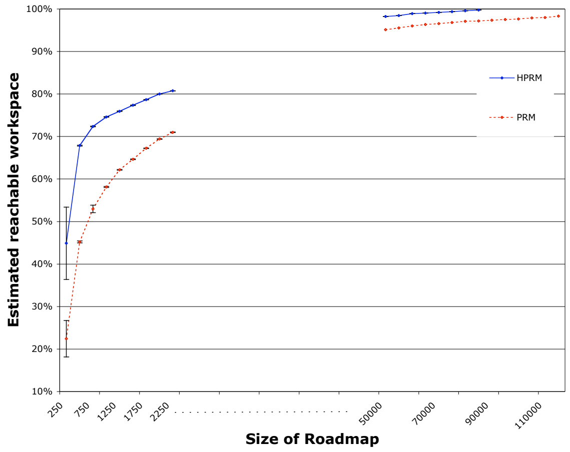


Figure 5: Experimental result for the environment shown in 4(a). The graph shows a comparison of workspace volume computed from hierarchical PRM and repetitive PRM sampling. Each data point plots the average of ten experiments. Standard deviations are plotted.

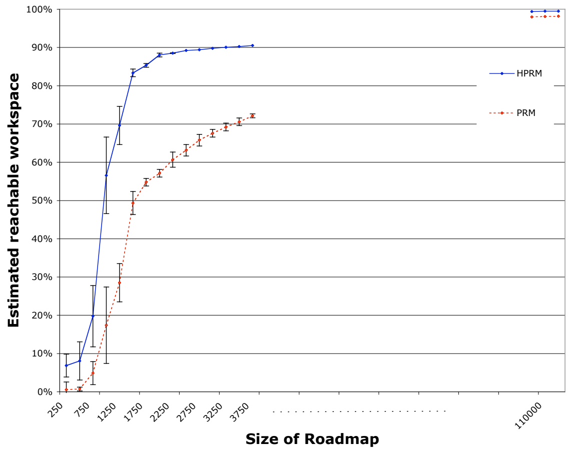


Figure 6: Experimental result for the environment shown in 4(b). The graph shows a comparison of workspace volume computed from hierarchical PRM and repetitive PRM sampling. Each data point plots the average of ten experiments. Standard deviations are plotted.

node is classified as c^4. The construction of $\mathcal{R}$ is incremental. The lower row shows incremental changes in $\mathcal{R}$. As more nodes are added both coverage and connectivity of $\mathcal{R}$ increases.

The connected component of the constructed $\mathcal{R}$ can be mapped to $\mathcal{W}$ such that $\mathcal{W}_{reach}$ is obtained. We use uniform cell decomposition to represent $\mathcal{W}$. Fig. 3 shows the estimated $\mathcal{W}_{reach}$ in different layers for the environment given in Fig. 4(a).

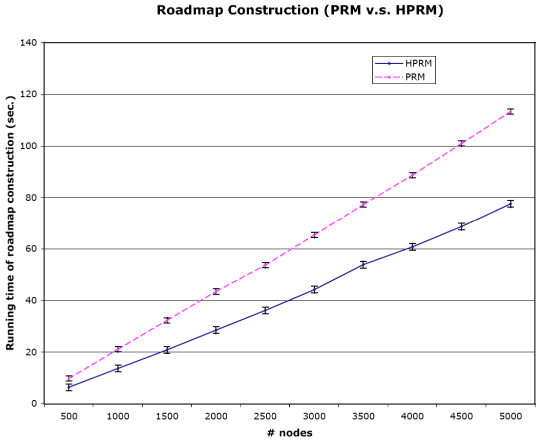


Figure 7: Comparison of the average running time of the construction of traditional roadmaps and the hierarchical ones. Averages are for 20 trials. Standard deviations are shown.

Fig. 5 and Fig. 6 show comparisons of the effectiveness of the hierarchical PRM and random sampling using repetitive PRM for reachable workspace estimation on the environments shown in Fig. 4 (a) and (b) respectively. The x-axis represents the number of samples used in the estimation. The y-axis represents the percentage of the estimated $\mathcal{W}_{reach}$ (the true $\mathcal{W}_{reach}$ of the robot was computed by brute force search). Because randomness is involved, the workspace volume for ten independent runs for each case were averaged. Standard deviations are also plotted. Both graphs show that the hierarchical approach is more effective than repetitive PRM sampling for reachable workspace estimation.

Finally we evaluate the time efficiency of our hierarchical approach. Figure 7 shows the comparison of the running time of the roadmap construction in the basic PRM and the hierarchical approach for the model given in 4(b). Because randomness is involved, running times for 20 independent runs for each case were averaged. The hierarchical PRM performed reasonably well in these experiments. As can be seen from the results, creating a hierarchical roadmap takes less time than creating a traditional roadmap. This is because the hierarchical PRM saves time in collision checking in easy regions.

5 SUMMARY AND FUTURE WORK

A hierarchical approach was presented for adapting probabilistic motion planners for reachable workspace estimation. Unlike traditional probabilistic motion planners that treat each DOF equally, we order the DOF's of the kinematic structure and consider a hierarchical approach to the planning task. Considering the characteristic of the reachable workspace estimation problem, this hierarchy exploration improves the planning process through two critical computations: occupational analysis and reachability analysis. Validation of configurations begins by doing fast tests on simple occupational representations and only progresses to more accurate (and more expensive) evaluations as necessary. Because randomness is involved it is hardly possible to estimate the entire reachable workspace from the probabilistic roadmap within reasonable time. However, by iteratively computing the maximal reachable workspace from each node and edge our hierarchical motion planner can be more effective in the computation process than the traditional ones.

The hierarchical workspace estimation algorithm is especially useful for mobile robots in environments with obstacles. Experiments were conducted on a simulated 5-DOF mobile manipulator in two 3D environments. Experiments show that the hierarchical approach can be an effective and efficient alternative to the repetitive PRM for reachable workspace estimation.

Our current hierarchical algorithm uses the coarse-to-fine hierarchical nature in the process of estimating the workspace. The hierarchical characteristic might also be employed in other aspects of motion planners. For example, one heuristic would be to let the established hierarchy lead the sampling process toward the boundaries of obstacles, i.e. to sample more densely near nodes with higher hierarchy labels than those with lower hierarchy labels.

We can also imagine a more sophisticated definition of reachable workspace which might involve establishing the number of configurations from which the kinematic structure can reach a given location. This might provide insights into different levels of reachability. A space for where there exists many reachable configurations should probably be considered more reachable than one with just a few.

ACKNOWLEDGEMENTS

The financial support of NSERC Canada is gratefully acknowledged.

REFERENCES

Alameldin, T., Badler, N. I., and Sobh, T. (1990). An adaptive and efficient system for computing the 3-d reachable workspace. In *IEEE International Conference on Systems Engineering*, pages 503–506.

Badescu, M. and Mavroidis, C. (2004). New performance indices and workspace analysis of reconfigurable hyper-redundant robotic arms. *The International Journal of Robotics Research*, 23:643–659.

Canny, J. F. (1988). *The Complexity of Robot Motion Planning*. MIT Press, Cambridge, MA.

Horsch, T., Schwarz, F., and Tolle, H. (1994). Motion planning for many degrees of freedom – random reflections at c-space obstacles. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA '94)*, pages 3318–3323.

Hsu, M.-S. and Kohli, D. (1987). Boundary surfaces and accessibility regions for regional structures of manipulators. *Mechanism and Machine Theory*, 22:277–289.

Kavraki, L. E., Svestka, P., Latombe, J.-C., and Overmars, M. (1996). Probabilistic roadmaps for path planning in high dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580.

Kumar, A. (1980). *Characterization of Manipulator Geometry*. PhD thesis, University of Houston.

Latombe, J.-C. (1991). *Robot Motion Planning*. Cluwer.

Lenarcic, J. and Umek, A. (1994). Simple model of human arm reachable workspace. *IEEE Transactions on Systems, Man and Cybernetics*, 24(8):1239–1246.

Morecki, A. and Knapczyk, J. (1999). *Basics of Robotics: Theory and Components of Manipulators and Robots*. SpringerWienNewYork.

Yang, J., Dymond, P., and Jenkin, M. (2008). Accessibility assessment via workspace estimation. *International Journal of Smart Home*, 3:73–90.

Zacharias, F., Borst, C., and Hirzinger, G. (2007). Capturing robot workspace structure: representing robot capabilities. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3229–3236.

AN ADAPTIVE CLASSIFIER DESIGN FOR ACCURATE SPEECH DATA CLASSIFICATION

Omid Dehzangi, Ehsan Younessian
Nanyang Technological University, Singapore
dehzangi@pmail.ntu.edu.sg, ehsa0001@ntu.edu.sg

Fariborz Hosseini Fard
SoundBuzz PTE LTD, Subsidiary of Motorola Inc., Singapore
cbfn87@motorola.com

Keywords: Nearest neighbor, Linear discriminant analysis, Adaptive distance measure, Weight learning algorithm.

Abstract: In this paper, an adaptive approach to designing accurate classifiers using Nearest Neighbor (NN) and Linear Discriminant Analysis (LDA) is proposed. A novel NN rule with an adaptive distance measure is proposed to classify input patterns. An iterative learning algorithm is employed to incorporate a local weight to the Euclidean distance measure that attempts to minimize the number of misclassified patterns in the training set. In case of data sets with highly overlapped classes, this may cause the classifier to increase its complexity and overfit. As a solution, LDA is considered as a popular feature extraction technique that aims at creating a feature space that best discriminates the data distributions and reduces overlaps between different classes of data. In this paper, an improved variation of LDA (im-LDA) is investigated which aims to moderate the effect of outlier classes. The proposed classifier design is evaluated by 6 standard data sets from UCI ML repository and eventually by TIMIT data set for framewise classification of speech data. The results show the effectiveness of the designed classifier using im-LDA with the proposed ad-NN method.

1 INTRODUCTION

The NN classifier is one of the oldest and the most successful methods of non-parametric pattern classification (Cover and Hart, 1998). However, it has some weaknesses in cases that patterns of different classes have overlap in some regions in the feature space. It also considers all the stored instances the same for classification, but the instances are different in being representative of their typical classes.

The performance of the NN classifier depends crucially on how to choose a suitable distance metric. Many methods have been developed to locally adapt the distance metrics such as the flexible metric method proposed in (Friedman, 1994), the discriminant adaptive method in (Hasti and Tibshirani, 1996) and the adaptive metric method in (Domeniconi et al., 2002). The common idea underlying these methods is that they estimate feature relevance locally at each query pattern. This leads to a weighted metric for computing the similarity between the query patterns and training data. In (Wang et. al., 2007), a simple locally adaptive distance measure is proposed that uses a heuristic measure to specify the weight of each training instance. The method we propose in this paper uses a locally adaptive metric to improve the performance of the basic NN classifier. An iterative learning algorithm is employed to incorporate a local weight to the Euclidean distance measure that attempts to minimize the number of misclassified patterns in the training set. In case of data sets with highly overlapped classes, examples in the overlapping area are considered to be noisy as for learning these examples. the learning algorithm would be in contradiction with other training examples or would need to increase its complexity in order to accommodate them. Learning these difficult examples may lead the algorithm to be unable to generalize well.

As a solution to this problem, linear discriminant analysis (LDA) is considered as one of the most traditional methods to find a linear feature transformation method, which maximizes the ratio of between-class scatter and the within-class scatter.

The earliest of such methods, Fisher's Linear Discriminant Analysis (LDA) (Fisher, 1936), tries to find a linear combination of input variables that best discriminates between different class distributions and is still a powerful technique for feature extraction to reduce overlaps between different classes of data (Duda and Hart, 2001). However, LDA does not take into account the conjunctions between different pairs of classes in a multi-class problem (Loog et al., 2001). In such cases, if one or more classes are far away from others (i.e. outlier classes), there is no need to maximize their between-class scatter covariances in the transformed space. Thus, they do not contribute in the estimation of the uniform between-class covariance (Jarchi and Boostani, 2006). In this paper, an improved version of LDA is investigated that redefines the between-class scatter matrix by integrating a simple weight into it. In the transformed feature space, different classes of data have lower degrees of overlap with one another. Then, our proposed ad-NN classifier can be applied to the input patterns in the new space with lesser risk of overfiting. In order to assess our method, combination of the im-LDA and the ad-NN are applied on eight UCI ML driven data sets. The proposed classifier design is also applied on TIMIT speech data set, attempting to classify huge amount of speech frames with 60 different phoneme classes.

2 ADAPTIVE DISTANCE USING WEIGHTED INSTANCES

The nearest neighbour classifier assigns label of a test pattern according to the class label of its nearest training instance. To introduce the weighted version of NN rule, here, the notation of basic NN rule is briefly described. Assume that classification of patterns in an m-dimensional space is under investigation. Having a set of training instances $\{(X_i,C_k)\}$, where X_i, i=1,…,n are training feature vectors and C_k ,k=1,…,M are the labels. NN rule finds the nearest neighbor of a new test pattern X using a distance function and assigns X to C_w (the class label of the winner class). The Euclidean distance have been conventionally used to measure the distance between X and Y:

$$\mathrm{d}(X,Y)=\sqrt{\sum_{i=1}^{m}(x_i - y_i)^2} \qquad (1)$$

In the first step, the distance as a dissimilarity measure between query pattern X and the j^{th} instance X_j is changed to a similarity measure. This is done by a linear conversion as follows:

$$\mu(X,X_j) = 1 - \mathrm{d}(X,X_j) / \sqrt{m} \qquad (2)$$

where, $\sqrt{m}$ is the maximum distance which can ever occur Between two training instances in the whole training set, since the data is normalized in the range of [0, 1] in the first place. As a result, instead of finding minimum distance pattern to X, we search for instance X_j such that $\mu(X,X_j)$ is maximized. This can be interpreted as normalizing the distance from [$\sqrt{m}$, 0] to a real number in the interval [0, 1]. Now, we can introduce the weighted nearest neighbor rule. The pattern is classified according to consequent class of the winner instance X_{win}. The winner instance is specified using:

$$win = \underset{j=1,\dots,n}{\operatorname{argmax}} \left\{ \mu(X,X_j) \,.\, w_j \right\} \qquad (3)$$

where w_j is the weight assigned to the j^{th} instance by the learning algorithm which is introduced in section 4.

2.1 Instance Weighting Algorithm

For an M-class problem, assume that n labeled patterns $\{X_i$, i=1,2,…,$n\}$ are available from various classes. In this section, we propose an algorithm to learn the weight of each pattern using other labeled training instances. The calculated weight is optimal in the sense that it maximizes the classification rate of the classifier on the training data. At the beginning, a weight of one is assigned to each pattern in the whole training set (i.e. w_k=1, k=1,..., n).

In the following, the weight of each instance is specified assuming that the weights of all other training instances are given and fixed. The weight w_k of instance k with class label of Q can be found as follows. First, the problem is considered a 2-class problem of class Q as positive "p" and class $\bar{Q}$ as negative "n". The w_k is set to zero (i.e., instance X_k does not contribute in classification decision). Given current weights of all other training instances and a training set of P positive and N negative labeled patterns, the decision results of the weighted instance NN classifier described in section 3 can be grouped into four categories: true positives (TPs) denoting samples correctly labelled as positives; false positives (FPs) denoting negative samples incorrectly labeled as positives; true negatives (TNs) denoting samples correctly labeled as negatives; and false negatives (FNs) denoting positive samples incorrectly labeled as negatives.

Training patterns of class Q which are classified correctly with the current values of instance weights are removed from the training set. These patterns will be classified correctly regardless of the value of w_k. Similarly, training patterns of class $\bar{Q}$ which are misclassified with the current assigned weights are also removed from the training set. These patterns will be misclassified regardless of the value of w_k. For each training pattern left in the training set (i.e. instances in TN and FN), a score is calculated using the following measure:

$$S(x_t)=\frac{\max\{\mu(X_t,X_j).w_j \mid j=1,2,...,\mathrm{n}, j\neq t\}}{\mu(X_t,X_k)} \qquad (4)$$

where $\mu(X_i,X_j)$ represents the similarity of patterns X_i and X_j calculated using (2).

Those instances that have X_k as their nearest neighbor are called associates of X_k. It can be shown that X_t is an associate of X_k, if $S(X_t)$ is less than w_k. We have,

$$w_k > S(X_t) \Rightarrow w_k \cdot \mu(X_t,X_k) > S(X_t)\cdot \mu(X_t,X_k) \qquad (5)$$

From (4) we have,

$$w_k.\mu(X_t,X_k) > \max\{\mu(X_t,X_j).w_j \mid j=1,...,n \quad j\neq t,k\} \qquad (6)$$

From (3) and (6), it is concluded that X_k is the winner instance to classify X_t. Given the weighted instance NN classifier introduced in previous section, associate set of instance X_k can be defined fomally as,

$$\begin{array}{l} Associate\_set(X_k,w_k)= \\ \{X_t \mid t=1,...,n \quad t\neq k \quad w_k > S(X_t)\} \end{array} \qquad (7)$$

By altering w_k, the associate set of X_k is changed that cause modification in classification error rate of the classifier. Our aim is to determine w_k such that the error-rate of the classifier on training set is minimized given that the weights of all other training instances are given and fixed. We define an accuracy measure as,

$$Accuracy = \mathrm{TP} - \mathrm{FP} \qquad (8)$$

We try to find a proper w_k such that associate set of X_k includes more FN instances which need to be classified as “p” and exclude more TN instances which are the instances of class “n”. The optimal weight of the instance X_k is calculated by maximizing *Accuracy* measure assuming that the weights of all other instances are given and fixed. To do this, the set of patterns X_t are ranked in ascending order of their scores. We define a threshold initialized with zero. Then, assuming that X_t and X_{t+1} are two successive patterns in the ranked list, a threshold is computed as,

$$th = (\mathrm{Score}(X_t) + \mathrm{Score}(X_t{+}1))/2 \qquad (9)$$

The threshold is then altered from the least score to the greatest and associated accuracy of the classifier with respect to the each threshold is measured. The value of the best threshold (i.e. leading to the best accuracy) is simply used as the weight of the instance X_k. The proposed instance weighting mechanism assigns a weight to each instance attempting to better discriminate between the patterns of the same class and patterns of all other classes. The search for the best combination of instance weights is conducted by optimizing each instance in turn assuming that the order of the instances to be optimized is fixed.

3 OVERVIEW OF LDA

The proposed algorithm in section 2.1 to learn the weight of each instance attempts to minimize the classification error in the training data. In case of data sets with highly overlapped classes, examples in the overlapping area are considered to be noisy as for learning these examples. Learning these difficult examples may lead the algorithm to be unable to generalize well. The goal of LDA is to find an optimal linear transformation of input feature vectors such that the class separability in the new space is maximized. In order to find an optimal linear discrimination transform, Fisher (Fisher, 1936) proposed a criterion that maximizes the ratio of between-class to within-class scatter matrices. The aim is to look for a linear discriminant transform w_{LDA} ,

$$w_{LDA} = \arg\max_{w}\{\varphi_{ij}\} \ , \qquad (10)$$

where, φ_{ij} is the Fisher criterion that is determined as follows:

$$\varphi_{ij} = \frac{|\ W^T S_B W\ |}{|\ W^T S_W W\ |} \qquad (11)$$

where S_B is between-class scatter matrix and S_W is within-class scatter matrix. Columns of W_{LDA} are eigenvectors corresponding to the (C-1) largest eigenvalues of $S_w{}^{-1}S_B$. Now, each sample X in data set can be transformed to a new space by multiplying to the matrix w_{LDA} :

$$Y = w_{LDA}{}^{T}.X \qquad (12)$$

Y is a new transformed sample corresponds to X.

3.1 The Improved LDA Method

By Projection of data in a lower dimensional space, LDA can also reduce the computation complexity caused by redundant information in the data which is useful for solving classification problems. However, LDA has some weaknesses. LDA considers all the classes the same to calculate between-class scatter matrix. If one or more classes are outliers, there is no need to maximize their between-class scatter covariance in the transformed space. Therefore, different weights should be integrated in the covariance estimation procedure. The aim in the weighted version of LDA is to alleviate the role of an outlier class. This is done by redefining between-class scatter matrix.

Assume that we use the between-class scatter matrix definition that is based on (Loog et al., 2001):

$$S_B = \sum_{i=1}^{C-1} \sum_{j=i+1}^{C} P_i\, P_j\, (\mu_i - \mu_j)(\mu_i - \mu_j)^T \qquad (13)$$

where, C is the number of classes, p_i is a prior probability for class i and p_j is a prior probability for class j, μ_i is the mean of training samples of class i.

Incorporating a weight $1/\varphi_{ij}$ into between-class scatter matrix leads to a new formula:

$$S_B = \frac{1}{N^2} \sum_{i=1}^{C-1} \sum_{j=i+1}^{C} 1/\varphi_{ij}\, N_i\, N_j\, (\mu_i - \mu_j)(\mu_i - \mu_j)^T \qquad (14)$$

where, φ_{ij} is Fisher discrimination value of the resulting w_{LDA} that is determined in (11), N is the number of training samples, N_i and N_j are the number of training samples of class i and j, respectively. By incorporating this weight, the farther the classes are from each other, the less their contribution is in the between class scatter matrix calculation.

4 EXPERIMENTAL RESULTS

In order to evaluate the performance of the designed classifier, two sets of experiments have been conducted.

4.1 Experiments using Standard Data

First, a number of standard data sets are used which derived from UCI ML repository (Merz and Murphy, 1996). Some statistics of the data sets are shown in Table 1.

Table 1: Statistics of the data sets used in our experiments.

Data set	# of attributes	# of patterns	# of Classes
Pima	8	768	2
Wine	13	178	3
Hepatitis	19	155	2
Image Seg.	15	210	7
Balance	4	625	3
Heart Clev.	13	303	5

To evaluate the classification accuracy on the data sets, the average result of ten trials of ten-fold cross validation is reported. Results, which are shown in the Table 2, illustrate that our proposed method is led to the best performance on all the data sets compared to basic combinations.

Table 2: Error rates of combination of LDA and NN along with our proposed method on 6 UCI ML data sets.

Data set	Basic NN	LDA+ Basic NN	Im-LDA+ Basic NN	Im-LDA+ ad-NN
Pima	29.06	25.96	25.54	22.47
Wine	5.09	3.23	2.88	2.32
Hepatitis	19.87	14.47	14.33	12.89
Image Seg.	8.63	7.48	6.21	5.32
Balance	18.69	17.51	15.67	14.19
Heart Clev	20.75	19.26	16.92	15.29
Ave. error	17.02	14.65	13.59	**12.08**

As it can be seen in Table 2, im-LDA outperforms LDA in multi-class data sets. This shows the effectiveness of the weight incorporated in the definition of between-class scatter matrix. Table 2 also shows that in the transformed feature space using im-LDA where different classes of data have lower degrees of overlap, ad-NN with adaptive local distance measure clearly improves the generalization ability of the basic NN.

4.2 Experiments using Real Data

In this section, we validate the proposed method on the TIMIT corpus (Garofolo, 1988) because of its high-quality phone labels. All results reported are framewise classification error rates for TIMIT complete test set (the 1344 si and sx sentences). The speech waveforms are parameterized by a standard Mel-Frequency Cepstral Coefficient (MFCC) front end. The cepstral analysis uses a 25 msec Hamming window with a frame shift of 10 msec. Each input pattern X_i consists of the current frame of 12 MFCCs and energy plus delta and acceleration coefficients, and two context frames on each side, making a total of (13 + 13 + 13) * 5 = 195 components. This formulation was arrived at by experimentation with varying numbers of context frames left and right of the frame being classified. The training set has about 1.1 million frames and the test set has about 400 thousand frames. Each frame has an associated 1-of-60 phonetic label derived from the TIMIT label files.

Due to the large number of training data and large number of classes, TIMIT data set seems to be a suitable task to evaluate our proposed classifier. In Table 3, framewise classification error rates on the TIMIT test set using our classifier is compared to the existing methods.

Table 3: Framewise phoneme classification error rate on TIMIT test set.

Classifier	Error Rate
Recurrent Neural Nets (Schuster, 1997)	34.7%
Bidirectional LSTM (Graves, 2005)	29.8%
im-LDA + ad-NN	28.7%

The results show that the proposed classifier design outperforms previous works in classification of speech frames on TIMIT task.

5 CONCLUSIONS

In this paper, a novel classifier design based on combination of an improved version of LDA and an adaptive distance NN was presented. LDA, as a preprocessing step, was used to transform input data to a new feature space in which different classes of data has lower degrees of overlap. In the classification step, a novel learning algorithm was used to assign a weight to each stored instance, which was then contributed in the distance measure, with the goal of improvement in generalization ability of the basic NN. In this way, different weighs were given to the transformed samples based on a learning scheme which optimized the weights according to the classification error rate. Our proposed method was evaluated by various UCI ML data sets. Results showed that the proposed method improves the generalization ability of basic NN. By using TIMIT speech data set, the effectiveness of our approach in real problems like speech data classification was also proved.

REFERENCES

Cover, T.M., Hart, P.E., 1967. Nearest Neighbor Pattern Classification. *IEEE Transaction on Information Theory* 13, 21-27.

Friedman, J., 1994. Flexible metric nearest neighbor classification. Technical Report 113, Stanford University Statistics Department.

Hastie, T., Tibshirani, R., 1996. Discriminant adaptive nearest neighbor classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18: 607-615.

Domeniconi, C., Peng, J., Gunopulos, D., 2002. Locally adaptive metric nearest neighbor classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24: 1281-1285.

Wang, J., Neskovic, P., Cooper, L.N., 2007. Improving nearest neighbor rule with a simple adaptive distance measure. *Pattern Recognition Letters*, 28: 207-213.

Fisher, R.A., 1936. The Use of Multiple Measurements in Taxonomic Problems, *Annals of Eugenics*, 7:179-188.

Duda, R.O., Hart, P.E., Stork, D., 2001. *Pattern Classification 2nd Edition.* Wiley, New York.

Loog, M., Duin, R.P.W., Haeb-Umbach, R., 2001. Multiclass linear dimension reduction by weighted pairwise fisher criteria, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23: 762-766.

Jarchi, D., Boostani, R., 2006. A New Weighted LDA Method in Comparison to Some Versions of LDA, *Transaction on Engineering and Computational Technology*, 18: 18-45.

Garofolo, J.S., 1988. *Getting started with the DARPA TIMIT CD-ROM: An acoustic phonetic continuous speech database*, National Institute of Standards and Technology (NIST), Gaithersburgh, MD.

Merz, C.J., Murphy, P.M., 1996. *UCIRepository of Machine Learning Databases*. Irvine, CA: University of California Irvine, Department of information and Computer Science. Internet: http://www.ics.uci.edu/~mlearn/MLRepository.html

Schuster, M., Paliwal, K.K., 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45: 2673-2681.

Graves, A., Schmidhuber, J., 2005. Framewise Phoneme Classification with Bidirectional LSTM and Other Neural Network Architectures. *International Joint Conference on Neural Networks.*

THREEDIMENSIONAL TRACKING USING OBJECT DEFOCUS

In Twodimensional Scanning Electron Microscope Images

Christian Dahmen

Division Microrobotics and Control Engineering, University of Oldenburg, Oldenburg, Germany

christian.dahmen@uni-oldenburg.de

Keywords: Image processing, Object tracking, Depth estimation, Focus analysis.

Abstract: This paper proposes a tracking algorithm for the extraction of threedimensional position data from SEM images. An algorithm based on active contours with region-based minimization is chosen as basis for twodimensional tracking. This algorithm is then augmented by the incorporation of defocus analysis to estimate the out-of-focus displacement of the object. To solve the ambiguity of the out-of-focus displacement, the astigmatism of the SEM images is used. The separate calculation of variances for the rows and columns of the image enables a successful direction estimation. With the information on the direction, the out-of-focus displacement and the working distance of the acquired image, the distance of the object to the electron gun can be calculated. In combination with the twodimensional part of the tracking, a full threedimensional coordinate set is generated. The approach is tested and evaluated using a positioning setup and the principal feasibility is shown.

1 INTRODUCTION

The SEM has been widely used as a imaging tool for the automated handling of micro- and nanoscale objects (see e.g. (Fatikow, 2007)). While there have been many manipulations executed manually, and with the necessary experience these manipulations have a high success rate, fully automated handling and manipulation of micro- or even nanoscale objects in the SEM still is very rarely encountered. The problems which have to be solved are manifold and not easy to handle. Integrated sensors in the actuators or the setup may deliver some information about the positions of end-effectors and tools, and enable estimates about the position of objects to be manipulated. The real actual position information of objects or tools though is difficult to recover from this data, because of various factors like thermal drift, play or object interactions on the nanoscale. A specific requirement for non-teleoperated processes is therefore the need for sensor feedback generation based on SEM images. The SEM is the only sensor which may deliver an overview over the whole scene, enabling the extraction of the positions of most or all objects engaged in the manipulation. Only the SEM determines these positions in a common coordinate system, which makes it possible to evaluate relative distances between the objects. One problem which is crucial for the success of any automation approach is the missing information about the position of objects in z-direction, which means orthogonal to the image plane. An example image illustrating the problem is shown in figure 1.

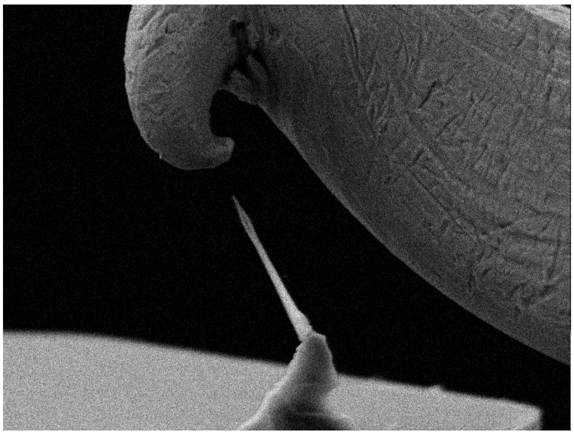


Figure 1: A screenshot showing two objects having different z-positions. One object is a silicon nanowire, the other a deformed STM tip. The distance between the objects in z-direction is not immediately visible.

2 STATE OF THE ART

Different algorithms have been described in the literature extracting the twodimensional position of objects from SEM images for automation purposes. The performance of these algorithms is good, making first simple automation scenarios possible. The approaches used for twodimensional tracking on SEM images and their possible extensions to 3D-tracking will be summarized in the following.

One of the first and most simple approaches used template matching as the basis of the algorithm (Sievers and Fatikow, 2006). A template image is extracted or loaded which contains the object to be tracked. The template image is then cross-correlated with a search area in the input image. The maximum value of the resulting array is in the place where the template is most likely to be found. Due to the use of cross correlation, this approach is very robust against additive noise, which is an advantage especially for fast scanned SEM images. Problems of this approach are that the algorithm is sensitive against certain changes in the object appearance which may occur during handling processes. Examples for these appearance changes are rotation of the object, scaling of the object due to magnification changes and partial occlusion by other objects in the setup. Removing these weaknesses for this method comes with increased computational effort so that a fast enough calculation is not always possible. Extraction of the z-position is not featured and cannot easily be added.

If instead of a template image a CAD model of the object is available, it is possible to use rigid body models to track the object in the SEM image (Kratochvil et al., 2007). The implementation uses measurement lines orthogonal to the model edges and tries to fit the model to visible edges in the image. Model edges which should be invisible are identified and not used for the pose estimation. Though edge detection is difficult in noisy SEM images, the approach yields good results using advanced techniques for discarding or outweighting false edges and through the high number of measurement lines used. When threedimensional CAD-models are used, it is possible to recover the threedimensional pose including in-plane and out-of-plane rotations, except the z-position. The extension for true threedimensional tracking relies on a model of the SEM image projection to yield the z-component of the position. This seems to be working for low magnifications.

Another possibility is the use of active contours or snakes (for details about this concept see (Blake and Isard, 2000)), which do not rely on much preexisting knowledge about the object. Active contours are parametrized curves in twodimensional space, that means in the image plane. After coarse initialization the contour is evolving to segment the object from the scene. The contours are coupled with an energy function dependent on their shape or appearance, and on the image data. This energy function is being minimized by moving contour points or the contour as a whole. The part dependent on the contour is called internal energy, the part dependent on the image data is the external energy. In the original formulation, the external energy function was defined to be dependent on the distance of the contour from edges in the image, as explained in (Kass et al., 1988). For the use with noisy SEM images, a region-based approach (see (Sievers, 2006) and (Sievers, 2007)) has shown to be useful. The external energy function here is dependent on the region statistics and the noise characteristics of the imaging source. The goal is to maximize the compound probability of the enclosed region. This approach has proven to be very robust to additive noise, and is inherently robust against scaling and rotation. If the contour minimization is restricted to the euclidean transform space, robustness against partial occlusion is added. Due to the model-free nature of this approach, threedimensional tracking is not immediately possible, but the coupling with focus-based methods is principally possible and shows first promising results in the SEM.

In this paper, the last tracking approach is taken as a basis, and extended to use defocus analysis for depth estimation. The extracted information is only the z-position of the tracked object, without any structural information about the object. For the recovery of threedimensional structure of objects, different methods may be used (see e.g. (Fernandez et al., 2006) or (Jähnisch and Fatikow, 2007)).

3 PRINCIPLE

The principle of the twodimensional tracking has been explained already in (Sievers, 2006). The important aspect is that the active contour algorithm does not only deliver the position information of the object, but at the same time calculates a segmentation of the object from the rest of the scene. This enables further analysis of the enclosed object.

Due to the working principle of the SEM, only a certain range around the set working distance is depicted sharp. Though this range is quite big in comparison to optical microscopes, defocusing is still evident, as can be seen in figure 2. The defocusing in the SEM has been used already in (Eichhorn et al., 2008) to determine the z-position of objects by generating a sharpness curve over a certain working distance range. One drawback is the amount of time needed to obtain this image sequence and therefore also the disability to monitor dynamic processes. But by analyzing the sharpness of the object which is of interest, it is also possible to directly conclude to the out-of-focus displacement of the object, as has been recently demonstrated for the SEM in (Dahmen, 2008).

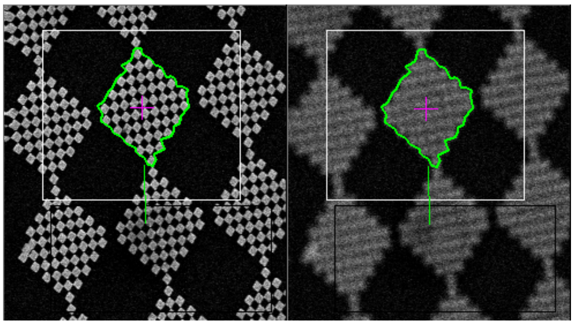


Figure 2: Comparison of object in focus and out of focus.

The sharpness measure used here is the grey-level variance

$$\sigma^2(I,A(C)) = \frac{1}{N_{A(C)}} \sum_{p\in C} I(p) - \bar{I}(A(C)) \qquad (1)$$

with

$$\bar{I}(A(C)) = \frac{1}{N_{A(C)}} \sum_{p\in A(C)} I(p)), \qquad (2)$$

and C the contour, $A(C)$ the enclosed area and I the image.

For the object enclosed by the contour this means

$$\sigma^2(I(WD),A(C)) = Max \Rightarrow z(Object) = WD \qquad (3)$$

with the working distance WD and the z-position of the Object enclosed by the contour $z(Object)$.

One problem which persists after this analysis still is that the out-of-focus displacement turns out to be ambiguous. Two possibilities exist for the solution of this problem, one assuming that the object is nearer than the working distance, the other assuming that the object is further away. This is a situation which is not optimal for automation purposes. Though in some cases the correct solution may be determined by the setup and context knowledge, it is desirable to determine the solution without additional information apart from the image. For this case, a normally undesired effect in SEM imaging may be taken advantage of.

During normal use of the SEM, astigmatism is something that is being diminished or removed by astigmatism correction. Astigmatism in the SEM leads to blurry images for the user. An important property of astigmatism is that the sharpness is direction dependent. The focal points are different for two perpendicular directions, as can be seen in figure 3. If we name the two perpendicular directions w_0 and w_1:

$$WD_{W_0} \neq WD_{W_1} \qquad (4)$$

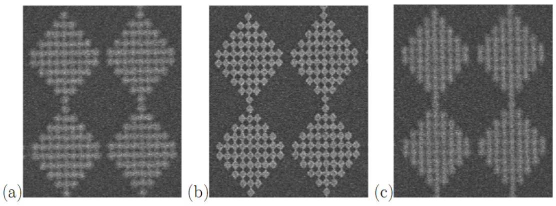


Figure 3: Astigmatism in the SEM, left and right slightly out of focus, middle focused.

To take advantage of this, we calculate additionally to the grey level variance of the image the grey level variance of the rows and columns separately by using

$$\sigma_x^2(I,A(C)) = \sum_y \frac{1}{N_{A(C)}(y)} \sum_{p(x,y)\in C} I(p) - \bar{I}(A(C),y) \qquad (5)$$

with

$$\bar{I}(A(C),y) = \frac{1}{N_{A(C)}(y)} \sum_{p(x,y)\in A(C)} I(p)) \qquad (6)$$

and

$$\sigma_y^2(I,A(C)) = \sum_x \frac{1}{N_{A(C)}(x)} \sum_{p(x,y)\in C} I(p) - \bar{I}(A(C),x) \qquad (7)$$

with

$$\bar{I}(A(C),x) = \frac{1}{N_{A(C)}(x)} \sum_{p(x,y)\in A(C)} I(p)) \qquad (8)$$

and C the contour, $A(C)$ the enclosed area and I the image.

The two values are normalized to their maximum, which is determined during the initialization phase (see section 4):

$$\hat{\sigma}_y^2(I,A(C)) = \frac{\sigma_y^2(I,A(C))}{max(\sigma_y^2(I,A(C)))} \qquad (9)$$

and

$$\hat{\sigma}_x^2(I,A(C)) = \frac{\sigma_x^2(I,A(C))}{max(\sigma_x^2(I,A(C)))} \qquad (10)$$

In this case we expect a working distance sweep to generate two slightly displaced curves like depicted in figure 4, under the assumption that the object in the image has suitable structure.

With the two curves having slightly displaced maxima, we can estimate from the ratio of the two values

$$r = \frac{\hat{\sigma}_x^2(I,A(C))}{\hat{\sigma}_y^2(I,A(C))} \tag{11}$$

to which side the out-of-focus displacement occurs.

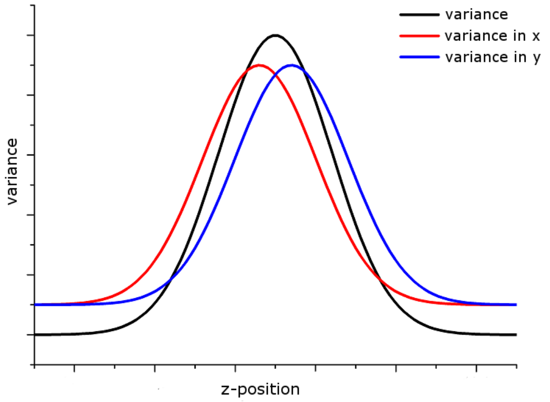


Figure 4: Expected curves for the different sharpness measures.

4 THE ALGORITHM

The twodimensional base algorithm is similar to the active contour algorithm described in section 2. The principle is depicted in figure 5. Important is that after an initial free minimization, the tracker does only translate and rotate the contour as a whole. This is necessary to enable the out-of-focus estimator to work, because else the enclosed area may change. The same is valid for scaling, which is therefore disabled in the algorithm. In order to enable continuous threedimensional tracking, the algorithm has to be initialized like shown in figure 6. An initial working distance sweep is carried out to acquire the characteristic curve of the tracked object. During this sweep, the twodimensional tracking has to be enabled already. After this, certain calculations and optimizations are carried out on the acquired curves, e.g. it is made sure that the curves are monotonic in sections. After the data has been processed, the tracker can track continuously until either object changes or imaging changes require a reinitialization.

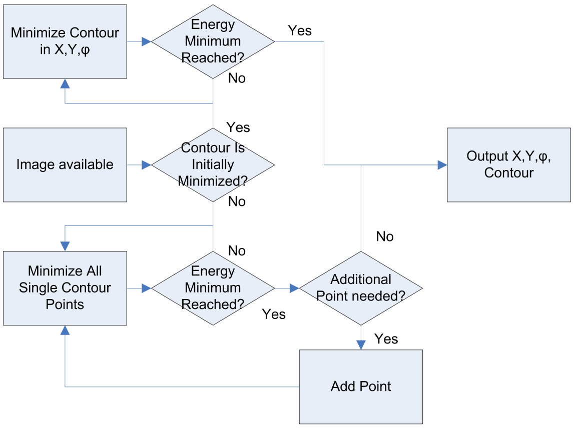


Figure 5: The basic tracking algorithm for the twodimensional part.

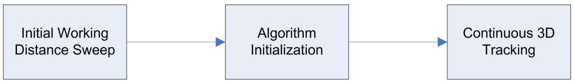


Figure 6: The initialization steps for the threedimensional tracking.

The threedimensional tracking itself consists of the twodimensional tracking algorithm, augmented with a sharpness calculation component, an out-of-focus displacement estimator and a direction estimator, like shown in figure 7. The active contour tracker delivers the twodimensional position and the segmentation of the object. This segmentation is then used to mask the original image. From the masked image, the object sharpness is calculated using variance calculation and the directional sharpness measures mentioned in the last section.

The variance value is used to estimate the out-of-focus displacement by comparison with the data acquired during initialization.

The directional sharpness values are used to estimate on which sidelobe of the initially acquired sharpness curve the object is in the actual image. After this is known, the information about the displacement value and the displacement direction is combined with the working distance at which the current image was captured. The result is the estimated working distance at which the object is placed. This is then joined with the position information from the twodimensional tracking to generate a complete coordinate set.

5 EXPERIMENTS

For the evaluation of the algorithm, the performance has been tested in a setup inside the SEM. Target object was a chessboard pattern as seen in figure 8. This chessboard pattern has been put on a stub which is mounted to a XYZ positioning setup using piezo slip-stick actuated axes with internal position sensors. For the experiment, the internal sensors of the axes were read out and the acquired data joined with the track-

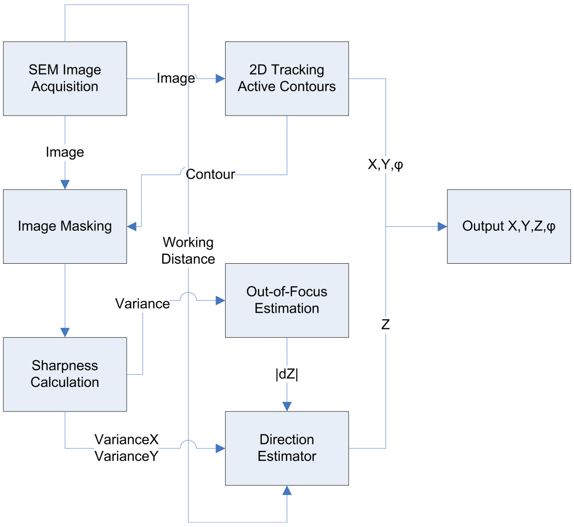


Figure 7: The complete 3D tracking algorithm in the initialized state.

ing data. During the experiment the axes were moved in closed loop mode using the internal sensors. The movement was in a pattern to verify the algorithm performance. All measurements have been executed at a magnification of 800x and a scanspeed of 5 (frame averaging of 2^5 frames) on a LEO scanning electron microscope. The twodimensional pattern of the movement has a size of $21\mu m$ times $39\mu m$. The pattern has been repeated at the z-axis positions of $0.17mm$, $0.256mm$ and $0.32mm$.

6 RESULTS

The first result was the generation of the expected curves from figure 4. The result can be seen in figure 9. The shift of the variance calculated in rows and columns separately is clearly visible, which enables the algorithm to work in the anticipated way, estimating the direction of the defocusing.

As can be seen in figure 10, the movement pattern used incorporated movement along each axis as well as in diagonal. Also visible is the distortion of the shape in comparison to the acquired sensor values. The reason for this was determined in additional tests to be a decalibrated sensor of one of the axes. This sensor did not deliver the correct position value, resulting in the closed loop control positioning to a wrong real axis position.

In figure 11 and figure 12 this gets more obvious. While the tracking position in figure 12 closely follows the sensor information, this is not the case in figure 11. Apart from the decalibrated sensor it can be stated that the twodimensional tracking is working.

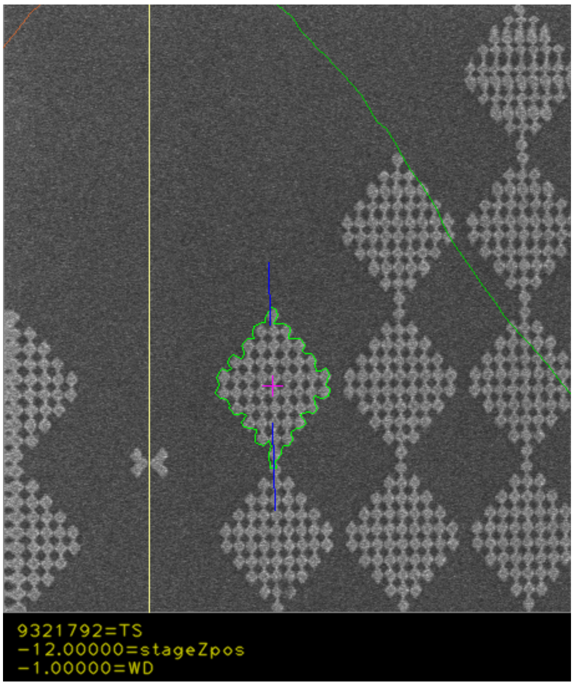

9321792=TS
-12.00000=stageZpos
-1.00000=WD

Figure 8: The structure used for the experiment, a chessboard pattern mounted on a XYZ positioning setup. Visible is the active contour tracking and segmenting one chessboard block.

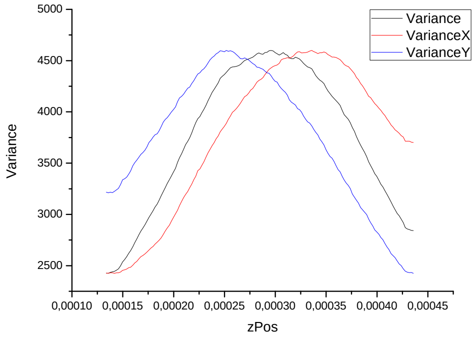


Figure 9: Measured curves for the variance and the variance calculated in rows and columns separately.

In order to verify the tracking, the experiment was repeated with a working and calibrated y-axis. The twodimensional tracking can be seen in figure 13. The shape of the movement is correct and the tracking is working and stable.

Figure 14 shows the determined z-Position from the tracking algorithm. Visible is that there is a certain systematic error in the tracked position, as the calculated values deviate from the expected values. The set working distance of the SEM is in the middle of the working distance range shown in figure 14, at the

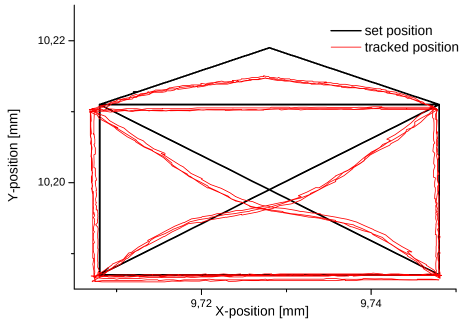


Figure 10: Tracked X-Y position of the chessboard pattern. The distorted shape and deviations are due to a decalibrated actuated axis for the y-direction.

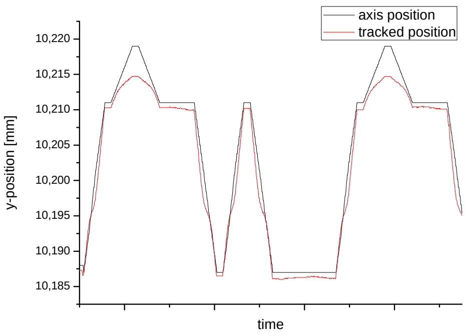


Figure 11: The tracked and set y-position over time.

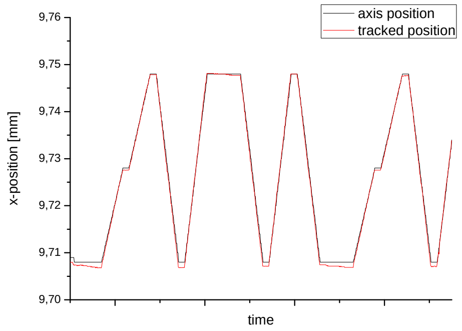


Figure 12: The tracked and set x-position over time.

value 0.28. The problem which occurs here is that the algorithm is most accurate not in the point of maximal focus, but within a certain range on the sidelobes of the sharpness curve. As can be seen in figure 9, the sharpness curve is relatively flat around the maximum. In this interval around the maximum, small changes in detected sharpness, which may also occur due to time variant behavior of the SEM imaging process or due to certain changes in the surrounding setup, will result in large errors in the estimated out-of-focus displacement. This explains also not only the big deviation from the axis set value, but also the large amount of variation during the movement on the same z-position. So it has to be stated that the optimal working condition for this algorithm is a slightly defocused image.

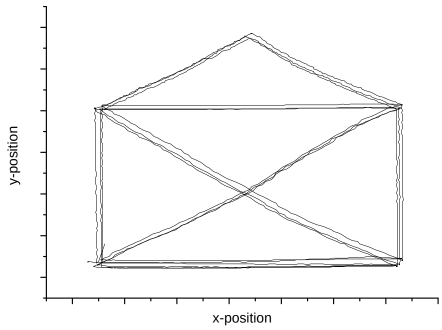


Figure 13: Tracked X-Y position of the chessboard pattern in the repeated measurement.

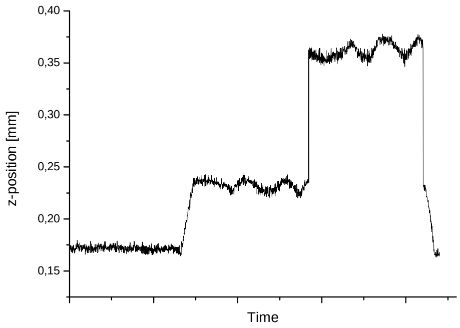


Figure 14: The tracked z-position over time.

In figure 15 the tracking result can be seen in three dimensions. The movement pattern is qualitatively visible, though the tracking in z-direction is not as good as in the image plane. Still the goal of the algorithm design has been reached, an estimate has been calculated for the z-position of the object which is principally useable.

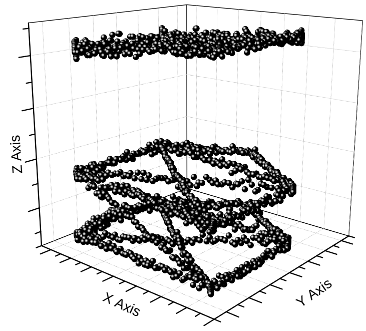


Figure 15: The tracked movement in 3D.

7 SUMMARY

In this paper a threedimensional tracking algorithm for the tracking of objects in the SEM has been presented. The algorithm takes advantage of the image defocusing which is evident when objects leave the focal plane. A twodimensional tracking algorithm based on active contours with a region-based minimization has been taken as the base algorithm. Added extensions include the segmentation of the object and the consecutive sharpness calculation. Additionally the variance of the rows and columns is calculated separately for determining the direction of the defocusing. This enables the analysis of the sharpness in different directions. If the image contains astigmatism, it is possible to estimate if the image focal plane lies in front of or behind the object. Experiments have shown that this approach is working and after an initialization phase qualitatively delivers a threedimensional position information. The z-position still contains a systematic error, which is most influential around the best focused point. This error has to be diminished by further analysis and change and optimization of the implementation. Overall the feasibility of this threedimensional tracking algorithm has been shown.

REFERENCES

Blake, A. and Isard, M. (2000). *Active Contours*. Springer.

Dahmen, C. (2008). Focus-based depth estimation in the sem. In *Proceedings of the International Symposium on Optomechatronic Technologies 2008*, volume 7266, page 72661O. SPIE.

Eichhorn, V., Fatikow, S., Wich, T., Dahmen, C., Sievers, T., Andersen, K., Carlson, K., and Bøggild, P. (2008). Depth-detection methods for microgripper based cnt manipulation in a scanning electron microscope. *Journal of Micro-Nano Mechatronics*. accepted.

Fatikow, S. (2007). *Automated Nanohandling by Microrobots*.

Fernandez, J., Sorzano, C., Marabini, R., and Carazo, J. (2006). Image processing and 3-D reconstruction in electron microscopy. *Signal Processing Magazine, IEEE*, 23(3):84–94.

Jähnisch, M. and Fatikow, S. (2007). 3d vision feedback for nanohandling monitoring in a scanning electron microscope. *International Journal of Optomechatronics*, 1(1):4–26.

Kass, M., Witkin, A., and Terzopoulos, D. (1988). Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331.

Kratochvil, B. E., Dong, L. X., and Nelson, B. J. (2007). Real-time rigid-body visual tracking in a scanning electron microscope. In *Proc. of the 7th IEEE Conf. on Nanotechnology (IEEE-NANO2007), Hong Kong, China*.

Sievers, T. (2006). Global sensor feedback for automatic nanohandling inside a scanning electron microscope. In *Proc. of 2nd I*PROMS NoE Virtual International Conference on Intelligent Production Machines and Systems*, pages 289–294.

Sievers, T. (2007). *Echtzeit-Objektverfolgung im Rasterelektronenmikroskop*. PhD thesis, University of Oldenburg.

Sievers, T. and Fatikow, S. (2006). Real-time object tracking for the robot-based nanohandling in a scanning electron microscope. *Journal of Micromechatronics*, 3(3-4):267–284.

AUTOMATIC EXTRACTION OF DISTANCE INFORMATION FROM STEREO IMAGES USING TRACKING DATA
In the Scanning Electron Microscope

Christian Dahmen, Tim Wortmann, Robert Tunnell and Sergej Fatikow
Division Microrobotics and Control Engineering, University of Oldenburg, Oldenburg, Germany
christian.dahmen@uni-oldenburg.de

Keywords: SEM image processing, Stereo, Depth detection, Active contours.

Abstract: This paper describes an approach to depth detection in scanning electron microscope images which aims at the automation of nanohandling procedures. It incorporates simultaneous tracking of an object in a stereo image pair, generated using a self-built magnetic lens for tilting the electron beam. Object depth is concluded from the object displacement. The tracking procedure is based on the active contours method, adjusted by a region based energy function. An automatic contour initialization is presented that makes use of edge detection. All methods described have been implemented and tested extensively in the designated automation environment. The approach is found to provide very accurate depth estimates even at a small level of magnification.

1 INTRODUCTION

The Scanning Electron Microscope (SEM) is widely used as imaging modality for the handling of micro- and nanoscale objects. Its special properties like fast scanning speed and flexible magnification within a great magnification range make it a very powerful tool for this application. While teleoperated manipulation of nanoscale objects has been shown already in different scenarios, automated setups requiring minimal user interactions to achieve a nanomanipulation goal are still of current research interest. Problematic for the processes is the lack of information concerning tools and workpieces. Though in most cases axes and actuators may deliver information about their position, the real position of the end-effectors is not known exactly. The factors which contribute to this are e.g. thermal drift and play, which may have effects in the order of magnitude of the nanoscale manipulation setups. These problems have been solved or diminished by applying image processing algorithms, specifically object tracking algorithms like in (Kratochvil et al., 2007) or (Sievers and Fatikow, 2006), to obtain a position information at least in the image plane. A problem which still persists is the missing depth information of the imaged objects. The SEM is only able to deliver two-dimensional images which even exhibit a great depth of field, so that even for teleoperated manipulation, a certain experience is needed.

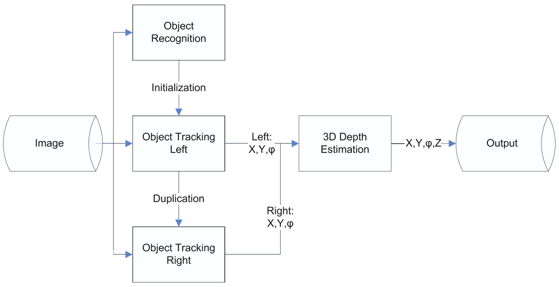


Figure 1: Overview of the proposed depth estimation algorithm.

In order to obtain depth information, different approaches are used throughout the literature. Example methods are depth from focus, depth from motion and depth from stereo. In this paper we base our work on the depth from stereo approach. Specifically we present an approach for depth recovery using tracking data from a stereo image pair.

One problem in depth from stereo approaches is the solution of the correspondence problem. Different methods exist to determine correspondencies, e.g. pixel-based methods. Due to the slow calculation of pixel-wise correspondences and the high amount of ambiguity inherent in the solution, feature based methods promise better performance. Due to the fact that in most cases the objects of interest have to be tracked anyways to determine their position in the image plane, it is reasonable to make use of this fact also for depth estimation. Tracking the object in parallel in a stereo image pair means that the correspondency problem is principally solved for the object.

The overall algorithm design can be seen in figure 1. In order to enable an automatic extraction of the z-position and the position in the image plane, we developed a recognition algorithm able to detect the objects used in our experiments and deliver the necessary initialization data for the tracking algorithm. The tracking algorithm is split into separate algorithms for each stereo pair image. The tracker for the left image is in the first place minimized freely to segment the object from the background. The contour data is then duplicated to the right image tracker, which uses just restricted transformations to determine the position of the object in the right image. Both position informations are fed to a depth estimation algorithm, which analyzes the displacement of the two tracked positions and determines the distance of the object.

2 ALGORITHM INITIALIZATION

A common drawback of active contour based approaches is the necessity for manual algorithm initialization. Usually a graphical user interface (GUI) is used to present a dialog where the user is asked to draw an initial estimate of the object's shape. Instead we choose another approach which allows automatic initialization of the active contour. The proposed method is based on the very popular edge detection method of Canny (Gonzalez and Woods, 2008) and an initial estimate of the object size. In the following we will give a brief overview of Canny edge detection and then show how it can be applied to our problem.

To reduce the effect of high frequency noise usually a Gaussian low-pass filter is applied to the input image before edge detection. The main advantages of the Gaussian low-pass are the efficient computability and the good stopband attenuation (Forsyth and Ponce, 2002). The convolution kernel is:

$$G_\sigma(x,y) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{(x^2+y^2)}{2\sigma^2}\right) \tag{1}$$

The actual contour detection algorithm consists of the following steps:

1. *Determination of edge points* - A measure of image gradient G in the direction of the image coordinate axes x,y is needed. The Sobel- and Prewitt operator (Jaehne, 1991) are two examples of first-order derivative approximation filters. From the filter responses G_x, G_y the local gradient magnitude $g(x,y)$ and orientation $\alpha(x,y)$ are computed:

   $$g(x,y) = \sqrt{G_x^2 + G_y^2} \tag{2}$$

   $$\alpha(x,y) = \tan^{-1}(G_y/G_x) \tag{3}$$

   Edge points are obtained by finding the local maxima of $g(x,y)$ in the direction of $\alpha(x,y)$.

2. *Nonmaximal suppression* - The prior step produces ridges along image edges that are not yet necessarily a thin line. Nonmaximal suppression removes pixels that are not on the top of the ridge and leaves a thin line.

3. *Hysteresis thresholding* - Two threshold values T_1 and T_2, with $\mathrm{T}_1 < \mathrm{T}_2$ are used to remove edges of low strength. Initially, all edge pixels where $g(x,y) < \mathrm{T}_1$ are removed. The next step removes all remaining pixels with $\mathrm{T}_1 < g(x,y) < \mathrm{T}_2$ that are not 8-connected to an edge segment with $g(x,y) < \mathrm{T}_2$. This procedure is depicted in figure 2.

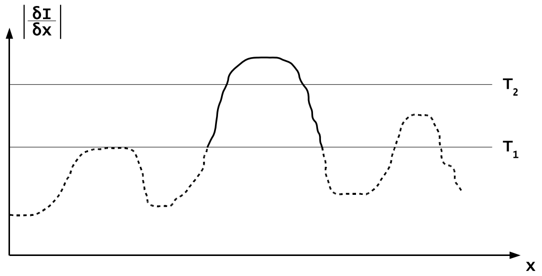


Figure 2: Hysteresis thresholding as used in Canny edge detection, illustrated by a one-dimensional signal I. Only connected line segments with derivative values higher than T_1 and at least one segment higher than T_2 survive this step.

The methods as described above have been applied to a sample SEM image taken from the automation sequence (figure 3). The Sobel operator has been chosen as edge emphasizing filter. It can be seen from the result of the Canny edge detection that is not only sensitive to the target object but also to background structure. For retrieving the actual object shape that is used to initialize the active contour we incorporate three further steps:

4. *Binary dilation* - Lines in the binary edge image are thickened to close gaps in the outer object hull. For this operation a *structuring element* is needed which is another binary image. A binary approximation of a circular disk is a common choice.

5. *Contour retrieval* - The dilated image is inspected for contours which are point sequences that enclose 8-connected binary objects. Only outer contours which are not wrapped by a larger contour are taken into account.

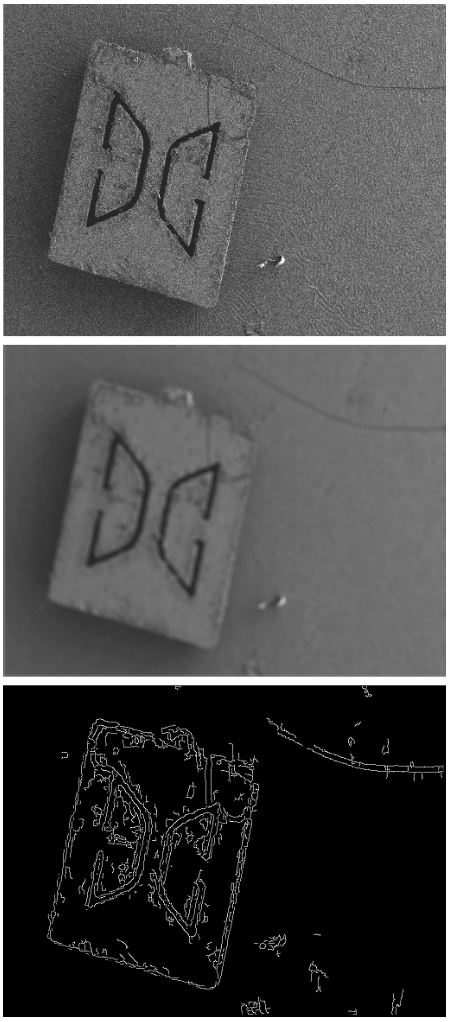


Figure 3: Processing steps: the initial image (top) after Gaussian filtering (middle) and Canny edge detection (bottom).

6. *Size thresholding* - From the outer object contours the enclosed area and thus the object mass can be derived directly. In this particular problem setting the object in demand is the largest object in the image scene. Also the physical dimensions are well-known. All but the largest object are dropped.

These processing steps have been applied to the output of the edge detection and the results can be seen in figure 4. The dilated image (top) points out and connects the image edges. Outer contours are detected and the enclosed regions are filled (middle image, overlaid with the original image). Finally small objects are removed from the image scene (bottom). The object boundary pixels are output of the initialization procedure and input to the active contour algorithm.

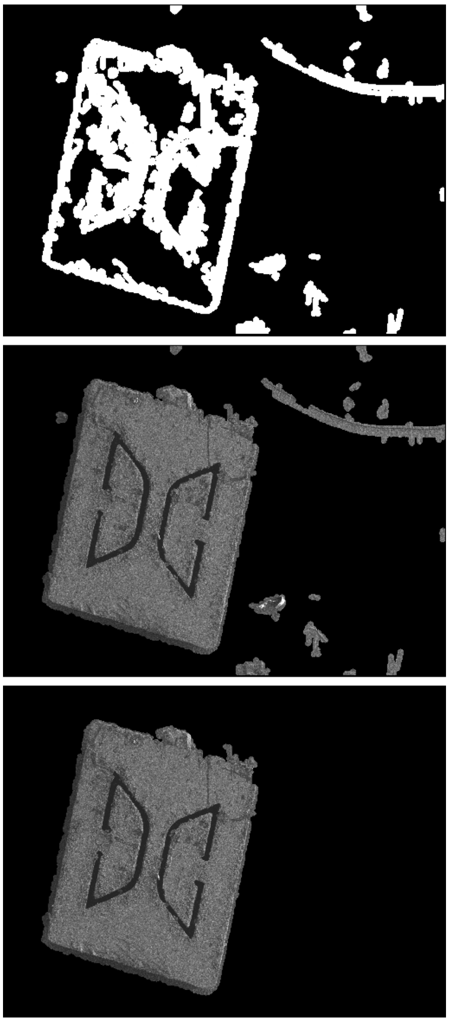


Figure 4: Binary dilation with a circular structuring element is applied (top). After region growing (middle), objects which do not meet the expected size are removed (bottom).

3 OBJECT TRACKING

In order for generating input data for the depth estimation, a tracking algorithm is used to track the position of the object not only in one image, but separately in the two stereo images. The tracking algorithm is based on active contours (for details on this concept see (Kass et al., 1988) and (Blake and Isard, 2000)). Active contours are curves in the image plane which segment the image into two regions, one enclosed by the contour and one surrounding the contour. The enclosed segment is the tracked object, the surrounding segment the rest of the scene. For determining the segmentation, active contours exhibit an energy function. This energy function is dependent on certain properties of the contour, like length, compactness etc., and on properties of the image data, like edges.

Though the original formulation of the active contours like described in (Kass et al., 1988) depends on edges for the energy function, it turned out that an energy function derived from image region statistics is more robust for SEM images (for details see (Sievers, 2006) and (Sievers, 2007)). Especially this is the case with very noisy images. The energy function used here is defined as

$$E(C) = E_{int}(C) + E_{ext}(C) \tag{4}$$

with

$$E_{int}(C) = \frac{L^3}{A} \tag{5}$$

with L the length of the contour, A the area enclosed by the contour and

$$E_{ext}(C) = N_a f\left(\frac{1}{N_a}\sum_{x \in a} x\right) + N_b f\left(\frac{1}{N_b}\sum_{x \in b} x\right) \tag{6}$$

with

$$f(z) = -zln(z) \tag{7}$$

with a and b being the two regions in which the contour segments the image.

The original algorithm has been used already for the tracking of objects in SEM images. Important in comparison to microscope is that the algorithm has to be as robust as possible against certain effects which may occur in SEM images, especially in SEM nanomanipulations setups. It has shown to be robust against:

- *noise - due to fast SEM image acquisition during manipulation*
- *partial occlusion - due to complex manipulation setups*
- *grey level fluctuations - due to charge accumulation and discharging effects in the SEM chamber*

The implementation is fast enough to track objects in SEM images with relatively high framerates. For the initial minimization, the contour points are handled individually, resulting in a deforming contour which fits itself to the object. After the initial minimization of the contour, the transformation for the subsequent steps is restricted to euclidean transforms. This enables the left image tracker to transfer the contour points to the right image tracker, and is also the reason for the robustness against occlusion. The transfer and duplication of the contour leads to the same object being tracked in both stereo images.

The steps executed in the minimization of the whole contour are the following, executed in order:

1. *Minimization by translation in X*
2. *Minimization by translation in Y*
3. *Minimization by rotation*

In the original algorithm (described in (Sievers, 2007)), additional minimization by scaling was executed, which is not used here due to coherency issues. When adding the scaling minimization, the tracking gets slightly more unstable, and a independent minimization on the two separate images may lead to incoherent states of the tracker, e.g. different scaling factors for the two images.

The modifications made to the algorithm mainly deal with the duplication of the tracker for stereo tracking. The left image tracker is extended with a component duplicating the contour like shown in figure 5, the right image tracker does not need an own free minimization of the contour, but is initialized by the left image tracker with the contour data as shown in figure 6.

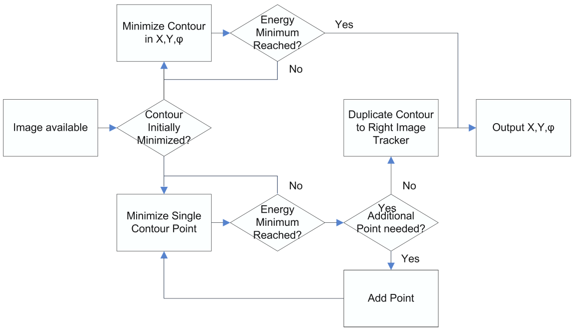

Minimize Contour in X,Y,φ
Energy Minimum Reached?
Yes
No
Image available
Contour Initially Minimized?
Duplicate Contour to Right Image Tracker
Output X,Y,φ
No
No
Minimize Single Contour Point
Energy Minimum Reached?
Yes
Additional Point needed?
Yes
Add Point

Figure 5: The tracking algorithm for the left image.

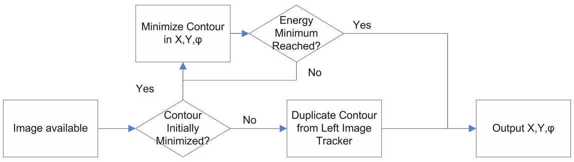

Minimize Contour in X,Y,φ
Energy Minimum Reached?
Yes
No
Yes
Image available
Contour Initially Minimized?
No
Duplicate Contour from Left Image Tracker
Output X,Y,φ

Figure 6: The tracking algorithm for the right image.

The active contour tracker itself is robust enough to track the objects in the experiment, though some parameter tuning was necessary due to the inhomogeneous appearance of the objects.

4 DEPTH ESTIMATION

Many modern applications use the well known advantages of SEMs. These applications often require knowledge of the exact position of involved objects or topographic information of a specimen. Therefore, measurements for all 3 dimensions are required. This requirement exceeds the possibilities of standard

SEMs, because they are only capable of delivering 2-dimensional images of a scene. Typical applications are grabbing and assembling processes such as the manipulation of carbon nanotubes (CNT).

There are different approaches to obtain 3-dimensional measurements in scanning electron microscopy. Beside tactile sensors and laser-based measuring methods, focus-based methods are popular in image processing. Due to problems with the standard methods over the last years stereoscopy-based methods in SEM applications gained in importance. The advantage of stereoscopy techniques is the ability to display dynamic processes in real time in contrast to focus-based methods which need to acquire a series of images for depth calculation.

The application of the stereoscopy principle in the SEM is similar to the application in the macro world. Different z-positions lead to different displacements between the two stereo images. This displacement leads to z-position dependent disparities of the corresponding points. The disparity is used to calculate the original z-position. The angular displacement between the two views is also called vergence angle (figure 7).

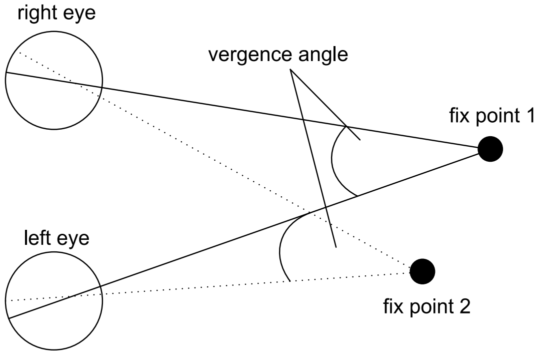


Figure 7: The principle of stereoscopy using the example of human vision.

The stereoscopy approach consists of two main parts: the stereo image acquisition and the stereo analysis. Different methods to acquire stereo images exist. Research currently follows the three following approaches:

- Dual-beam scanning electron microscope
- Specimen table tilting
- SEM beam deflection

The first two approaches have some disadvantages. Dual-beam scanning microscopes are very expensive and far away from being the standard in most SEM environments. A standard specimen table is already able to perform the required shift and tilt movements for the second method, but the experimental setup of many applications is not compatible with the tilting of the specimen table, e.g. setups for manipulation processes with mobile robots. These are core reasons why our group follows the approach of electron beam deflection (see also in (Jahnisch and Fatikow, 2007)). The beam is deflected by two units:

- Internal SEM beam shift unit
- External self-built-magnetic lens

The advantage of this setup is the easy integration of the system into standard SEMs.

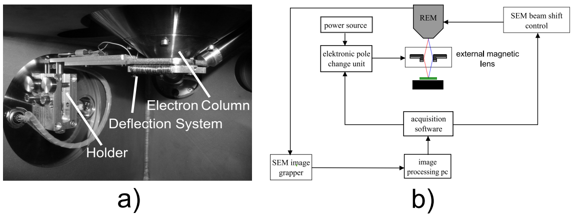


Figure 8: a) An image of the integrated external magnetic lens in the SEM vacuum chamber. b) A sketch of the principle of electron beam deflection and the control system of the stereo image acquisition in the SEM.

To acquire stereo images, special software was written to synchronize the SEM beam shift control and the external magnetic lens. Figure 8 shows the principle and control flow of the stereo image acquisition. The disadvantage and main difference of stereoscopy in the SEM to macro world stereoscopy is the small vergence angle due to technical limits imposed by the construction of the acquisition system. This is one of the reasons why there are special requirements to the stereo image analysis in the SEM. The main challenge of the analysis is to solve the correspondence problem. The solution is derived by finding the corresponding points or objects in the stereo image pair.

Currently there are just a few analysis methods to solve correspondence in the scanning electron microscopy. The algorithm used in our vision feedback group (see (Jahnisch and Fatikow, 2007)) is based on the human visual cortex and is able to calculate disparities without any previous knowledge of the specimen. This procedure is a pixel-based correspondence solving algorithm. On the one hand this method is very versatile, but on the other hand the algorithm needs almost noise free stereo input images and a very long computation time to create reliable disparities. Current research is looking for new methods to calculate the correspondence in SEM stereo images with the following features:

- Fast calculation for real-time applications
- High noise robustness

- High reliability and accuracy for automation applications

Feature-based correspondence analysis is interesting due to its flexibility. Therefore, common methods for image tracking, recognition and classification are reviewed with respect to their capability of calculating correspondence under SEM-specific conditions. Due to the experience with the application of active contours (see section 3) in SEM applications, their high noise robustness and the fast calculation time after an initialization step, they are a promising candidate for correspondence analysis. The principle is to initialize the active contour based tracking algorithm on one of the two stereo images and use the derived contour to initialize the tracking algorithm on the other stereo image. Due to the small vergence angle and the small acquisition time difference of a stereo image pair the correspondence between objects can be solved.

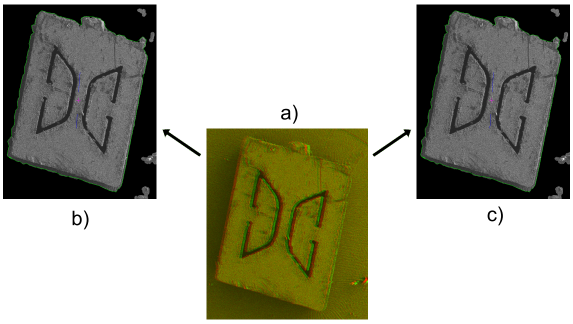


Figure 9: a) Shows the red green image of the stereo image pair. b) Preprocessed left image with active contour. c) Preprocessed right image with active contour initialized by the right image contour.

In the following step, the center points and the orientations of both contours are used to calculate the disparity and to calculate the angle difference between the contours. This difference reflects the z position and orientation of the specimen. To validate this new procedure, a series of stereo images with different z-positions of a specimen in the SEM were taken. The following two diagrams show two series of measurement with different SEM scan speeds.

Figure 10 shows a series of small z-differences with their corresponding disparities. Due to the small magnification of 60x the resolution of the stereo images is insufficient to detect disparities to z-displacements of a few μm. That is the reason for the noise with small displacements. Thanks to the features of SEMs and the scalability of the active contours algorithm, the disparity calculation of small z-displacement is still possible in high magnifications.

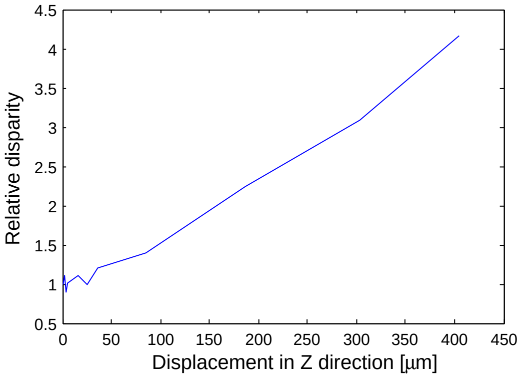


Figure 10: Graph shows small z-displacements with corresponding disparities.

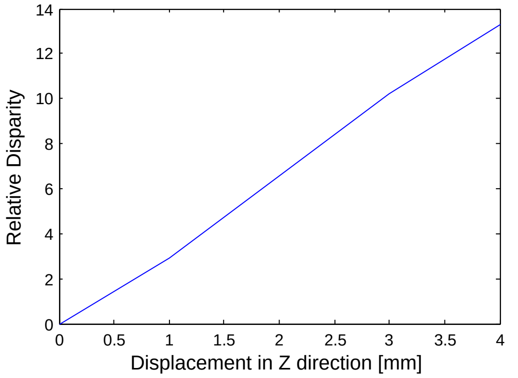


Figure 11: Graph shows z-displacements in the range of 0,5 to 4 mm with corresponding disparities.

This new feature-based correspondence problem is just limited by the resolution and the vergence angle. Recently, the algorithm was tested with CNTs using a magnification of 1000x with the ability to detect z displacement in the single-digit μm domain. In the future, it is planned to build a new SEM scan generator to increase the stereo image resolution and to build new lens systems to get a higher vergence angle.

5 SUMMARY

In this paper, we have shown a new approach for depth estimation using SEM images. The use of a duplicated two-dimensional tracking algorithm generates data which can be used to determine depth. The algorithm is initialized and then duplicates the contour information to process both images from a stereo image pair. The displacements detected are a measure of depth. The results of this algorithm are very promising and further effort will be made to improve the approach. Even with the results obtained already, the featured algorithm shows to be a promising tool for the automation of nanohandling processes. With magnifications as low as 60x which has been used in this experiment, depth differences in the order of 50*μm* can be detected.

REFERENCES

Blake, A. and Isard, M. (2000). *Active Contours*. Springer.

Forsyth, D. and Ponce, J. (2002). *Computer Vision: A Modern Approach*. Prentice Hall.

Gonzalez and Woods (2008). *Digital Image Processing*. Prentice Hall.

Jaehne, B. (1991). *Digital Image Processing*. Springer.

Jahnisch, M. and Fatikow, S. (2007). 3-D vision feedback for nanohandling monitoring in a scanning electron microscope. *International Journal of Optomechatronics*, 1(1):4–26.

Kass, M., Witkin, A., and Terzopoulos, D. (1988). Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331.

Kratochvil, B. E., Dong, L. X., and Nelson, B. J. (2007). Real-time rigid-body visual tracking in a scanning electron microscope. In *Proc. of the 7th IEEE Conf. on Nanotechnology (IEEE-NANO2007), Hong Kong, China*.

Sievers, T. (2006). Global sensor feedback for automatic nanohandling inside a scanning electron microscope. In *Proc. of 2nd I*PROMS NoE Virtual International Conference on Intelligent Production Machines and Systems*, pages 289–294.

Sievers, T. (2007). *Echtzeit-Objektverfolgung im Rasterelektronenmikroskop*. PhD thesis, University of Oldenburg.

Sievers, T. and Fatikow, S. (2006). Real-time object tracking for the robot-based nanohandling in a scanning electron microscope. *Journal of Micromechatronics*, 3(3-4):267–284.

REAL TIME OBJECT DELIMITERS EXTRACTION FOR ENVIRONMENT REPRESENTATION IN DRIVING SCENARIOS

Andrei Vatavu, Sergiu Nedevschi and Florin Oniga
Computer Science Department, Technical University of Cluj-Napoca, 26-28 G. Baritiu Street, Cluj-Napoca, Romania
{Andrei.Vatavu, Sergiu.Nedevschi, Florin.Oniga}@cs.utcluj.ro

Keywords: Contour Tracing, Environment Representation, Border Scanning, Environment Perception, Polyline Extraction, Object Delimiters.

Abstract: The environment representation is essential for driving assistance. However, the performances achieved in complex environments are still unsatisfactory regarding the accuracy, confidence and real time capabilities. This paper presents a real-time 2.5D environment representation model, for driving scenarios, based on object delimiters extraction from a 3D occupancy grid obtained from dense stereo. We propose two approaches to extract the polyline delimiters: an improved contour tracing called 3A Tracing and a polyline extraction method through the occupancy grid radial scanning. The advantages and drawbacks for each of these methods have been discussed.

1 INTRODUCTION

In the context of in-vehicle navigation systems, the environment perception and its convenient representation is an important requirement (Pijpers, 2007). The process of environment representation building has to be accurate and characterized by a low computational cost.

Usually, the Driving Assistance Applications detect the objects through 2D or 3D points grouping processes. The detected objects are represented by geometric primitives such as 2D bounding boxes (Dellaert, 1997) or 3D cuboids (Nedevschi, 2007). As an alternative approach, the objects may be represented by polylines. One of the advantages of the polyline based objects representation is the close approximation of the object contour by the polygonal model while having a number of vertices as small as possible. In the same time the polyline could inherit the type, position and height properties of the associated object.

The polyline object representation may lead to the creation of subsequent algorithms that are computationally fast due to the fact that only a small subset of points is employed.

The road feature identification through the object delimiters detection can be used in the unstructured environments as an alternative solution to the lane detection algorithms.

The object delimiters extraction is studied in some areas like mobile robots (Harati, 2007; Magin, 1994; Joshi, 2002; Laviers, 2004; Veeck, 2004), or autonomous vehicle systems (Kolski, 2006; Madhavan, 2002; Goncalves, 2007). The polyline representation is very common in many algorithms, such as localization and mapping (Joshi, 2002; Laviers, 2004; Veeck, 2004; Madhavan, 2002), contour tracking (Prakash, 2007) and path planning (Madhavan, 2002).

The polyline extraction methods differ by the nature of the information as well as by the sensors used for data acquisition process. Current systems use laser (Harati, 2007; Veeck, 2004; Kolski, 2006; Madhavan, 2002), sonar (Goncalves, 2007; Laviers, 2004) or vision sensors (Goncalves, 2007).

Two main directions can be distinguished for the delimiters extraction:

- **The contour processing of already detected objects from the scene** (Gonzales, 2002);
- **The radial scanning of the environment**. This method is common for the systems based on sonar or laser sensors (Harati, 2007; Kolski, 2006).

A method for map representation as a set of line segments or polylines is described in (Laviers, 2004). An occupancy grid is created here from sonar information. The data is converted to a list of vertices using the Douglas Peucker line reduction algorithm.

In (Veeck, 2004) a method that learns sets of polylines from laser range information is presented. The polylines are iteratively optimized using the Bayesian Information Criterion.

The polyline representation was chosen in (Madhavan, 2002) for terrain-aided localization of autonomous vehicle. The new range data obtained from the sensor are integrated into the polyline map by attaching line segments to the end of the polyline as the vehicle moves gradually along the tunnel.

In this paper we present and evaluate several methods for real-time environment representation by extracting object delimiters from the traffic scenes using a Dense Stereovision System (Nedevschi, 2007). The delimiters detection is based on processing the information provided by a 3D classified occupancy grid obtained from the raw dense stereo information. One of the problems in representing the environment through the occupancy grid is a large volume of data. Therefore we propose a more compact 2.5D model by representing the environment as a set of polylines with associated height features. We present two approaches to extract object delimiters:

- **The 3A Tracing.** The classical algorithm for contour tracing is improved by developing a new method named 3A Tracing Algorithm;
- **The radial scanning of the occupancy grid.** We have developed a Border Scanning method that is able to detect delimiters of complex objects taking into account the nature of information from the traffic scene (curb, object, and road).

A polyline map is generated as the result of the delimiters extraction process. Each polyline element inherits the type (object, curb), position and height properties of the associated objects from the occupancy grid.

In the next section, we describe the proposed Delimiters Extraction architecture. The delimiters detection approaches are presented in section 3. Experimental results are given in section 4, and section 5 concludes the paper with final remarks.

2 PROPOSED ARCHITECTURE

Our delimiters detection approaches have been conceived for an urban driving assistance system. We extended our Dense Stereo-Based Object Recognition System (DESBOR) by developing an Object Delimiters Detection component. A detailed description about the DESBOR system is presented in (Nedevschi, 2007).

The Object Delimiters Detection system architecture consists in the following modules (see Figure 1):

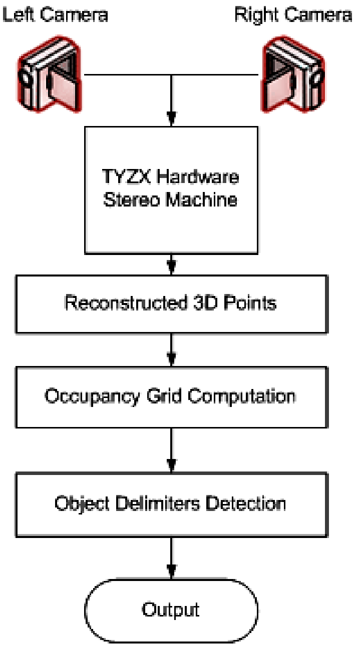

Left Camera
Right Camera
TYZX Hardware Stereo Machine
Reconstructed 3D Points
Occupancy Grid Computation
Object Delimiters Detection
Output

Figure 1: System Architecture.

TYZX Hardware Stereo Machine. The 3D reconstruction is performed by the "TYZX" hardware board (Woodill, 2004).

Reconstructed 3D Points. The reconstructed 3D points are used for the occupancy grid generation.

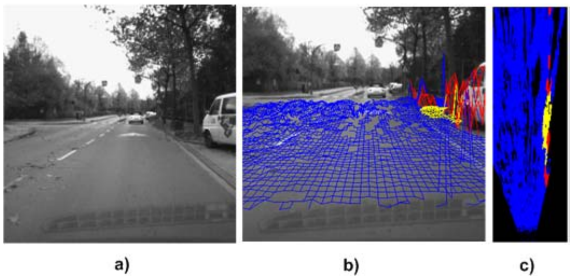


Figure 2: The Occupancy Grid (c) is computed from the Elevation Map (b) of a scene (a). The occupancy grid cells are roughly classified (blue – road, yellow – traffic isle, red – obstacles).

Occupancy Grid Computation. The occupancy grid (see Figure 2.c) represents a description of the scene, computed from the raw dense stereo information represented as a digital elevation map (see Figure 2.b). The occupancy grid cells are classified into road, traffic isle and object cells. A detailed description about the occupancy grid computation is presented in (Oniga, 2007).

Object Delimiters Detection. The Object Delimiters detection uses the occupancy grid results as the input and generates a set of unstructured polygons approximated with the objects contour. The delimiters can be extracted from the occupancy grid through both 3A Tracing and Border Scanning algorithms.

Object Delimiters Detection Output. A polyline map is generated as the result of delimiters extraction process. For each polyline element we keep the following information: a list of vertices, the delimiter type (object, curb), and the height of the object for which we apply the polyline extraction.

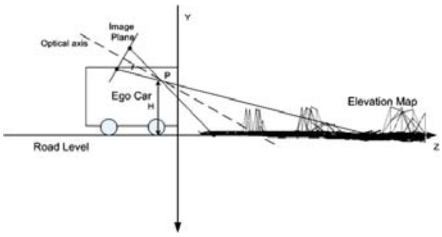


Figure 3: The car coordinate system.

It must be noted that the car coordinate system coincide with the world coordinate system having its origin on the ground in front of the car (see Figure 3). The position and orientation of the stereo cameras are determined by the absolute extrinsic parameters (Marita, 2006).

3 OBJECT DELIMITERS EXTRACTION APPROACHES

A set of steps have been identified for the delimiters extraction:

Step 1: Object Labeling. In this step each object from the occupancy grid is labeled with a unique identifier.

Step 2: The Contour Extraction. We compute the contours of the non-drivable blobs (objects, traffic isles) from the occupancy grid. Each contour point will represent a single cell in the grid map.

Step 3: The Polygonal Approximation. Given a curve C we will find a polygon that closely approximates C while having as small a number of vertices as possible.

Next, we will present several algorithms developed by us for delimiters extraction. All these methods have in common the 1st and 3rd step. The 2nd step is different in each case. We have used two main approaches for the contour extraction:

1) The Contour Tracing for a Given Object. Once an object cell has been identified, contour tracing is performed starting from this point, adding each traversed cell to the current contour. In this paper we present an improved version of contour tracing, the 3A Tracing Algorithm.

2) The Border Scanning. A radial scanning is performed with a given radial step, traversing the interest zone and accumulating the contour points at the same time. The main difference of this approach is that we scan only the visible parts from the ego-car position. Two main improvements of the Border Scanning method are discussed: the Border Scanning using a variable step, and the Combined Border Scanning, taking into account the occupancy grid blob's nature (traffic isles, obstacles).

3.1 The 3A Tracing Algorithm

The classical contour tracing algorithm collects the contour points of an object by traversing the object boundary.

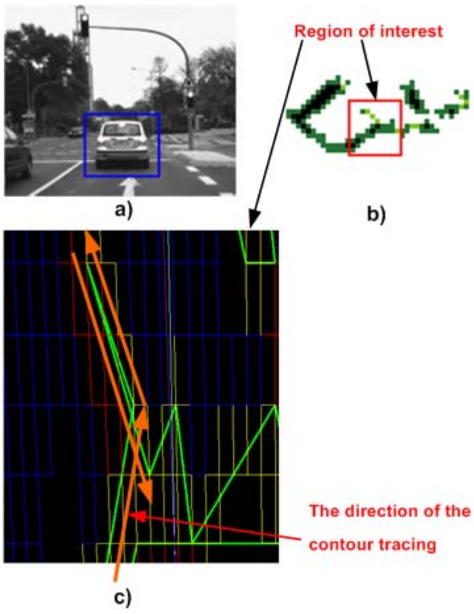


Figure 4: Contour tracing of the care points (b) from the scene (a). There are cases when two polygonal segments can intersect each other (c), after the polygonal approximation of the car contour.

A disadvantage of the classical algorithm is that there are cases when the same delimiter point can be passed many times. This may lead to the incorrect representation, after the contour approximation step (see Figure 4).

To avoid this problem we have developed an extended contour tracing algorithm named 3A Tracing. In this method we use two stacks, Stack A and Stack B. The name 3A Tracing comes from the next three main phases (see Figure 5):

Phase 1: Accumulation. The tracing is made analogue to the Contour Tracing algorithm. All accumulated points are pushed onto the stack A. The traversed points are marked with a flag in order to know whether they were traversed or not at least one time. Once we found a terminal point (from which the tracing is made in the inverse sense) we pass to the 2nd phase of the algorithm.

Phase 2: Adjustment. In this phase the tracing continues in the inverse sense by extracting already passed points (drawn with light green) from the Stack A, and pushing them onto the Stack B. The Adjustment is repeated until we reach a contour point that has not passed yet. Once the new contour point is found we pass to the 3rd phase of the algorithm.

Phase 3: Approximation. Polygonal approximation is applied to each of the two stacks. After the polygonal approximation process the two stacks will be cleared and the algorithm is repeated from the Phase 1.

The algorithm stops when the start point is reached once again.

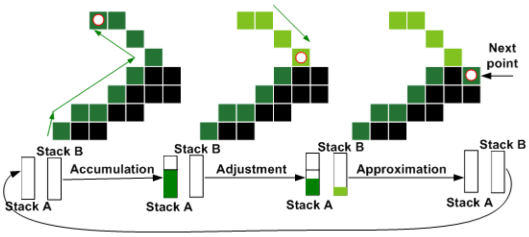


Figure 5: 3A Tracing Algorithm. In the Accumulation phase, all traversed points are pushed onto the Stack A. In the Adjustment stage, the already passed points are extracted form the Stack A and pushed onto the Stack B. Polygonal approximation is applied in the last step of algorithm.

Although the 3A Tracing algorithm eliminates some particular cases in which two polygonal segments may intersect, like in the Contour Tracing, it works only on the connected components. Therefore this method does not take into account the cases of more complex objects, when a single obstacle is represented as many disjoint patches.Therefore we have elaborated an extraction method through the radial scanning of the Elevation Map.

3.2 The Border Scanner Algorithm

The Border Scanner algorithm performs a radial scanning with a given radial step. The scanning axis moves in the radial direction, having a fixed center at the Ego Car position. The scanning process is made into the limits of *Q_from* and *Q_to* angles, thus only the interest area are scanned, where the delimiters can be detected (see Figure 6). Having a radial axis with a *Qrad* slope, *Q_from* < *Qrad* < *Q_to*, we try to find the nearest point from the Ego Car situated on this axis. In this way, all subsequent points will be accumulated into a *Contour List*, moving the scanning axis in the radial direction. At each radial step we verify that a new object has been reached. If a new label has been found then the polygonal approximation on the *Contour List* points is performed. The list will be cleared, and the algorithm will be continued finding a new polygon.

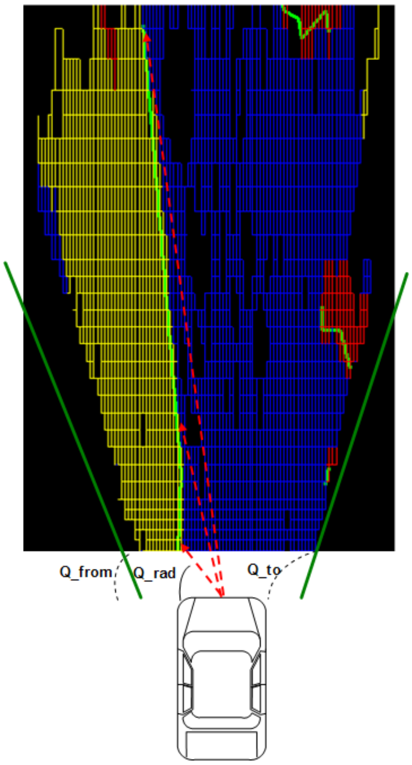


Figure 6: Border Scanning on the Occupancy Grid Points.

Advantages. The obtained results are more close to the real obstacle delimiters from the scene. The problem of the complex objects presented in the case of Contour Tracing algorithms is eliminated. Therefore many disjoint patches that belong to the same object can be enveloped by a single delimiter.

Disadvantages. A little obstacle (noise present in the occupancy grid) can occlude a great part from the scene, if this obstacle is too near to the Ego-Car. The scanning is influenced by the presence of such false obstacles.

3.2.1 The Border Scanning Algorithm using Variable Step

Having a constant radial step, the detected pixel density will decrease with the depth distance. The distance between two consecutive detected pixels is greater at the far depths. The idea is that some important information about the delimiters can be lost at the far distances.

A good solution is to use a scanning method with a variable step, thus the radial step should be adapted with the distance.

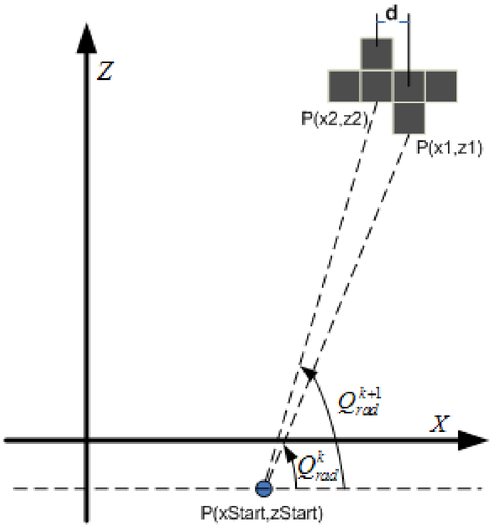


Figure 7: Radial angle estimation for the next step in the Variable Step Border Scanning approach.

If we have a point $P_1(x_1, z_1)$ of a given object and a radial axis containing the point P_1 with a radial angle Q^k_{rad} at the k step, then we estimate the radial angle at the $k+1$ step (see Figure 7):

$$Q^{k+1}_{rad} = \arctan(\frac{z_1 - z_{Start}}{x_1 - x_{Start} - d}) \qquad (1)$$

Where:

- x_1, z_1 are the coordinates of the P_1 point;;
- x_{Start}, z_{Start} are the ego-car point coordinates.
- d is considered the distance between any two adjacent points.

However, there are situations when no object point can be reached on the current scanning axis. Therefore we cannot estimate the radial angle for the next step, because we don't know the distance of the current object point from the Ego-Car. In this case, like in the simple Border Scanning method, we use a fixed step, until a new object point will be found.

3.2.2 The Combined Border Scanning

We know that the occupancy grid cells are classified into obstacles (cars, pedestrians etc.) or traffic isles (road-parallel patches). If we take into account only the first nearest point from the car, many relevant objects delimiters may be omitted. For example, the first obstacle from the car can be a curb. In this case, we are interested not only in the curb delimiters but also in the delimiters above the curb or behind the curb. Therefore we extended our Border Scanning algorithm by developing a method that takes into consideration the obstacle's nature making a decision based on two types of information "What have we found?" and "What we have to find?". The algorithm consists in two passes: one for the object delimiters detection, and second for the traffic isles delimiters detection.

In the Table 1 is presented the returned result when we want to find a delimiter taking in account the point type we have found.

Table 1: The Combined Border Scanning method. The result is returned, taking into consideration the found point type.

Delimiter's type we want to find	Point Type we have found	Returned result
OBJECT	OBJECT	FOUND
CURB	OBJECT	NOT FOUND
CURB	CURB	FOUND

4 EXPERIMENTAL RESULTS

For the experimental results we have tested a set of 15 scenarios from the urban traffic environment using a 2.66GHz Intel Core 2 Duo Computer with 2GB of RAM.

Figure 8 shows a comparative result between the Contour Tracing and 3A Tracing algorithms, using an approximation error of two points. One can notice that the polygonal segment intersection in the case of classical contour tracing algorithm (see Figure 8.b) was eliminated by applying the 3A Tracing algorithm (see Figure 8.c).

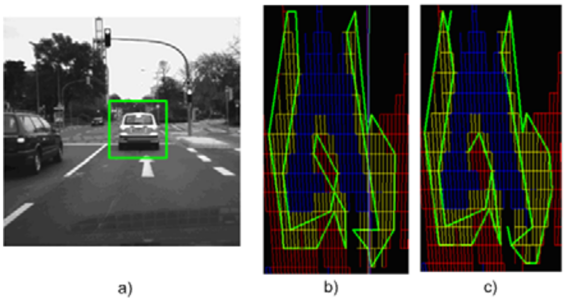


Figure 8: Delimiters detection through the Contour Tracing algorithm (b) and 3A Tracing algorithm (c). The detection is performed on the occupancy grid computed from the scene (a).

The difference between the result of delimiters detection in the case of Simple border Scanner and Combined Border Scanner algorithms is presented in the Figure 9. It can be observed that in the case of Combined Border Scanner (see Figure 9.b) the side fence’s delimiter is detected in spite of his position behind the curb (Figure 9.c).

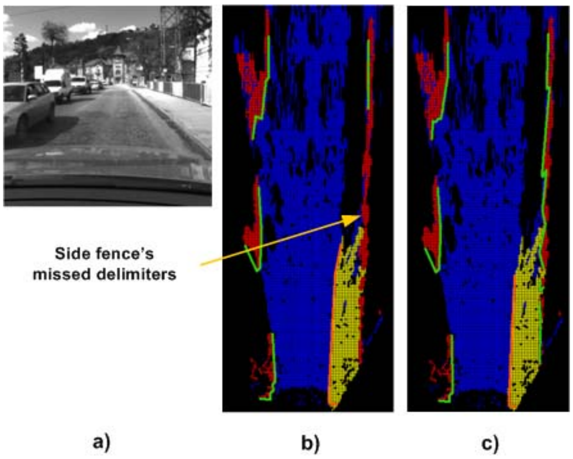


Figure 9: Border scanning of a scene (a). The side fence’s delimiters are missed in the case of the Simple Border Scanning (b) and have been detected in the case of the Combined Border Scanning algorithm (c).

In the Table 2 the results from the Variable Step Border Scanner and Fixed Step Border Scanner are computed for the same driving scene. It can be observed that the number of detected points is greater in the case of Variable Step Border Scanner algorithm, thereby 11466 points, which means 28 detected points per frame in comparison with 22 detected points per frame in the case of Fixed Step Border Scanner algorithm.

Table 2: Fixed Step Border Scanner vs. Variable Step Border Scanner.

	Fixed Step Border Scanner	Variable Step Border Scanner
Number of Frames	406	406
Detected points	9058	11466
The radial step (radians)	0.01	variable
Points per Frames	22	28
Average processing time per frame	4 ms	5 ms

The average extraction time using the 3A Tracing algorithm is about 0.7ms per frame and depends on the angular resolution in the case of Border Scanner approach.

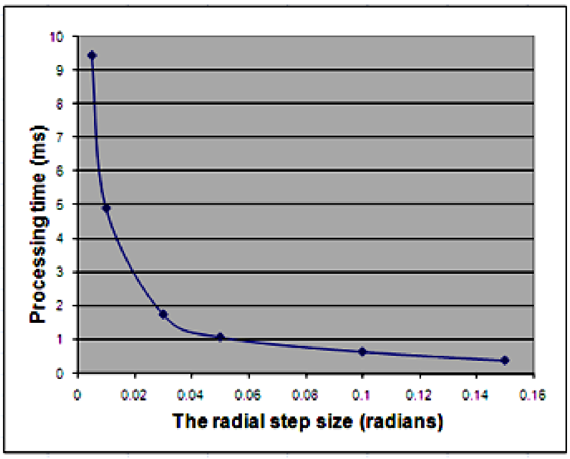


Figure 10: The processing time vs. the radial step size.

Figure 10 shows how the radial step size variation affects the system response time using the Combined Border Scanning approach.

Figure 11 is a diagram that shows the impact of radial step size on delimiters detection rate using the border scanner method. We can observe that, with a higher radial step size we obtain an increase in processing time while the detection rate decreases. The solution is a tradeoff between the system processing time and detection rate.

Figure 12 presents results for various traffic scenes using the Combined Border Scanning method. For the border scanning algorithm with a radial step of 0.01 radians the average processing time is about 5ms and the delimiters detection rate is 98.66%.

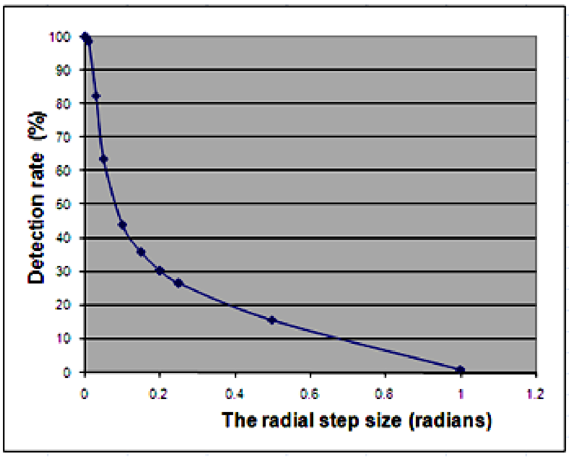


Figure 11: The detection rate vs. the radial step size.

5 CONCLUSIONS

In this paper we present and evaluate several methods for real-time environment representation through the object delimiter extraction and characterization from dense stereovision images. The delimiters detection is based on processing the information provided by a 3D classified occupancy grid obtained from the raw dense stereo information. The result is a more compact 2.5D model for representing the environment, as a set of polylines. Each polyline element inherits the type (object, curb), position and height properties of the associated object from the occupancy grid.

We have developed an improved Contour Tracing method named 3A Tracing algorithm that eliminates the situation when two polygonal segments can intersect each other.

Another approach presented in this paper is the polyline extraction through the radial scanning of the occupancy grid. Although the tracing approach is more computationally-efficient, the results provided by the Border Scanner algorithm are more appropriate for detecting the real obstacle delimiters from the scene. The algorithm is able to extract only the visible area from the ego-vehicle since the occluded points do not offer relevant information. Using the Border Scanner algorithm, our system is fast and achieves a high rate of detection: 98.66%.

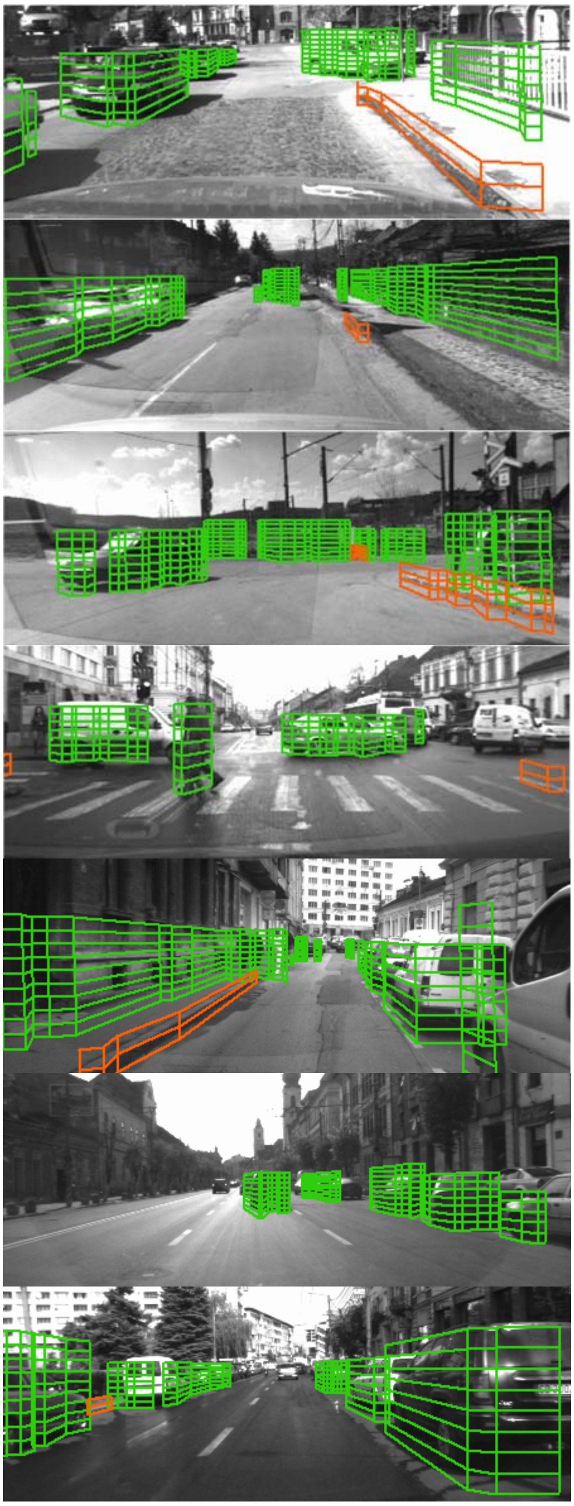


Figure 12: Object delimiters detection through the Combined Border Scanning algorithm for various traffic scenes. The delimiters are projected onto the Left Image and are represented as grids labeled as Traffic Isles (orange) or Objects (light green). The grid height is the same as the enveloped object by the current delimiter.

REFERENCES

Dellaert, F., Thorpe, C., 1997. Robust Car Tracking using Kalman filtering and Bayesian templates. *In proceedings of Conference on Intelligent Transportation Systems*, vol. 3207, 1997. p. 72-83.

Goncalves, A., Godinho, A., Sequeira, J., 2007. Lowcost sensing for autonomous car driving in highways. *In proceedings of ICINCO2007 - 4th International Conference on Informatics in Control, Automation and Robotics*, Angers, France, 2007.

Gonzales, R. C., Woods, R. E., 2002, *Digital Image Processing*, Addison Wesley, second edition.

Harati, A., Siegwart, R., 2007. A new approach to segmentation of 2d range scans into linear regions. *In Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, CA, USA, October 29 - November 2, 2007.

Joshi, R. R., 2002. Novel metrics for map-matching in in-vehicle navigation system. *In IEEE Intelligent Vehicle Symposium*, Vol. 1, 2002, pp. 36-43.

Kolesnikov, A., Franti, P., 2005, Data reduction of large vector graphics. *Pattern Recognition*, March, 2005, 38 (3), 381-394.

Kolski, S., Ferguson, D., Bellino, M., Siegwart, R., 2006. Autonomous driving in structured and unstructured environments. *In proceedings of IEEE Intelligent Vehicles Symposium*, 2006.

Laviers, K. R., Peterson, G. L., 2004. Cognitive robot mapping with polylines and an absolute space representation. *In proceedings of the IEEE International Conference on Robotics and Automation*, Hilton New Orleans Riverside, New Orleans, LA, USA, April 26-May 1 2004, pp. 3771-3776.

Madhavan, R., 2002. Terrain aided localization of autonomous vehicles. *In Symposium on Automation and Robotics in Construction*, Gaithersburg, September 01, 2002.

Magin, G., Russ, A., 1994. Supporting real-time update of an environment representation for autonomous mobile robots real-time systems. *In EuroMicro Workshop on Real-Time Systems*, Jun 15-17 1994, pp. 34-39.

Marita, T., Oniga, F., Nedevschi, S., Graf, T., Schmidt, R., 2006. Camera calibration method for far range stereovision sensors used in vehicles. *In Proceedings of IEEE Intelligent Vehicles Symposium, (IV2006)*, Tokyo, Japan, June 13-15 2006, pp. 356-363.

Nedevschi, S., Danescu, R., Marita, T., Oniga, F., Pocol, C., Sobol, S., Tomiuc, C. Vancea, C., Meinecke, M. M., Graf, T., To, T. B., Obojski, M. A., 2007. A sensor for urban driving assistance systems based on dense stereovision. In Proceedings of Intelligent Vehicles 2007, Istanbul, June 13-15 2007, pp. 278-286.

Oniga, F., Nedevschi, S., Meinecke, M. M., To, T. B., 2007. Road surface and obstacle detection based on elevation maps from dense stereo. *In Proceedings of the IEEE Intelligent Transportation Systems Conference*, Seattle, WA, 2007, pp. 859-865.

Pijpers, M., 2007. *Sensors in adas*, Universiteit Twente.

Prakash, S., Thomas, S., 2007. Contour tracking with condensation/stochastic search. *In Dept. of CSE, IIT Kanpur*, September 26 2007.

Veeck M., Burgard, W., 2004. Learning polyline maps from range scan data acquired with mobile robots. *In proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2004.

Woodill, J. I., Gordon, G., Buck, R., 2004. Tyzx deepsea high speed stereo vision system. *In Proceedings of the IEEE Computer Society Workshop on Real Time 3-D Sensors and Their Use*, Conference on Computer Vision and Pattern Recognition, 2004.

VISION-BASED AUTONOMOUS APPROACH AND LANDING FOR AN AIRCRAFT USING A DIRECT VISUAL TRACKING METHOD

Tiago F. Gonçalves, José R. Azinheira
IDMEC, IST/TULisbon, Av. Rovisco Pais N.1, 1049-001 Lisboa, Portugal
{t.goncalves, jraz}@dem.ist.utl.pt

Patrick Rives
INRIA-Sophia Antipolis, 2004 Route des Lucioles, BP93, 06902 Sophia-Antipolis, France
Patrick.Rives@sophia.inria.fr

Keywords: Aircraft autonomous approach and landing, Vision-based control, Linear optimal control, Dense visual tracking.

Abstract: This paper presents a feasibility study of a vision-based autonomous approach and landing for an aircraft using a direct visual tracking method. Auto-landing systems based on the *Instrument Landing System* (ILS) have already proven their importance through decades but general aviation stills without cost-effective solutions for such conditions. However, vision-based systems have shown to have the adequate characteristics for the positioning relatively to the landing runway. In the present paper, rather than points, lines or other features susceptible of extraction and matching errors, dense information is tracked in the sequence of captured images using an *Efficient Second-Order Minimization* (ESM) method. Robust under arbitrary illumination changes and with real-time capability, the proposed visual tracker suits all conditions to use images from standard CCD/CMOS to *Infrared* (IR) and radar imagery sensors. An optimal control design is then proposed using the homography matrix as visual information in two distinct approaches: reconstructing the position and attitude (pose) of the aircraft from the visual signals and applying the visual signals directly into the control loop. To demonstrate the proposed concept, simulation results under realistic atmospheric disturbances are presented.

1 INTRODUCTION

Approach and landing are known to be the most demanding flight phase in fixed-wing flight operations. Due to the altitudes involved in flight and the consequent nonexisting depth perception, pilots must interpret position, attitude and distance to the runway using only two-dimensional cues like perspective, angular size and movement of the runway. At the same time, all six degrees of freedom of the aircraft must be controlled and coordinated in order to meet and track the correct glidepath till the touchdown.

In poor visibility conditions and degraded visual references, landing aids must be considered. The *Instrument Landing System* (ILS) is widely used in most of the international airports around the world allowing pilots to establish on the approach and follow the ILS, in autopilot or not, until the decision height is reached. At this point, the pilot must have visual contact with the runway to continue the approach and proceed to the flare manoeuvre or, if it is not the case, to abort. This procedure has proven its reliability through decades but landing aids systems that require onboard equipment are still not cost-effective for most of the general airports. However, in the last years, the *Enhanced Visual Systems* (EVS) based on *Infrared* (IR) allowed the capability to proceed to non-precision approaches and obstacle detection for all weather conditions. The vision-based control system proposed in the present paper intends then to take advantage of these emergent vision sensors in order to allow precision approaches and autonomous landing.

The intention of using vision systems for autonomous landings or simply estimate the aircraft position and attitude (pose) is not new. Flight tests of a vision-based autonomous landing relying on feature points on the runway were already referred by (Dickmanns and Schell, 1992) whilst (Chatterji et al., 1998) present a feasibility study on pose determination for an aircraft night landing based on a model of the *Approach Lighting System* (ALS). Many others have followed in using vision-based control on fixed/rotary wings aircrafts, and even airships, in several goals since autonomous aerial refueling ((Kimmett et al., 2002), (Mati et al., 2006)), stabilization with respect to a target ((Hamel and Mahony, 2002), (Azinheira et al., 2002)), linear structures following ((Silveira et al., 2003), (Rives and Azinheira, 2004), (Mahony and Hamel, 2005)) and, obviously, automatic landing ((Sharp et al., 2002), (Rives and Azinheira, 2002), (Proctor and Johnson, 2004), (Bourquardez and Chaumette, 2007a), (Bourquardez and Chaumette, 2007b)). In these problems, different types of visual features were considered including geometric model of the target, points, corners of the runway, binormalized Plucker coordinates, the three parallel lines of the runway (at left, center and right sides) and the two parallel lines of the runway along with the horizon line and the vanishing point. Due to the standard geometry of the landing runway and the decoupling capabilities, the last two approaches have been preferred in problems of autonomous landing.

In contrast with feature extraction methods, direct or dense methods are known by their accuracy because all the information in the image is used without intermediate processes, reducing then the sources of errors. The usual disadvantage of such method is the computational consuming of the sum-of-squared-differences minimization due to the computation of the Hessian matrix. The *Efficient Second-order Minimization* (ESM) (Malis, 2004) (Behimane and Malis, 2004) (Malis, 2007) method does not require the computation of the Hessian matrix maintaining however the high convergence rate characteristic of the second-order minimizations as the Newton method. Robust under arbitrary illumination changes (Silveira and Malis, 2007) and with real-time capability, the ESM suits all the requirements to use images from the common CCD/CMOS to IR sensors.

In vision-based or visual servoing problems, a planar scene plays an important role since it simplifies the computation of the projective transformation between two images of the scene: the planar homography. The Euclidean homography, computed with the knowledge of the calibration matrix of the imagery sensor, is here considered as the visual signal to use into the control loop in two distinct schemes. The *position-based visual servoing* (PBVS) uses the recovered pose of the aircraft from the estimated projective transformation whilst the *image-based visual servoing* (IBVS) uses the visual signal directly into the control loop by means of the interaction matrix. The controller gains, from standard output error LQR design with a PI structure, are common for both schemes whose results will be then compared with a sensor-based scheme where precise measures are considered.

The present paper is organized as follows: In the Section 2, some useful notations in computer vision are presented, using as example the rigid-body motion equation, along with an introduction of the considered frames and a description of the aircraft dynamics and the pinhole camera models. In the same section, the two-view geometry is introduced as the basis for the IBVS control law. The PBVS and IBVS control laws are then presented in the Section 3 as well as the optimal controller design. The results are shown in Section 4 while the final conclusions are presented in Section 5.

2 THEORETICAL BACKGROUND

2.1 Notations

The rigid-body motion of the frame b with respect to frame a by ${}^a\mathbf{R}_b \in \mathbb{SO}(3)$ and ${}^a\mathbf{t}_b \in \mathbb{R}^3$, the rotation matrix and the translation vector respectively, can be expressed as

$$ {}^a\mathbf{X} = {}^a\mathbf{R}_b{}^b\mathbf{X} + {}^a\mathbf{t}_b \tag{1} $$

where, ${}^a\mathbf{X} \in \mathbb{R}^3$ denotes the coordinates of a 3D point in the frame a or, in a similar manner, in homogeneous coordinates as

$$ {}^a\mathcal{X} = {}^a\mathbf{T}_b{}^b\mathcal{X} = \begin{bmatrix} {}^a\mathbf{R}_b & {}^a\mathbf{t}_b \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} {}^b\mathbf{X} \\ 1 \end{bmatrix} \tag{2} $$

where, ${}^a\mathbf{T}_b \in \mathbb{SE}(3)$, $\mathbf{0}$ denotes a matrix of zeros with the appropriate dimensions and ${}^a\mathcal{X} \in \mathbb{R}^4$ are the homogeneous coordinates of the point ${}^a\mathbf{X}$. In the same way, the Coriolis theorem applied to 3D points can also be expressed in homogenous coordinates, as a result of the derivative of the rigid-body motion relation in Eq. (2),

$$ \begin{aligned} {}^a\dot{\mathcal{X}} &= {}^a\dot{\mathbf{T}}_b{}^b\mathcal{X} = {}^a\dot{\mathbf{T}}_b{}^a\mathbf{T}_b^{-1}{}^a\mathcal{X} = \\ &= \begin{bmatrix} \widehat{\omega} & \mathbf{v} \\ \mathbf{0} & 0 \end{bmatrix} {}^a\mathcal{X} = {}^a\widehat{\mathbf{V}}_{ab}{}^a\mathcal{X}\,, \quad {}^a\widehat{\mathbf{V}}_{ab} \in \mathbb{R}^{4\times4} \end{aligned} \tag{3} $$

where, the angular velocity tensor $\widehat{\omega} \in \mathbb{R}^{3\times3}$ is the skew-symmetric matrix of the angular velocity vector ω such that $\omega \times \mathbf{X} = \widehat{\omega}\mathbf{X}$ and the vector ${}^a\mathbf{V}_{ab} = [\mathbf{v}, \omega]^\top \in \mathbb{R}^6$ denotes the velocity screw and indicates the velocity of the frame b moving relative to a and viewed from the frame a. Also important in the present paper is the definition of stacked matrix, denoted by the superscript " ^{s} ", where each column is rearranged into a single column vector.

2.2 Aircraft Dynamic Model

Let $\mathcal{F}_0$ be the earth frame, also called NED for North-East-Down, whose origin coincides with the desirable touchdown point in the runway. The latter, unless explicitly indicated and without loss of generality, will be considered aligned with North. The aircraft linear velocity $\mathbf{v} = [u, v, w] \in \mathbb{R}^3$, as well as its angular velocity $\omega = [p, q, r] \in \mathbb{R}^3$, is expressed in the aircraft body frame $\mathcal{F}_c$ whose origin is at the center of gravity where u is defined towards the aircraft nose, v towards the right wing and w downwards. The attitude, or orientation, of the aircraft with respect to the earth frame $\mathcal{F}_0$ is stated in terms of Euler angles $\Phi = [\phi, \theta, \psi] \subset \mathbb{R}^2$, the roll, pitch and yaw angles respectively.

The aircraft motion in atmospheric flight is usually deduced using Newton's second law and considering the motion of the aircraft in the earth frame $\mathcal{F}_0$, assumed as an inertial frame, under the influence of forces and torques due to gravity, propulsion and aerodynamics. As mentioned above, both linear and angular velocities of the aircraft are expressed in the body frame $\mathcal{F}_b$ as well as for the considered forces and moments. As a consequence, the Coriolis theorem must be invoked and the kinematic equations appear naturally relating the angular velocity rate ω with the time derivative of the Euler angles $\dot{\Phi} = \mathbf{R}^{-1}\omega$ and the instantaneous linear velocity $\mathbf{v}$ with the time derivative of the NED position $\left[\dot{N}, \dot{E}, \dot{D}\right]^\top = \mathbf{S}^\top \mathbf{v}$.

In order to simplify the controller design, it is common to linearize the non-linear model around an given equilibrium flight condition, usually a function of airspeed $\mathbf{V}_0$ and altitude h_0. This equilibrium or trim flight is frequently chosen to be a steady wings-level flight, without presence of wind disturbances, also justified here since non-straight landing approaches are not considered in the present paper. The resultant linear model is then function of the perturbation in the state vector $\mathbf{x}$ and in the input vector $\mathbf{u}$ as

$$\begin{bmatrix} \dot{\mathbf{x}}_v \\ \dot{\mathbf{x}}_h \end{bmatrix} = \begin{bmatrix} \mathbf{A}_v & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_h \end{bmatrix} \begin{bmatrix} \mathbf{x}_v \\ \mathbf{x}_h \end{bmatrix} + \begin{bmatrix} \mathbf{B}_v \\ \mathbf{B}_h \end{bmatrix} \begin{bmatrix} \mathbf{u}_v \\ \mathbf{u}_h \end{bmatrix} \tag{4}$$

describing the dynamics of the two resultant decoupled, lateral and longitudinal, motions. The longitudinal, or vertical, state vector is $\mathbf{x}_v = [u, w, q, \theta]^\top \in \mathbb{R}^4$ and the respective input vector $\mathbf{u}_v = [\delta_E, \delta_T]^\top \in \mathbb{R}^2$ (elevator and throttle) while, in the lateral case, the state vector is $\mathbf{x}_h = [v, p, r, \phi]^\top \in \mathbb{R}^4$ and the respective input vector $\mathbf{u}_h = [\delta_A, \delta_R]^\top \in \mathbb{R}^2$ (ailerons and rudder). Because the equilibrium flight condition is slowly varying for manoeuvres as the landing phase, the linearized model in Eq. (4) can then be considered constant along all the glidepath.

2.3 Pinhole Camera Model

The onboard camera frame $\mathcal{F}_c$, rigidly attached to the aircraft, has its origin at the center of projection of the camera, also called pinhole. The corresponding z-axis, perpendicular to the image plane, lies on the optical axis while the x- and y- axis are defined towards right and down, respectively. Note that the camera frame $\mathcal{F}_c$ is not in agreement with those usually defined in flight mechanics.

Let us consider a 3D point $\mathcal{P}$ whose coordinates in the camera frame $\mathcal{F}_c$ are ${}^c\mathbf{X} = [X, Y, Z]^\top$. This point is perspectively projected onto the normalized image plane $\mathcal{I}_m \in \mathbb{R}^2$, distant one-meter from the center of projection, at the point $\mathbf{m} = [x, y, 1]^\top \in \mathbb{R}^2$ such that

$$\mathbf{m} = \frac{1}{Z}{}^c\mathbf{X}. \tag{5}$$

Note that, computing the projected point $\mathbf{m}$ knowing coordinates $\mathbf{X}$ of the 3D point is a straightforward problems but the inverse is not true because Z is one of the unknowns. As a consequence, the coordinates of the point $\mathbf{X}$ could only be computed up to a scale factor, resulting on the so-called lost of depth perception.

When a digital camera is considered, the same point $\mathcal{P}$ is projected onto the image plane $\mathcal{I}$, whose distance to the center of projection is defined by the focal length $f \in \mathbb{R}^+$, at the pixel $\mathbf{p} = [p_x, p_y, 1] \subset \mathbb{R}^3$ as

$$\mathbf{p} = \mathbf{K}\mathbf{m} \tag{6}$$

where, $\mathbf{K} \in \mathbb{R}^{3\times3}$ is the camera intrinsical parameters, or calibration matrix, defined as follows

$$\mathbf{K} = \begin{bmatrix} f_x & fs & p_{x_0} \\ 0 & f_y & p_{y_0} \\ 0 & 0 & 1 \end{bmatrix} \tag{7}$$

The coordinates $\mathbf{p}_0 = [p_{x_0}, p_{y_0}, 1]^\top \in \mathbb{R}^3$ define the principal point, corresponding to the intersection between the image plane and the optical axis. The parameter s, zero for most of the cameras, is the skew factor which characterizes the affine pixel distortion and, finally, f_x and f_y are the focal lengths in the both directions such that when $f_x = f_y$ the camera sensor presents square pixels.

2.4 Two-views Geometry

Let us consider a 3D point $\mathcal{P}$ whose coordinates ${}^c\mathbf{X}$ in the current camera frame are related with those ${}^0\mathbf{X}$ in the earth frame by the rigid-body motion in Eq. (1) of $\mathcal{F}_c$ with respect to $\mathcal{F}_0$ as

$${}^c\mathbf{X} = {}^c\mathbf{R}_0{}^0\mathbf{X} + {}^c\mathbf{t}_0. \tag{8}$$

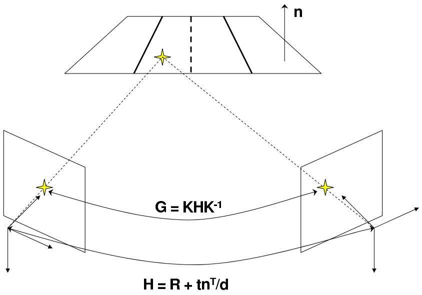


Figure 1: Perspective projection induced by a plane.

Let us now consider a second camera frame denoted reference camera frame $\mathcal{F}_*$ in which the coordinates of the same point $\mathcal{P}$, in a similar manner as before, are

$$ {}^*\mathbf{X} = {}^*\mathbf{R}_0 {}^0\mathbf{X} + {}^*\mathbf{t}_0. \quad (9)$$

By using Eq. (8) and Eq. (9), it is possible to relate the coordinates of the same point $\mathcal{P}$ between reference $\mathcal{F}_*$ and current $\mathcal{F}_c$ camera frames as

$$\begin{aligned} {}^c\mathbf{X} &= {}^c\mathbf{R}_0 {}^*\mathbf{R}_0^\top {}^*\mathbf{X} + {}^c\mathbf{t}_0 - {}^c\mathbf{R}_0 {}^*\mathbf{R}_0^\top {}^*\mathbf{t}_0 = \quad (10)\\ &= {}^c\mathbf{R}_* {}^*\mathbf{X} + {}^c\mathbf{t}_* \end{aligned}$$

However, considering that $\mathcal{P}$ lies on a plane Π, the plane equation applied to the coordinates of the same point in the reference camera frame gives us

$$ {}^*\mathbf{n}^\top {}^*\mathbf{X} = d^* \Leftrightarrow \frac{1}{d^*} {}^*\mathbf{n}^\top {}^*\mathbf{X} = 1 \quad (11)$$

where, ${}^*\mathbf{n}^\top = [n_1, n_2, n_3]^\top \in \mathbb{R}^3$ is the unit normal vector of the plane Π with respect to $\mathcal{F}_*$ and $d^* \in \mathbb{R}^+$ the distance from the plane Π to the optical center of same frame. Thus, substituting Eq. (11) into Eq. (10) results on

$$ {}^c\mathbf{X} = \left({}^c\mathbf{R}_* + \frac{1}{d^*} {}^c\mathbf{t}_* {}^*\mathbf{n}^\top \right) {}^*\mathbf{X} = {}^c\mathbf{H}_* {}^*\mathbf{X} \quad (12)$$

where, ${}^c\mathbf{H}_* \in \mathbf{R}^{3\times3}$ is the so-called Euclidean homography matrix. Applying the perspective projection from Eq. (5) along with the Eq. (6) into the planar homography mapping defined in Eq. (12), the relation between pixels coordinates $\mathbf{p}$ and ${}^*\mathbf{p}$ illustrated in Figure 1 is obtained as follows

$$ {}^c\mathbf{p} \propto \mathbf{K} {}^c\mathbf{H}_* \mathbf{K}^{-1} {}^*\mathbf{p} \propto {}^c\mathbf{G}_* {}^*\mathbf{p} \quad (13)$$

where, $\mathbf{G} \in \mathbb{R}^{3\times3}$ is the projective homography matrix and " $\propto$ " denotes proportionality.

3 VISION-BASED AUTONOMOUS APPROACH AND LANDING

3.1 Visual Tracking

The visual tracking is achieved by directly estimating the projective transformation between the image taken from the airborne camera and a given reference image. The reference images are then the key to relate the motion of the aircraft $\mathcal{F}_b$, through its airborne camera $\mathcal{F}_c$, with respect to the earth frame $\mathcal{F}_0$. For the PBVS scheme, it is the known pose of the reference camera with respect to the earth frame that will allows us to reconstruct the aircraft position with respect to the same frame. What concerns the IBVS, where the aim is to reach a certain configuration expressed in terms of the considered feature, the path planning is then an implicity need of such scheme. For example, if lines are considered as features, the path planning is defined as a function of the parameters which define those lines. In the present case, the path planning shall be defined by images because it is the dense information that is used in order to estimate the projective homography ${}^c\mathbf{G}_*$.

3.2 Visual Servoing

3.2.1 Linear Controller

The standard LQR optimal control technique was chosen for the controller design, based on the linearized models of both longitudinal and lateral motions in Eq. (4). Since not all the states are expected to be driven to zero but to a given reference, the control law is more conveniently expressed as an optimal output error feedback. The objective of the following vision-based control approaches is then to express the respective control laws as a function of the visual information, which is directly or indirectly related with the pose of the aircraft. As a consequence, the pose state vector $\mathbf{P} = [n, e, d, \phi, \theta, \psi]^\top \in \mathbb{R}^6$, in agreement to the type of vision-based control approach, is given differently from the velocity screw $\mathbf{V} = [u, v, w, p, q, r]^\top \in \mathbf{R}^6$, which could be provided from an existent *Inertial Navigation System* (INS) or from some filtering method based on the estimated pose. Thus, the following vision-based control laws are more correctly expressed as

$$ \mathbf{u} = -\mathbf{k_P}(\mathbf{P} - \mathbf{P}^*) - \mathbf{k_V}(\mathbf{V} - \mathbf{V}^*) \quad (14)$$

where, $\mathbf{k_P}$ and $\mathbf{k_V}$ are the controller gains relative to the pose and velocity states, respectively.

3.2.2 Position-based Visual Servoing

In the position-based, or 3D, visual servoing (PBVS) the control law is expressed in the Cartesian space and, as a consequence, the visual information computed into the form of planar homography is used to reconstruct explicitly the pose (position and attitude). The airborne camera will be then considered as only another sensor that provides a measure of the aircraft pose.

In the same way that, knowing the relative pose between the two cameras, $\mathbf{R}$ and $\mathbf{t}$, and the planar scene parameters, $\mathbf{n}$ and d, it is possible to compute the planar homography matrix $\mathbf{H}$ it is also possible to recover the pose from the decomposition of the estimated projective homography $\mathbf{G}$, with the additional knowledge of the calibration matrix $\mathbf{K}$. The decomposition of $\mathbf{H}$ can be performed by singular value decomposition (Faugeras, 1993) or, more recently, by an analytical method (Vargas and Malis, 2007). These methods result into four different solutions but only two are physically admissible. The knowledge of the normal vector $\mathbf{n}$, which defines the planar scene Π, allows us then to choose the correct solution.

Therefore, from the decomposition of the estimated Euclidean homography

$$ {}^{c}\widetilde{\mathbf{H}}_{*} = \mathbf{K}^{-1c}\widetilde{\mathbf{G}}_{*}\mathbf{K}, \tag{15} $$

both ${}^{c}\widetilde{\mathbf{R}}_{*}$ and ${}^{c}\widetilde{\mathbf{t}}_{*}/d^{*}$ are recovered being respectively, the rotation matrix and normalized translation vector. With the knowledge of the distance d^{*}, it is then possible to compute the estimated rigid-body relation of the aircraft frame $\mathcal{F}_{b}$ with respect to the inertial one $\mathcal{F}_{0}$ as

$$ {}^{0}\widetilde{\mathbf{T}}_{b} = {}^{0}\mathbf{T}_{*}\left({}^{b}\mathbf{T}_{c}{}^{c}\widetilde{\mathbf{T}}_{*}\right)^{-1} = \begin{bmatrix} {}^{0}\widetilde{\mathbf{R}}_{b} & {}^{0}\widetilde{\mathbf{t}}_{b} \\ \mathbf{0} & 1 \end{bmatrix} \tag{16} $$

where, ${}^{b}\mathbf{T}_{c}$ corresponds to the pose of the airborne camera frame $\mathcal{F}_{c}$ with respect to the aircraft body frame $\mathcal{F}_{b}$ and ${}^{0}\mathbf{T}_{*}$ to the pose of the reference camera frame $\mathcal{F}_{*}$ with respect to the earth frame $\mathcal{F}_{0}$. Finally, without further considerations, the estimated pose $\widetilde{\mathbf{P}}$ obtained from ${}^{0}\widetilde{\mathbf{R}}_{b}$ and ${}^{0}\widetilde{\mathbf{t}}_{b}$ could then be applied to the control law in Eq. (17) as

$$ \mathbf{u} = -\mathbf{k}_{\mathbf{P}}(\widetilde{\mathbf{P}} - \mathbf{P}^{*}) - \mathbf{k}_{\mathbf{V}}(\mathbf{V} - \mathbf{V}^{*}) \tag{17} $$

3.2.3 Image-based Visual Servoing

In the image-based, or 2D, visual servoing (IBVS) the control law is expressed directly in the image space. Then, in contrast with the previous approach, the IBVS does not need the explicit aircraft pose relative to the earth frame. Instead, the estimated planar homography $\widetilde{\mathbf{H}}$ is used directly into the control law as some kind of pose information such that reaching a certain reference configuration $\mathbf{H}^{*}$ the aircraft presents the intended pose. This is the reason why an IBVS scheme needs implicitly for path planning expressed in terms of the considered features.

In IBVS schemes, an important definition is that of interaction matrix which is the responsible to relate the time derivative of the visual signal vector $\mathbf{s} \in \mathbb{R}^{k}$ with the camera velocity screw ${}^{c}\mathbf{V}_{c*} \in \mathbb{R}^{6}$ as

$$ \dot{\mathbf{s}} = \mathbf{L}_{\mathbf{s}}{}^{c}\mathbf{V}_{c*} \tag{18} $$

where, $\mathbf{L}_{\mathbf{s}} \in \mathbb{R}^{k\times 6}$ is the interaction matrix, or the feature jacobian. Let us consider, for a moment, that the visual signal vector $\mathbf{s}$ is a matrix and equal to the Euclidean homography matrix ${}^{c}\mathbf{H}_{*}$, the visual feature considered in the present paper. Thus, the time derivative of $\mathbf{s}$, admitting the vector ${}^{*}\mathbf{n}/d^{*}$ as slowly varying, is

$$ \dot{\mathbf{s}} = {}^{c}\dot{\mathbf{H}}_{*} = {}^{c}\dot{\mathbf{R}}_{*} + \frac{1}{d^{*}}{}^{c}\dot{\mathbf{t}}_{*}{}^{*}\mathbf{n}^{\top} \tag{19} $$

Now, it is known that both ${}^{c}\dot{\mathbf{R}}_{*}$ and ${}^{c}\dot{\mathbf{t}}_{*}$ are related with the velocity screw ${}^{c}\mathbf{V}_{c*}$, which could be determined using Eq. (3), as follows

$$ \begin{aligned} {}^{c}\widehat{\mathbf{V}}_{c*} &= {}^{c}\dot{\mathbf{T}}_{*}{}^{c}\mathbf{T}_{*}^{-1} = \\ &= \begin{bmatrix} {}^{c}\dot{\mathbf{R}}_{*}{}^{c}\mathbf{R}_{*}^{\top} & {}^{c}\dot{\mathbf{t}}_{*} - {}^{c}\dot{\mathbf{R}}_{*}{}^{c}\mathbf{R}_{*}^{\top}{}^{c}\mathbf{t}_{*} \\ \mathbf{0} & 1 \end{bmatrix} \end{aligned} \tag{20} $$

from where, ${}^{c}\dot{\mathbf{R}}_{*} = {}^{c}\widehat{\omega}\,{}^{c}\mathbf{R}_{*}$ and ${}^{c}\dot{\mathbf{t}}_{*} = {}^{c}\mathbf{v} + {}^{c}\widehat{\omega}\,{}^{c}\mathbf{t}_{*}$. By using such results back in Eq. (19) results on

$$ \begin{aligned} {}^{c}\dot{\mathbf{H}}_{*} &= {}^{c}\widehat{\omega}\left({}^{c}\mathbf{R}_{*} + \frac{1}{d^{*}}{}^{c}\mathbf{t}_{*}{}^{*}\mathbf{n}^{\top}\right) + \frac{1}{d^{*}}{}^{c}\mathbf{v}^{*}\mathbf{n}^{\top} = \\ &= {}^{c}\widehat{\omega}\,{}^{c}\mathbf{H}_{*} + \frac{1}{d^{*}}{}^{c}\mathbf{v}^{*}\mathbf{n}^{\top} \end{aligned} \tag{21} $$

Hereafter, in order to obtain the visual signal vector, the stacked version of the homography matrix ${}^{c}\dot{\mathbf{H}}_{*}^{s}$ must be considered and, as a result, the interaction matrix is given by

$$ \dot{\mathbf{s}} = {}^{c}\dot{\mathbf{H}}_{*}^{s} = \begin{bmatrix} \mathbf{I}(3)^{*}\mathbf{n}_{1}/d^{*} & -{}^{c}\widehat{\mathbf{H}}_{*1} \\ \mathbf{I}(3)^{*}\mathbf{n}_{2}/d^{*} & -{}^{c}\widehat{\mathbf{H}}_{*2} \\ \mathbf{I}(3)^{*}\mathbf{n}_{3}/d^{*} & -{}^{c}\widehat{\mathbf{H}}_{*3} \end{bmatrix} \begin{bmatrix} {}^{c}\mathbf{v} \\ \\ {}^{c}\omega \end{bmatrix} \tag{22} $$

where, $\mathbf{I}(3)$ is the 3×3 identity matrix and $\mathbf{H}_{i}$ is the ith column of the matrix as well as $\mathbf{n}_{i}$ is the ith element of the vector. Note that, $\widehat{\omega}\mathbf{H}$ is the external product of ω with all the columns of $\mathbf{H}$ and $\omega \times \mathbf{H}_{1} = -\mathbf{H}_{1} \times \omega = -\widehat{\mathbf{H}}_{1}\omega$.

However, the velocity screw in Eq. (18), as well as in Eq. (22), denotes the velocity of the reference frame $\mathcal{F}_{*}$ with respect to the airborne camera frame $\mathcal{F}_{c}$ and viewed from $\mathcal{F}_{c}$ which is not in agreement with the aircraft velocity screw that must be applied into the control law in Eq. (17). Instead, the velocity screw shall be expressed with respect to the reference camera frame $\mathcal{F}_{*}$ and viewed from aircraft body frame $\mathcal{F}_{b}$, where the control law is effectively applied. In this manner, and knowing that the velocity tensor ${}^{c}\widehat{\mathbf{V}}_{c*}$ is a skew-symmetric matrix, then

$$ {}^{c}\widehat{\mathbf{V}}_{c*} = -{}^{c}\widehat{\mathbf{V}}_{c*}^{\top} = -{}^{c}\widehat{\mathbf{V}}_{*c} \tag{23} $$

Now, assuming the airborne camera frame $\mathcal{F}_c$ rigidly attached to the aircraft body frame $\mathcal{F}_b$, to change the velocity screw from the aircraft body to the airborne camera frame, the adjoint map must be applied as

$$\begin{aligned} {}^c\widehat{\mathbf{V}} &= {}^b\mathbf{T}_c^{-1}{}^b\widehat{\mathbf{V}}{}^b\mathbf{T}_c = \\ &= \begin{bmatrix} {}^b\mathbf{R}_c^\top{}^b\widehat{\omega}{}^b\mathbf{R}_c & {}^b\mathbf{R}_c^\top{}^b\mathbf{v} + {}^b\mathbf{R}_c^\top{}^b\widehat{\omega}{}^b\mathbf{t}_c \\ \mathbf{0} & 0 \end{bmatrix} \end{aligned} \tag{24}$$

from where, ${}^b\mathbf{R}_c^\top{}^b\widehat{\omega}{}^b\mathbf{R}_c = \widehat{{}^b\mathbf{R}_c^\top{}^b\omega}$ and ${}^b\mathbf{R}_c^\top{}^b\widehat{\omega}{}^b\mathbf{t}_c = -{}^b\mathbf{R}_c^\top{}^b\widehat{\mathbf{t}}_c{}^c\omega$ and, as a result, the following velocity transformation ${}^c\mathbf{W}_b \in \mathbb{R}^{6\times 6}$ is obtained

$${}^c\mathbf{V} = {}^c\mathbf{W}_b{}^b\mathbf{V} = \begin{bmatrix} {}^b\mathbf{R}_c^\top & -{}^b\mathbf{R}_c^\top{}^b\widehat{\mathbf{t}}_c \\ \mathbf{0} & {}^b\mathbf{R}_c^\top \end{bmatrix} \begin{bmatrix} {}^b\mathbf{v} \\ {}^b\omega \end{bmatrix} \tag{25}$$

Using the Eq. (25) into the Eq. (22) along with the result from Eq. (23) results as follows

$$\dot{\mathbf{s}} = {}^c\dot{\mathbf{H}}_*^s = -\mathbf{L_s}{}^c\mathbf{W}_b{}^b\mathbf{V}_{*c} \tag{26}$$

Finally, let us consider the linearized version of the previous result as

$$\mathbf{s} - \mathbf{s}^* = {}^c\mathbf{H}_*^s - \mathbf{H}^{*s} = -\mathbf{L_s}{}^c\mathbf{W}_b{}^b\mathbf{W}_0\left(\mathbf{P} - \mathbf{P}^*\right) \tag{27}$$

where,

$${}^b\mathbf{W}_0 = \begin{bmatrix} \mathbf{S}_0 & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_0 \end{bmatrix} \tag{28}$$

are the kinematic and navigation equations, respectively, linearized for the same trim point as for the aircraft linear model $[\phi,\theta,\psi]_0^\top = [0,\theta_0,0]^\top$. It is then possible to relate the pose error $\mathbf{P} - \mathbf{P}^*$ of the aircraft with the Euclidean homography error ${}^c\mathbf{H}_*^s - \mathbf{H}^{*s}$. For the present purpose, the reference configuration is $\mathbf{H}^* = \mathbf{I}(3)$ which corresponds to match exactly both current $\mathcal{I}$ and reference $\mathcal{I}^*$ images. The proposed homography-based IBVS visual control law is then expressed as

$$\mathbf{u} = -\mathbf{k_P}\left(\mathbf{L_s}{}^c\mathbf{W}_0\right)^\dagger\left({}^c\widetilde{\mathbf{H}}_*^s - \mathbf{H}^{*s}\right) - \mathbf{k_V}(\mathbf{V} - \mathbf{V}^*) \tag{29}$$

where, $\mathbf{A}^\dagger = \left(\mathbf{A}^\top\mathbf{A}\right)^{-1}\mathbf{A}^\top$ is the Moore-Penrose pseudo-inverse matrix.

4 RESULTS

The vision-based control schemes proposed above have been developed and tested in an simulation framework where the non-linear aircraft model is implemented in Matlab/Simulink along with the control aspects, the image processing algorithms in C/C++ and the simulated image is generated by the FlightGear flight simulator. The aircraft model considered corresponds to a generic category B business jet aircraft with $50m/s$ of stall speed, $265m/s$ of maximum speed and $20m$ wing span. This simulation framework has also the capability to generate atmospheric condition effects like fog and rain as well as vision sensors effects like pixels spread function, noise, colorimetry, distortion and vibration of different types and levels.

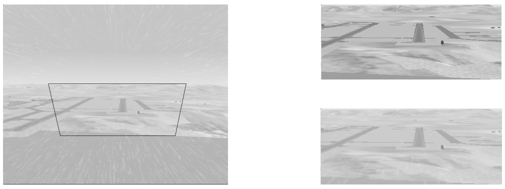


Figure 2: Screenshot from the dense visual tracking software. The delimited zone (left) corresponds to the bottom-right image, warped to match with the top-right image. The warp transformation corresponds to the estimated homography matrix.

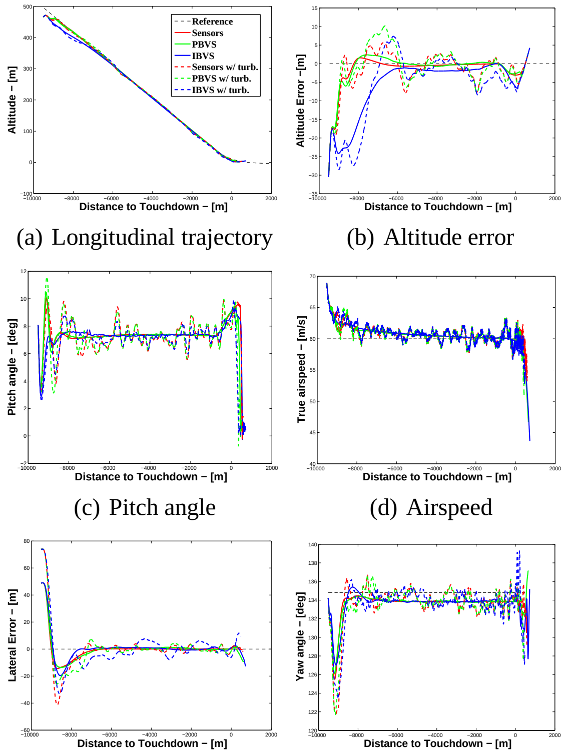
The chosen airport scenario was the Marseille-Marignane Airport with an nominal initial position defined by an altitude of $450m$ and a longitudinal distance to the runway of $9500m$, resulting into a 3 *degrees* descent for an airspeed of $60m/s$. In order to have illustrative results and to verify the robustness of the proposed control schemes, two sets of simulations results are presented. The first with an initial lateral error of $50m$, an altitude error of $30m$ and a steady wind composed by $10m/s$ headwind and $1m/s$ of crosswind. The latter, with a different initial lateral error of $75m$, considers in addition the presence of turbulence. What concerns the visual tracking aspects, a database of $200m$ equidistant images along the runway axis till the $100m$ height, and $50m$ after that, was considered and the following atmospheric conditions imposed: fog and rain densities of 0.4 and 0.8 ([0,1]). The airborne camera is considered rigidly attached to the aircraft and presents the following pose ${}^b\mathbf{P}_c = [4m, 0m, 0.1m, 0, -8\ \mathit{degrees}, 0]^\top$. The simulation framework operates with a $50ms$, or 20Hz, sampling rate.

For all the following figures, the results of the two simulations are presented simultaneously and identified in agreement with the legend in Figure 3(a). When available, the corresponding references are presented in black dashed lines. Unless specified, the detailed discussion of results is referred to the case without turbulence.

Let us start with the longitudinal trajectory illustrated in Figure 3(a) where it is possible to verify immediately that the PBVS results are almost coincident with the ones where the sensor measurements were considered ideal (Sensors). Indeed, because the same control law is used for these two approaches, the results differ only due to the pose estimation errors from the visual tracking software. For the IBVS approach, the first observation goes to the convergence of the aircraft trajectory with respect to the reference descent that occurs later than for the other approaches. This fact is a consequence not only of the limited validity of the interaction matrix in Eq (27), computed for a stabilized descent flight, but also of the importance of the camera orientation over the position, for high altitudes, when the objective is to match two images. In more detail, the altitude error correction in Figure 3(b) shows then the IBVS with the slowest response and, in addition, a static error not greater than 2*m* as a cause of the wind disturbance. In fact, the path planning does not contemplates the presence of the wind, from which the aircraft attitude is dependent, leading to the presence of static errors. These same aspects are verified in the presence of turbulence but now with a global altitude error not greater than 8*m*, after stabilization. The increasing altitude error at the distance of 650*m* before the touchdown corresponds to the natural loss of altitude when proceeding to the pitch-up, or flare, manoeuvre (see Figure 3(c)) in order to reduce the vertical speed and correctly land the aircraft. What concerns the touchdown distances, both Sensors and PBVS results are very close and at a distance around 330*m* after the threshold line while, for the IBVS, this distance is of approximately 100*m*. In the presence of turbulence, these distances became shorter mostly due to the oscillations in the altitude control during the flare manoeuvre. Again, Sensors and PBVS touchdown points are very close and about 180*m* from the runway threshold line while, for the IBVS, this distance is about 70*m*.

The lateral trajectory illustrated in Figure 3(e) shows a smooth lateral error correction for all the three control schemes, where both visual control laws maintain an error below the 2*m* after convergence. Once more, the oscillations around the reference are a consequence of pose estimation errors from visual tracking software, which become more important near the Earth surface due to the high displacement of the pixels in the image and the violation of the planar assumption of the region around the runway. The consequence of these effects are perceptible in the final part not only in the lateral error correction but also in the yaw angle of the aircraft in Figure 3(f). For the latter, the static error is also an influence of the wind disturbance which imposes an error of 1 *degree* with respect to the runway orientation of exactly 134.8 *degrees* North.

In the presence of turbulence, Sensors and PBVS control schemes present a different behavior during the lateral error correction manoeuvre. Indeed, due to the important bank angle and the high pitch induced by the simultaneous altitude error correction manoeuvre, the visual tracking algorithm lost information on the near-field of the camera essential for the precision of the estimated translation. The resultant lateral error estimative, greater than it really is, forces the lateral controller to react earlier in order to minimize such error. As for the longitudinal case, the IBVS presents a slower response on position error corrections resulting into a lateral error not greater than 8*m* which contrasts with the 4*m* from the other two approaches.



(a) Longitudinal trajectory (b) Altitude error

(c) Pitch angle (d) Airspeed

(e) Lateral trajectory (f) Yaw angle

Figure 3: Results from the vision-based control schemes (PBVS and IBVS) in comparison with the ideal situation of precise measurements (Sensors).

It should be noted the precision of the dense visual tracking software. Indeed, the attitude estimation errors are often below 1 *degree* for transient responses and below 0.1 *degrees* in steady state. Depending on the quantity of information available in the near field of the camera, the translation error could vary between the 1*m* and 4*m* for both lateral and altitude errors and between 10*m* and 70*m* for the longitudinal distance. The latter is usually less precise due to its alignment with the optical axis of the camera.

5 CONCLUSIONS

In the present paper, two vison-based control schemes for an autonomous approach and landing of an aircraft using a direct visual tracking method are proposed. For the PBVS solution, where the vision system is nothing more than a sensor providing position and attitude measures, the results are naturally very similar with the ideal case. The IBVS approach based on a path planning defined by a sequence of images shown clearly to be able to correct an initial pose error and land the aircraft under windy conditions. Despite the inherent sensitivity of the vision tracking algorithm to the non-planarity of the scene and the high pixels displacement in the image for low altitudes, a shorter distance between the images of reference was enough to deal with potential problems. The inexistence of a filtering method, as the Kalman filter, is the proof of the robustness of the proposed control schemes and the reliability of the dense visual tracking. This clearly justify further studies to complete the validation and the eventual implementation of this system on a real aircraft.

ACKNOWLEDGEMENTS

This work is funded by the FP6 3rd Call European Commission Research Program under grant Project N.30839 - PEGASE.

REFERENCES

Azinheira, J., Rives, P., Carvalho, J., Silveira, G., de Paiva, E.C., and Bueno, S. (2002). Visual servo control for the hovering of an outdoor robotic airship. In *IEEE International Conference on Robotics and Automation*, volume 3, pages 2787–2792.

Behimane, S. and Malis, E. (2004). Real-time image-based tracking of planes using efficient second-order minimization. In *IEEE International Conference on Intelligent Robot and Systems*, volume 1, pages 943–948.

Bourquardez, O. and Chaumette, F. (2007a). Visual servoing of an airplane for alignment with respect to a runway. In *IEEE International Conference on Robotics and Automation*, pages 1330–1355.

Bourquardez, O. and Chaumette, F. (2007b). Visual servoing of an airplane for auto-landing. In *IEEE International Conference on Intelligent Robots and Systems*, pages 1314–1319.

Chatterji, G., Menon, P., K., and Sridhar, B. (1998). Vision-based position and attitude determination for aircraft night landing. *AIAA Journal of Guidance, Control and Dynamics*, 21(1).

Dickmanns, E. and Schell, F. (1992). Autonomous landing of airplanes by dynamic machine vision. *IEEE Workshop on Application on Computer Vision*, pages 172–179.

Faugeras, O. (1993). *Three-dimensional computer vision: a geometric view point*. MIT Press.

Hamel, T. and Mahony, R. (2002). Visual servoing of an under-actuated dynamics rigid-body system: an image-based approach. In *IEEE Transactions on Robotics and Automation*, volume 18, pages 187–198.

Kimmett, J., Valasek, J., and Junkins, J. L. (2002). Vision based controller for autonomous aerial refueling. In *Conference on Control Applications*, pages 1138–1143.

Mahony, R. and Hamel, T. (2005). Image-based visual servo control of aerial robotic systems using linear images features. In *IEEE Transaction on Robotics*, volume 21, pages 227–239.

Malis, E. (2004). Improving vision-based control using efficient second-order minimization technique. In *IEEE International Conference on Robotics and Automation*, pages 1843–1848.

Malis, E. (2007). An eficient unified approach to direct visual tracking of rigid and deformable surfaces. In *IEEE International Conference on Robotics and Automation*, pages 2729–2734.

Mati, R., Pollini, L., Lunghi, A., and Innocenti, M. (2006). Vision-based autonomous probe and drogue aerial refueling. In *Conference on Control and Automation*, pages 1–6.

Proctor, A. and Johnson, E. (2004). Vision-only aircraft flight control methods and test results. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*.

Rives, P. and Azinheira, J. (2002). Visual auto-landing of an autonomous aircraft. Reearch Report 4606, INRIA Sophia-Antilopis.

Rives, P. and Azinheira, J. (2004). Linear structure following by an airship using vanishing point and horizon line in visual servoing schemes. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 255–260.

Sharp, C., Shakernia, O., and Sastry, S. (2002). A vision system for landing an unmanned aerial vehicle. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1720– 1727.

Silveira, G., Azinheira, J., Rives, P., and Bueno, S. (2003). Line following visual servoing for aerial robots combined with complementary sensors. In *IEEE International Conference on Robotics and Automation*, pages 1160–1165.

Silveira, G. and Malis, E. (2007). Real-time visual tracking under arbitrary illumination changes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–6.

Vargas, M. and Malis, E. (2007). Deeper understanding of the homography decomposition for vision-base control. Research Report 6303, INRIA Sophia-Antipolis.

LIMITS OF HUMAN INTERACTION IN DYNAMICALLY SIMILAR TELEOPERATION SYSTEMS

Under Unknown Constant Time Delay with Impedance Control

V. Parra-Vega, V. P. García-Alvizo, E. Olguín-Díaz
Robotics and Advance Manufacturing Group, CINVESTAV, Carr. Saltillo-Monterrey, Km 13.5, Saltillo, México
(vicente.garcia, vicente.parra, ernesto.olguin)@cinvestav.edu.mx

L. G. García-Valdovinos
Centro de Ingeniería y Desarrollo Industrial, CIDESI, Querétaro, México
ggarcia@cidesi.mx

Keywords: Teleoperation, Haptic.

Abstract: Bilateral teleoperation system are prone to instability coming out from the time-delay introduced by the indeterministic communication channel. This problem has been subject of intensive research under the assumption of non-equal master-slave teleoperators, however, what are the implications of dynamically similar teleoperation system (DSTS), is there simpler stability relationship and trade offs among several involved system and feedback parameters? When we consider a *linear* DSTS system, there arises the question whether there is analytically any advantage, as it was observed heuristically in several experiments (Cho and Park, 2002). In this paper, the stability analysis of such system is reported under an impedance control scheme (Garcia-Valdovinos, 2006) when the delay is considered constant but unknown. by applying the Llewellyn's and Raisbeck's criteria, it is found and explicit and straightforward relationships between the dynamic and kinematic scaling and the stability of the system. This result explicitly suggests clearly guidelines among key factors, such as time delay, desired velocities and feedback gains in terms of the scaling parameters, arises a clear advantage when dealing with dynamically similar systems. This explains why the transparency of the teleoperation system is improved by augmenting/reducing the dynamic/kinematic scaling factor, for given desired frequency, time delay and feedback gains. Simulations and preliminary experimental results illustrate different cases subject to a number of conditions, which can be very useful to design a physical teleoperation system. A preliminary mechatronics design is presented.

1 INTRODUCTION

A bilateral teleoperation system is composed of a master robot and a slave robot, with a human operator commanding the master robot in order to produce the desired position and contact force trajectories[1] for the remotely located slave robot. In turns, the slave robot follows these trajectories so as to produce contact forces to be sent to the master robot as desired force trajectories. In this way, a force/position-force control system is implemented in the master/slave station with a communication channel introducing delayed position and forces signals (Hokayem and Spong, 1984). It is well known that the source of instability of bilateral teleoperation system appears because the time-delay introduced by the indeterministic communication channel is not passive (Q.W. Deng, 2007). Then, the limits to achieve human operator stable interaction with the slave robot, placed at a remote location, through the master robot, becomes an issue, in particular successful teleoperation requires a certain degree of transparency[2] and/or telepresence[3]. Stable interaction is intuitive and easier when mechanical teleoperators are alike? What are the limits of stable interaction for dynamically similar teleoperation system (DSTS) subject to time-delay in the communication channel?

Furthermore, when we consider a *linear* DSTS, there arises the question whether there is analytically any advantage for being the master and slave robot dynamically similar, through the kinematic and dy-

[1] Depending whether is in contact or not.

[2] Manipulation of the slave robot without any dynamics involved in between.

[3] Sensation of being physically in the remote environment.

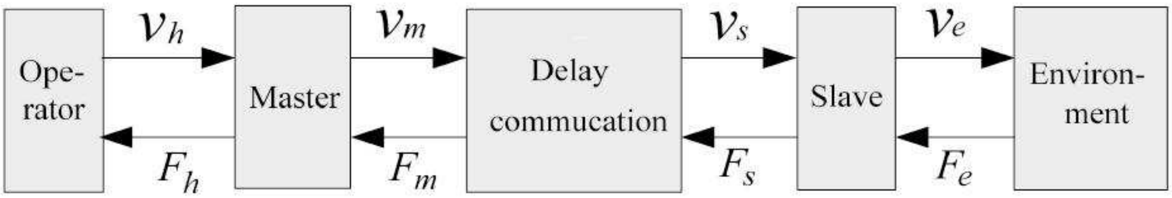


Figure 1: Basic Bilateral Teleoperation Scheme.

namic scaling parameters. It seems DSTS is preferred to carry out over dissimilar teleoperation systems, as it was observed heuristically in several experiments (Cho and Park, 2002).

On one hand, telepresence depends of the degree of transparency, which can be understood as the simultaneous convergence of the position and force error between the master and slave robot. However it also depends on the subjective cognition of the human operator of being there, which depends, among other aspects, on visual aids of the remote environment, kinesthetic coupling, the ability to deal with delayed signals and the man-machine interface. How does these factors are related in DSTS to guarantee stable teleoperation?

In this paper, motivated by the empirical observation that a DSTS is easier to handle, it is argued that dynamic and kinematic similarity introduce a clear trade-off of some of these aspects, so it is reasonable to expect a simpler a trade-off. Analytical results are found based on an impedance control scheme (Garcia-Valdovinos, 2006), when the delay is considered constant but unknown.

1.1 Motivation

When the master and slave teleoperated robots are related linearly by scaling factors of position and force, as well as scaling factors on dynamic and kinematic parameters, a relationship between geometry, power and perception arise to give to the operator a linear relationship behavior. Such scaling factors might give to humans the ability to increase their commanding, perceptual and cognitive skills in different teleoperation tasks, depending on the task undergoing. It has been observed experimentally that by tuning properly these scaling factors a human operator improves his ability to better teleoperate such system (Cho and Park, 2002). Can DSTS yield teleoperation tasks with greater dexterity? What are the trade-offs? It is of interest to understand deeper this phenomena using formal dynamical system tools to analyze properties of stability of dynamically similar teleoperation system.

1.2 Contribution and Organization

Our basic hypothesis is that as long as the human perceives linear correlated variations in both teleoperated robots, he can improve the command of the closed-loop bilateral teleoperation system since spatial and temporal attributes of the visual remote location and kinesthetic coupling will vary linearly without distortion. So cognitively, the human can quickly learn to command the task with greater dexterity. Additionally in this paper we deal with unknown time delay so we design a novel controller to deal with unknown constant time delay (Garcia-Valdovinos, 2006), (Cho and Park, 2002). A computed-torque controller is employed in the master station and a computed torque second order sliding mode controller in the slave station is proposed to produce a desired impedance in closed loop. Then, absolute stability theory and passivity is used to analyze the closed-loop stability properties and therefore the limits of human-teleoperation stability and thus we found the stability trade-offs. To this end, a review is presented in Section 2. Then, in Section 3 the dynamically similar coupled system is presented, while controllers are explained in Section 4. With this result at hand, absolute stability using Llewelyn criteria (Llewellyn, 1952), and passivity using Raisbeck criteria (Raisbeck, 1994), are analyzed in Sections 5 and 6 respectively. The Llewellyn's analysis reveals that a good choice for *dynamic* scaling factors give us the opportunity for greater bounds on position and force scaling to execute tasks of high performance. A quality criterium for transparency analysis is also presented in Section 7. Simulations on a 1 DoF teleoperation systems are shown to illustrate how this dynamic scaling factor improve the performance of the system, shown in Section 8 to better understand the numerical performance. Final conclusions are given in Section 9.

2 DYNAMICALLY SIMILAR DELAYED TELEOPERATION SYSTEM

A dynamically similar teleoperation system has constant scaling factors which relate kinematic and dynamic parameters of the master and slave robots (Goldfarb, 1999). This similarity between the systems is poorly understood so far, since there is not theoretical apparent evident advantage to work out with bilateral system, despite some analysis reported in (Li and Lee, 2003), where the advantages has not been addressed properly in terms of explicit trade-offs of feedback gains, system parameters, desired trajectories and time delay. We surmise in this paper that DSTS improves significantly the ability of humans operating the master teleoperator to carry out efficiently teleoperation tasks at remote environments, when there is an unknown time delay involved in the communication channel.

Impedance control has been explored in (Cho and Park, 2002) to enforce a desired impedance dynamics in closed loop in order to program arbitrarily the desired impedance parameters. This closed-loop linear dynamic allows to model the entire system as a 2-port network to relate the force and flows of input and output, respectively, by an impedance matrix or an hybrid matrix. This matrix can be used to describe the stability of the entire system using the Absolute Stability Theory (K. Hashtrudi-Zaad, 2000), where a tight relationship between *output* scaling factors and impedance parameters can be found to give sufficient conditions on stability. However, when *dynamic* scaling factors are introduced, the master and slave robot dynamics are related by constants, either in kinematic and/or dynamic parameters, thus, a sort of advantages can emerge from this relation since a single feedback parameter appears. In this paper, we offer an analysis in terms of both the Llewelyn's criteria and Raisbeck's criteria, and verify its real time performance, which demonstrates a clear and intuitive trade off in terms of scaling factors of the DSTS.

3 DYNAMICALLY SIMILAR SYSTEM

Consider the dynamics of a linear teleoperation system consisting of two n-DoF manipulators decoupled systems as follows

$$M_m\ddot{x}_m + B_m\dot{x}_m + K_m x_m = F_{mc} + F_m \qquad (1)$$

$$M_s\ddot{x}_s + B_s\dot{x}_s + K_s x_s = F_{sc} + F_s \qquad (2)$$

where $\ddot{x}_i$ and $\dot{x}_i$ denote acceleration and velocity of the robot i,respectively; F_{mc}, F_{sc} are the control force inputs and F_m, F_s are external forces to the master and slave systems, respectively; and M_i, B_i and K_i with $i = m,s$ are the inertia, dampness and stiffness positive coefficients of the systems. Let $K > 0$ be the kinematic scaling factor that relates both master and slave configuration spaces such that

$$x_s = \frac{x_m}{K} \qquad (3)$$

This system is said dynamically similar after the coordination (3) if there exists a scalar $\zeta > 0$ such that

$$M_m\ddot{x}_m + B_m\dot{x}_m + K_m x_m = \frac{M_s\ddot{x}_s + B_s\dot{x}_s + K_s x_s}{\zeta} \qquad (4)$$

Thus, the dynamic parameters of (1)-(2) are related linearly by

$$\zeta M_m = M_s, \quad \zeta B_m = B_s, \quad \zeta K_m = K_s \qquad (5)$$

The apparent advantage of this dynamic relationship has not been well explored in the context of teleoperation with unknown constant time delay, though there are a lot of heuristical intuition which leads us to conclude that these systems would allow greater kinesthetic coupling with greater manipulability dexterity as consequence. To this end, it is introduced an impedance control system (Garcia-Valdovinos, 2006) to enforce a 2-port closed-loop desired linear system.

4 IMPEDANCE CONTROL LAW

4.1 1 DoF Teleoperation System

Similarly to (1)-(2), let a 1 DoF master/slave teleoperation system be modeled as a mass-spring-damper system, where external master force F_m on the master is nothing but the human commanding force F_h and the external slave force force F_s stands as the environmental contact force F_e, then (1)-(2) becomes

$$M_m\ddot{x}_m + B_m\dot{x}_m + K_m x_m = F_{mc} + F_h \qquad (6)$$

$$M_s\ddot{x}_s + B_s\dot{x}_s + K_s x_s = F_{sc} - F_e \qquad (7)$$

where negative sign appears in F_e due to the positive convention of the inertial frame axis.

4.2 Impedance Control Law for the Master

For completeness, the control law (Garcia-Valdovinos, 2006) is introduced here. Consider the following master controller

$$\begin{aligned} F_{mc} \quad = \quad & -F_h + B_m\dot{x}_m + K_m x_m + \frac{M_m}{M_m}(F_h - \\ & K_f F_e^{dy} - \bar{B}_m\dot{x}_m - \bar{K}_m x_m) \end{aligned} \tag{8}$$

Eq. (8) into (6) gives rise to the following desired impedance equation for the master robot

$$\bar{M}_m\ddot{x}_m + \bar{B}_m\dot{x}_m + \bar{K}_m x_m = F_h - K_f F_e^{dy} \tag{9}$$

where positive $\bar{M}_m, \bar{B}_m, \bar{K}_m$ are the desired inertia, dampness and stiffness for the master robot, respectively, and $F_e^{dy} = F_e(t - T_s)$, being T_s the delay from the slave to the master station. That is, the *master* impedance control law enforces a desired impedance (9) in closed-loop, whose parameters are chosen by the user depending of a specific task, such that:

- when the slave robot *is not* touching the environment, $F_e^{dy} = 0$, then (9) becomes a mass-spring-damper system driven solely by the human force F_h, notice that in this case the controller is in *position* impedance mode..
- when the slave robot *is* touching the environment, $F_e^{dy} > 0$, then (9) becomes a mass-spring-damper system driven by force error $F_h - K_f F_e^{dy}$. In this case, actuators in the master station makes the human perceives a contact force equal to $K_f F_e^{dy}$, while the human virtually recreates, cognitively, the surface of the object according to this vector, through kinesthetic sensations of the scaled and delayed slave contact force, which arise normal at the contact slave point and the visual image coming from the slave station. Notice that when the slave is contact, the master control is in impedance *force* control mode.

To achieve such effects, it is necessary to control the slave robot in impedance position and force control modes, according to the contact regime.

4.3 Impedance Control Law for the Slave (Garcia-Valdovinos, 2006)

Similarly to the master controller, the objective in the slave station is to impose a desired impedance to the slave robot

$$\bar{M}_s\,\ddot{\tilde{x}}_s + \bar{B}_s\,\dot{\tilde{x}}_s + \bar{K}_s\tilde{x}_s = -F_e \tag{10}$$

where positive $\bar{M}_s, \bar{B}_s, \bar{K}_s$ are the desired inertia, dampness and stiffness for the slave robot, respectively. The position tracking error $\tilde{x}_s$ is expressed as follows

$$\tilde{x}_s = x_s - K_p x_m^{dy} \tag{11}$$

where $x_m^{dy} = x_m(t - T_m)$, being T_m the delay from the master to the slave station. Now, let the following control law for the slave robot be

$$\begin{aligned} F_{sc} \quad = \quad & -\frac{M_s}{M_s}\left(\bar{B}_s\dot{x}_s + \bar{K}_s x_s + F_e + K_i\sigma\right) \\ & + M_s K_p \bar{M}_m^{-1}\left(F_h^{dy} - K_f F_e^{dy} - \bar{B}_m\dot{x}_m^{dy}\right. \\ & \left. - \bar{K}_m x_m^{dy}\right) + F_e + B_s\dot{x}_s - K_g\Omega \end{aligned} \tag{12}$$

$$\sigma \quad = \quad \int_0^t \mathrm{sgn}\left(I_e(\tau)\right) d\tau \tag{13}$$

where $F_h^{dy} = F_h(t - T_m)$. Notice that the feedforward term F_e^{dy} allows control without any measurement of the time delay. In any case, notice that F_e^{dy} is available for measurement at any time.

Notice that the gain K_g is a new control variable that weights the extended error variable Ω. The proposed sliding surface I_e is proposed naturally out of (10), that is we want (10) to be the attractive convergent manifold, then the extended error manifold is

$$I_e = \bar{M}_s\ddot{x}_s + \bar{B}_s\dot{x}_s + \bar{K}_s x_s + F_e \tag{14}$$

Then we can build a high order sliding surface Ω as a function of the sliding surface I_e as follows[4]

$$\Omega = \frac{1}{M_m}\left(\int\limits_0^t I_e(\tau)d\tau + \int\limits_0^t\int\limits_0^t \mathrm{sgn}(I_e(\tau))d\sigma d\tau\right) \tag{15}$$

Finally, substituting (12)-(13) into (7) gives rise to the closed-loop error equation for the slave robot:

$$\dot{\Omega} = -\beta\Omega \tag{16}$$

where $\beta = \frac{K_g}{M_s} > 0$ is Lipschitz. Consequently, all closed-loop signals in the slave station are bounded, enforcing exponential convergence of $\Omega \to 0$. Therefore, this chain of implications means that a second order sliding mode is enforced, and a sliding mode arises, at $I_e = 0$, which means that (10) arises in finite-time.

A closer analysis shows that the slave impedance control law enforces a desired impedance in closed-loop whose parameters are chosen by the user depending of a specific task, such that:

- when the slave robot *is not* touching the environment, $F_h^{dy} = 0, F_e = 0$ and (10) becomes an unforced mass-spring-damper system such that $\tilde{x}_s \to 0$ and the slave tracks the desired delayed position and velocities of the master. Notice that when the slave is not in contact, the slave control is in position impedance control mode.

[4] Notice that if (14) converge to zero, then (9) appears and the human would perceive the desired impedance to control at will the slave robot.

- when the slave robot *is* touching the environment, $F_e > 0$ and (10) becomes a mass-spring-damper system driven by the slave contact force F_e. In this case, actuators in the slave station make that the slave robot maintains contact ($| \tilde{x}_s | > 0$) while F_e stays around F_h^{dy}. Notice that when the slave is in contact, the slave control is in impedance *force-position* control mode.

With this result, it is now important to analyze the absolute stability properties to find the conditions under which this result is valid.

5 ABSOLUTE STABILITY ANALYSIS

With the desired impedance imposed by the controllers (8) and (12)-(13), the closed-loop dynamics (9) and (10) can be modeled as a 2-port network. Transforming this dynamic into the frequency domain and doing some algebra, closed-loop system can be represented as

$$\begin{bmatrix} F_h \\ V_s \end{bmatrix} = H \begin{bmatrix} V_m \\ -F_e \end{bmatrix} \tag{17}$$

where H is the so called Hybrid Matrix. Using the relationship (5) the hybrid matrix is built from elements depending of function of the desired master impedance parameter as follows

$$H = \begin{bmatrix} \frac{\bar{M}_m s^2 + \bar{B}_m s + \bar{K}_m}{s} & K_f e^{-T_s s} \\ K_p e^{-T_m s} & \zeta \frac{s}{M_m s^2 + B_m s + K_m} \end{bmatrix} \tag{18}$$

which is fundamental to carry out the implications of a unique dynamic scaling factor. To proceed, it is useful to give the following definition on 2-port systems:

Definition: Absolute Stability Criteria for 2-port Systems: *A two-port system (17)-(18) is absolute stable if it does not exist a set of impedances for which the entire system become unstable. If the network is not absolutely stable, it is potentially unstable. By the conditions of the llewellyn's criteria a 2-port network is absolutely stable if and only if*

1. *h_{11} and h_{22} have no poles in the right half plane*
2. *Any poles of h_{11} and h_{22} on the imaginary axis are simple with real and positives residues*
3. *For all real values of* ω

- $\mathrm{Re}\{h_{11}\} \geq 0$
- $\mathrm{Re}\{h_{22}\} \geq 0$
- $2\mathrm{Re}\{h_{11}\}\mathrm{Re}\{h_{22}\} - \mathrm{Re}\{h_{12}h_{21}\} - |h_{12}h_{21}| \geq 0$

Notice that since the human operator is physically holding with his hand the master robot, it is imperative to ensure stable behavior, thus it is required to guarantee the fulfillment of previous Definition. To this end, notice that conditions 1. and 2. are trivially satisfied with positive impedance parameters. The third condition, when using (18), becomes:

$$\begin{array}{ll} \textbf{A.} & [\cos(T_m + T_s)\omega - 1] K_p K_f + 2\zeta\nu \geq 0 \\ \textbf{B.} & K_p K_f \leq \zeta\nu \end{array} \tag{19}$$

where

$$\nu = \frac{(\bar{B}_m \omega)^2}{(\bar{K}_m - \bar{M}_m \omega^2)^2 + (\bar{B}_m \omega)^2} \tag{20}$$

Inequality (19), necessary for the absolute stability of the system, shows that the scaling factors of position, force, and dynamical similarity ζ are critical for the design and performance of the teleoperation system.

Therefore, the consequences of introducing a dynamic similar system in teleoperation, from the point of view of Absolute Stability, are:

1. A unique similarity factor ζ is introduced, which offers a simpler analysis and easy to tune system.
2. The similarity factor ζ allows to derive simpler conditions of absolute stability.
3. The similarity factor ζ improves the design methodology of teleoperators based in impedance controllers. That is, there is a clear trade-off of all important parameters of the system, depending on the desired performance (ω), impedance parameters $\bar{M}_m, \bar{B}_m \bar{K}_m$, position scaling K_p and force scaling K_f and time delays, a ζ can be found.
4. From (18), the scaling factor ζ allows bigger margin on other parameters, thus the opportunity to improve performance based on the physical structure of the teleoperation system.
5. Due to the fact that both master and slave impedance parameters are related by this factor, the whole set of parameters can be expressed in terms of each other, which minimize the number of parameters implied in the design process making it easier to establish a performance limit.

6 PASSIVITY ANALYSIS

Passivity is a powerful criteria to analysis the energetic coupling of a closed loop system, a more conservative implication in comparison to Lyapunov stability criteria, however since the human operator is physically coupled with a typically mechanical system in closed-loop, it is important to analyze the passivity of the closed-loop system.

A two-port network is said to be passive if for all inputs of energy, the output energy is equal or less than the input energy. If the network is not passive, it is active. Raisbeck's passivity criterion is used to determine the passivity of the system.

Definition: Raisbeck Passivity Criterion for 2-port Systems: *It is said that a 2 port-network is passive if and only if*

1. *The parameters of the hybrid matrix H have no poles in the right half plane*
2. *Any poles of the elements of the hybrid matrix on the imaginary axis are simple and their residues satisfy the following conditions, for all real values of* ω,

- $r_{11} \geq 0, \quad r_{22} \geq 0$
- $r_{11}r_{22} - r_{12}r_{21} \geq 0$
- $4\mathrm{Re}\{h_{11}\}\mathrm{Re}\{h_{22}\} - [\mathrm{Re}\{h_{12}\} + \mathrm{Re}\{h_{21}\}]^2 - [\mathrm{Im}\{h_{12}\} - \mathrm{Im}\{h_{21}\}]^2 \geq 0$

where r_{ij} *denotes the residue of* h_{ij}.

In a similar way, the first two items are satisfied with positive impedance parameters, and the third is satisfied if the following inequality is fulfilled

$$K_p^2 + K_f^2 - 2K_pK_f\cos(T_1+T_2)\omega \leq 4\zeta\nu \tag{21}$$

Then absolute stability is a more relaxed stability criteria than passivity. The passivity condition is necessary to assure a complete energetic stability performance of the closed loop system. Thus, we can choose the dynamic scaling factor in order to have a greater upper and lower bounds to vary the other factors of position and force without affect passivity, because it is of primary interest to maintain passivity since the human is physically holding the mechanical master robot.

7 TRANSPARENCY ANALYSIS

In order to determine the transparency of the system, a quality criterion in teleoperation systems based on the impedance matrix is derived from equations (9) and (10),

$$\begin{bmatrix} F_h \\ F_e \end{bmatrix} = \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{bmatrix} \begin{bmatrix} V_m \\ V_s \end{bmatrix} \tag{22}$$

The elements of the impedance matrix are in function of desired impedance parameters and scaling factors as follows

$$\begin{aligned} Z_{11} &= \frac{\xi}{s}\left(1+\zeta K_pK_f e^{-(T_m+T_s)s}\right) \\ Z_{12} &= -\frac{\xi}{s}K_f e^{-T_s s} \\ Z_{21} &= \frac{\xi}{s}K_p e^{-T_m s} \\ Z_{22} &= -\frac{\xi}{s} \end{aligned}$$

where

$$\xi = \bar{M}_m s^2 + \bar{B}_m s + \bar{K}_m$$

The ideal transparency is reached by the system when the (input) environmental impedance Z_e is equal to the output human impedance Z_h, so that, we have

$$Z_h = Z_e \tag{23}$$

From (22), we have

$$Z_h = Z_{11} + \frac{Z_{12}Z_{21}}{Z_e + Z_{22}} \tag{24}$$

where Z_{11}, Z_{22}, Z_{12}, Z_{21} are the elements of the impedance matrix Z. Expressing the elements of Z in terms of the master impedance and substituting them in (24), it gives

$$\begin{aligned} Z_h &= \xi\left(1+\zeta K_pK_f e^{-(T_m+T_s)s}\right) + \\ &\quad \xi^2 \frac{\left(K_pK_f e^{-(T_m+T_s)s}\right)}{Z_e s^2 - \xi s} \end{aligned} \tag{25}$$

Now, we can analyze the transparency of the system in terms of free motion regime, which means that $Z_e = 0$ ideally, and constrained motion regime $Z_e = \infty$, in the worst case, then we have the following:

1. When $Z_e \to 0$, Z_h becomes (25), such that when the dynamic scaling factor ζ is greater, $Z_h \to 0$, or smaller desired impedance parameters is tuned, the higher transparency is obtained in free motion due to
$$Z_e \to 0 \Rightarrow Z_h \to \frac{1}{\zeta}\frac{\xi}{s}$$
2. In contact tasks $Z_e \to \infty$, ideally so does the output impedance. In this case, the transparency relation (24) becomes
$$Z_h \to Z_{11} = \frac{\xi}{s}\left(1+\zeta K_pK_f e^{-(T_m+T_s)s}\right)$$

Notice that the dynamic scaling factor ζ is directly proportional to Z_h, hence, the greater it is the better the transparency is.

8 SIMULATION

In this section the effect of the dynamic scaling is shown for a 1 DoF teleoperation system. Simulations were made in 3 cases:

1. All dynamic, kinematic and force scaling factors are the unit.
2. The scaling factors of position and force are $K_p = 2$ and $K_f = 0.01$, respectively, and the dynamic scaling factor is $\zeta = 0.1$.
3. $zeta$ is increased to 10 and the rest of the parameters are preserved as in case 2.

A smooth force profile was introduced arbitrarily as the force exerted by the human. This force trajectory was designed in such a way that the teleoperator goes from an initial position to the contact point arriving softly with null velocity. This is in order to avoid large spikes due to hard contact. Once the slave robot is in the contact point, the human begin to apply an intermittently force on the constraint 3.

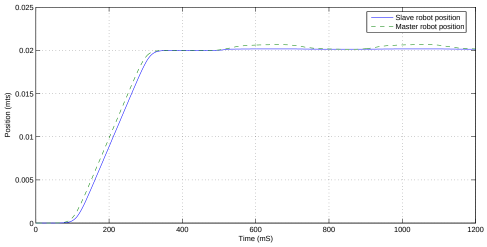


Figure 2: Position error of the master (dashed line) and slave (dotted line) for Case 1.

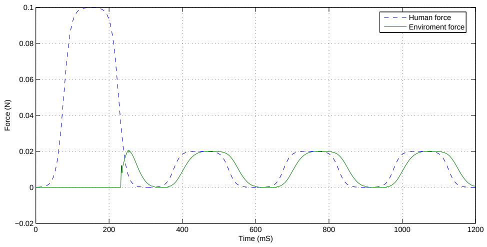


Figure 3: Human force error (master/dashed line) and constraint force error (slave/dotted line) for Case 1.

In Case 1, the chosen scaling factors and the impedance parameters give us an acceptable performance and a stable behavior. The position error between master and slave in Figure 2 shows that both robots follow the same constrained trajectory. Due to the impedance programmed for the master robot a slight movement toward the constraint is allowed. However, the slave robot stands along the physical constraint. With this stable response under constrained operation, the scaling factor of position is increased to obtain a larger workspace with slave robot and the force scaling factor is tuned in order to apply a greater force profile on the constraint and protect the human to receive a large reflected force that could be potentially dangerous.

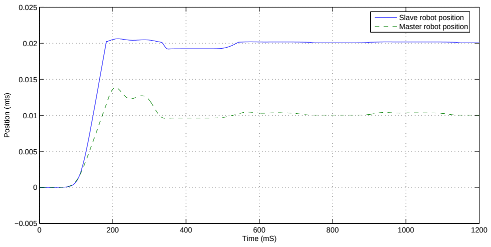


Figure 4: Position error of the master (dashed line) and slave (dotted line) for Case 2.

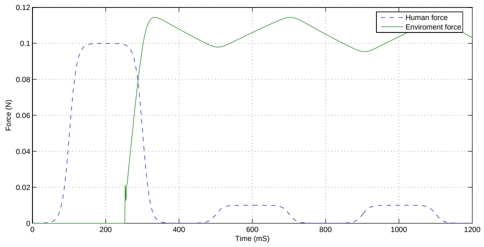


Figure 5: Human force error (master/dashed line) and constraint force error (slave/dotted line) for Case 2.

In Case 2, the change described in the scaling factors make the system unstable. The slave robot go away from its position as can be seen in Figure 4, while the force at the constraint (see Figure 5) disappears.

In order to handle this behavior, the dynamic scaling factor is increased (Case 3). Then the slave robot can reach the scaled position and force as shown in Figures 6 and 7.

The results in simulation show that in case 1, we obtain a acceptable performance and a stable behavior in the system but when we try to increase the performance changing the values of the others factors the

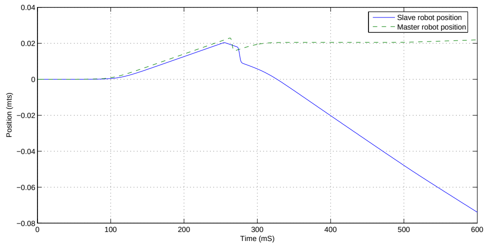


Figure 6: Position error of the master (dashed line) and slave (dotted line) for Case 3.

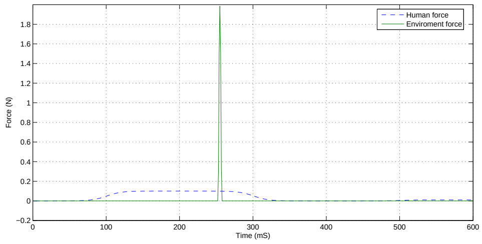


Figure 7: Human force error (master/dashed line) and constraint force error (slave/dotted line) for Case 3.

stability of the system is affected (Case 2). By making a new choice on this dynamic scaling factor we can preserve the scaling factors already chosen for an specific task without affect the stability

9 CONCLUSIONS

Using a novel impedance controller and advanced stability tools, precise conditions to guarantee stability, even in harsh conditions, is proposed for dynamically similar bilateral teleoperation robotic system. In this case, this system depends on a constant parameter, which relates explicitly and clearly a trade off between stability, passivity and transparency. The controller enforces convergence in finite time due to the sliding surface, which is nothing but the impedance equation, thus the closed-loop system dynamics is entirely governed by the desired controlled equation. This yields useful boundaries to vary impedance, scaling parameters and frequency, in terms of the bounded ime delay, which in turn allows to introduce a desired performance criteria in terms of surrounding physical conditions. This result seemingly allows to establish a simpler methodology to design dynamically similar teleoperators with a given desired performance in realistic conditions.

REFERENCES

Cho, H. and Park, J. (2002). Impedance controller design of internet-based teleoperation using absolute stability concept. In *IEEE Int.Conf. on Intelligent Robots and Systems*, pages 2256–2261.

Garcia-Valdovinos, L. G. (2006). *Teleoperadores Bilaterales No Lineales de Impedancia Convergente con Ambiente Semiestructurado*. Phd. thesis, CINVESTAV.

Goldfarb, M. (1999). Similarity and invariance in scaled bilateral telemanipulation. *ASME J. Dynam. Syst., Meas., Contr.*, volume 121:79–87.

Hokayem, P. and Spong, M. (1984). Bilateral teleoperation: An historical survey. *IEEE Control System Magazine*, volumen 4:30–34.

K. Hashtrudi-Zaad, S. S. (2000). Analysis and evaluation of stability and performance robustness for teleoperation control architectures. In *IEEE Int.Conf. on Intelligent Robotics and Automation*, pages 3107–3113.

Li, P. and Lee, D. (2003). Passive bilateral feedforward control of linear dynamically similar teleoperated manipulators. In *IEEE Transactions on Robotic and Automation*, volume volume 19, pages 443–456.

Llewellyn, F. (1952). Some fundamental properties of transmission systems. In *Proc. IRE*, volume volume 40, pages 271–283.

Q.W. Deng, Q. Wei, Z. L. (2007). Analysis of absolute stability for time-delay teleoperation systems. *The International Journal of Automation and Computing*, pages 203–207.

Raisbeck, G. (1994). A definition of passive linear metworks in terms of time and energy. *J. of Applied Physics*, volume 25.

SHORT PAPERS

MOBILE PLATFORM SELF-LOCALIZATION IN PARTIALLY UNKNOWN DYNAMIC ENVIRONMENTS

Patrice Boucher, Sousso Kelouwani and Paul Cohen
Perception and Robotics Laboratory, Ecole Polytechnique de Montreal
2500, Chemin de Polytechnique, Montreal, Canada
{*patrice.boucher, sousso.kelouwani, cohen*}*@ai.polymtl.ca*

Keywords: Navigation, Localization, Dynamic environments, Point-based model, Extended Kalman Filter, 2D Point matching, Registration, Robotic platform slipping, Homogeneous matrices.

Abstract: Localization methods for mobile platforms are commonly based on an observation model that matches onboard sensors measures and environmental a priori knowledge. However, their effectiveness relies on the reliability of the observation model, which is usually very sensitive to the presence of unmodelled elements in the environment. Mismatches between the navigation map, itself an imperfect representation of the environment, and actual robot's observations introduce errors that can seriously affect positioning. This article proposes a 2D point-based model for range measurements that works with a new method for 2D point matching and registration. The extended Kalman filter is used in the localization process since it is of the most efficient tool for tracking a robotic platform's configuration in real time. The method minimizes the impact of measurement noise, mismodelling and skidding on the matching procedure and allows the extended Kalman filter observation model to be robust against skidding and unmodelled obstacles. Its $O(n \cdot m)$ complexity enables real-time optimal points matching. Simulation and experiments demonstrate the effectiveness and robustness of the proposed algorithm in dynamic and partially unknown environments.

1 INTRODUCTION

In the context of map-based navigation, a robotic platform must regularly and reliably estimate its configuration (position and orientation) within a known map of the environment. This problem is commonly referred to as the localization problem. By knowing its configuration and perceiving obstacles in the environment, the platform can choose appropriate actions in order to reach a given destination. However, moving efficiently requires an accurate localization method combined with fast real-time sensory data processing. The proposed algorithm in this paper fulfills these two requirements, using extended Kalman filtering with a novel observation model for platform localization.

Proposed by Stanley F. Schmidt in 1970 (Schmidt, 1970), the extended Kalman filter is commonly used for parameters estimation with non linear models subject to noise. The Kalman filter computes a configuration estimate in two steps. The first one is the prediction step, based on the dynamic model of the system. The second step, known as the correction step, is based on an observation model that draws relationships between the platform configuration and key measurements. In indoor environments, the localization of a robotic platform is often based on observations provided by on-board sensors such as laser range finder (Carlson et al., 2008), infrared sensor (Wei et al., 2005) and sonar sensors. Observed features are matched with a priori data about the environment in order to estimate the most plausible platform's configuration.

As explained in (Thrun et al., 2005) and (De Laet et al., 2008), the matching process from range finder data can be addressed with beams-based models, feature-based methods and correlation-based approaches. In order to deal with unmodelled objects, the beams-based and correlation-based approaches compute complex probabilistic functions given the a priori knowledge about the navigation environment (De Laet et al., 2008). Since each range measurement is considered separately, such models do not take advantage of the natural features of the surrounding platform area. On the other hand, despite that feature-based models can be robust against unexpected objects through selectivity, feature extraction and recognition may be computationally expensive and the features must be sufficiently distinctive and numerous. The lack of robustness of observation models against unmodelled objects is usually compensated by adding such objects onto the map through simultaneous localization and map building (SLAM). However, it remains attractive to have at the base a robust observation model without map modification for avoiding complications at upper levels.

For these reasons, we introduce in this paper a 2D point-based approach which works with a local occupancy grid-map instead of using direct sensor measurements or high level features. In this way, the association process is made between a set of points, extracted from the measurements, with a second set extracted from the grid-map. The configuration is then deduced by matching both sets. Two-dimensional point matching involves two main issues : pairing two sets of 2D points and geometrical matching. The most commonly used methods for geometric matching include SVD (Singular Value Decomposition) (Arun et al., 1987), unit quaternions methods (Horn, 1987) and double quaternions methods (Walker et al., 1991). Various approaches also solve the problem of pairing and matching simultaneously. Many of them are based on iterative algorithms as in (Zhang, 1994) and (Ho et al., 2007). Moreover, (Censi et al., 2005) proposes a Hough Scan Matching (HSM) approach based on the Hough Transform. However, these approaches do not explicitly mention the matching error in the mathematical formulation, a fact that cause ambiguity in the accurate evaluation of the homogeneous matrices. Since the approach presented in this paper needs robustness against matching errors caused by unmodelled objects, these methods are not convenient for a robust 2D points observation model.

In summary, the main contributions of this paper are: (1) a fast method of 2D points registration with complexity $O(n \cdot m)$ ($O(n)$ for the geometric matching) that takes into account the presence of matching errors and measurement noise for enabling realistic accuracy evaluation of the homogeneous matrices; (2) a simple and fast 2D point-based observation model that works in presence of unmodelled objects (3) a novel method for robotic platform localization based upon extended Kalman filtering. The rest of the paper is organized in five sections. Section 2 presents a mathematical formulation of the problem. In section 3, a new method for finding 2D homogeneous matrices is presented. In section 4, we present how the overall methodology can be combined with extended Kalman filtering for platform localization. Section 5 presents and discusses experimental results.

2 PROBLEM STATEMENT

The dynamic equation of a robotic platform moving in a 2D plan can be represented at each instant k by :

$$X_{k+1} = f(X_k, V_k) + \psi_k$$

where X_k is the platform state variable at instant k, V_k is the speed of the platform at instant k, ψ_k is the uncertainty (noise) on the dynamic model and $f(.,.)$ is the function used to compute the predicted state.

The observation model is represented by:

$$Z_k = h(X_k) + \xi_k$$

where Z_k represents the observations by the platform sensors, ξ_k is the uncertainty (noise) on sensor observations and $h(.)$ is the function used to get observations when the platform is in state X_k.

In real applications, f and h are non linear. In order to apply Kalman filtering, the Jacobean of f and h are computed over a nominal path. Furthermore, the following assumptions must hold:

1. ψ_k is uncorrelated with the state initial estimate;
2. ψ_k and ξ_k are uncorrelated;
3. ψ_k and ξ_k are zero mean random process.

Some of these assumptions may not hold if the following conditions occur during platform motion:

- The observations are disturbed by unmodelled obstacles;
- The platform slips on the floor.

The aim is to find an observation model that reduces significantly the impact of the platform slipping and the presence of unmodelled obstacles.

3 FINDING OPTIMAL HOMOGENEOUS TRANSFORMATION MATRICES

In this section, we present a generic method for finding homogeneous transformation matrices between two sets of 2D points.

3.1 Problem Definition

Assume 2 sets P and Q of 2D points. Assume that X_k is the state vector of the platform at time k representing its configuration in the navigation environment. P is the set of points measured by the platform sensors at configuration X_k and Q is the set of points given by the navigation map at that configuration. P and Q are called real set and virtual set, respectively:

$$\begin{aligned} P &= \{p_i, i = 1, \ldots, N\} && (1) \\ Q &= \{q_i, i = 1, \ldots, N\} && (2) \end{aligned}$$

We suppose that each pair $\{p_i,\, q_i\}$ corresponds to a single physical point in the environment. We also assume that q_i is obtained by applying the homogeneous transformation (T, R) on p_i, where T is a translation vector and R is the rotation matrix.

$$q_i = T + Rp_i \qquad \forall i \in \{1, \ldots, N\} \qquad (3)$$

In the context of platform navigation with onboard sensors, the set P is affected by measurement noise. Furthermore, the real correspondence between real and virtual points is unknown. We call $\tilde{P}$ the set of noisy measurement and $\tilde{Q}$ the set of virtual points obtained from the map and affected by pairing error:

$$\begin{aligned} \tilde{P} &= \{\tilde{p}_i, i = 1, \ldots, N\} \\ \tilde{Q} &= \{\tilde{q}_i, i = 1, \ldots, N\} \end{aligned}$$

Representing by δ_i^M the measurement error on p_i and by δ_i^C the pairing error affecting $\{p_i, q_i\}$, the following expressions can be written :

$$\begin{aligned} \tilde{p}_i &= p_i + \delta_i^M && \forall i \in \{1, \ldots, N\} && (4) \\ \tilde{q}_i &= q_i + \delta_i^C && \forall i \in \{1, \ldots, N\} && (5) \end{aligned}$$

Pluggins these equations back into equation (3) yields to the expression of the pairing error δ_i^C :

$$\delta_i^C = -T - R\left(\tilde{p}_i - \delta_i^M\right) + \tilde{q}_i \qquad \forall i \in \{1, \ldots, N\} \quad (6)$$

3.2 Computing the Homogeneous Transformation Matrices

The following assumptions are made:

1. The measurement noise and pairing error are gaussian processes with zero mean and variance σ_M^2 and σ_C^2 respectively:

$$\begin{aligned} \delta_i^M &\longrightarrow N(0, \sigma_M^2) \\ \delta_i^C &\longrightarrow N(0, \sigma_C^2) \end{aligned}$$

2. No $\{R, T\}$ other than $\{R^*, T^*\}$ minimizes the quadratic paring error.
3. The expectation of a random variable tends to be equal to its sampling average:

$$\overline{\chi_i} = E\left[\chi_i\right] = \frac{1}{N}\sum_i^N \chi_i$$

Finding T^* as Function of R^*

Inserting expressions (4) and (5) in (3), the translation vector $\tilde{t_i}$ is given by:

$$\begin{aligned} \tilde{t_i} &= T^* + \delta_i^C - R^*\delta_i^M && (7) \\ &= \tilde{q}_i - R^*\tilde{p}_i && (8) \end{aligned}$$

and its expectation is :

$$E[\tilde{t_i}] = T^* = \overline{\tilde{q}_i} - R^*\overline{\tilde{p}_i} \qquad (9)$$

Finding R^*

From equation (7), the expression δ_i^C is computed as a function of T^* and R^* :

$$\delta_i^C = \tilde{t}_i - T^* + R^*\delta_i^M \qquad (10)$$

R^* and T^* must minimize the quadratic pairing error, therefore:

$$J^* = \min E\left[(\delta_i^C)^T \delta_i^C\right] \qquad (11)$$

By putting equation (10) in (11), replacing $E[(\delta_i^M)^T\delta_i^M]$ by σ_M^2 and noticing that $E[(\delta_i^C)^T R^* \delta_i^M] = \sigma_M^2$ since δ_i^C and δ_i^M are correlated via (10), the following expression is obtained:

$$J^* = E\left[(\tilde{t}_i - T^*)^T(\tilde{t}_i - T^*)\right] - \sigma_M^2 \qquad (12)$$

This result shows that the variance of the translation vector $\tilde{t}_i$, defined by ΔT^2, is equal to the sum of the minimum paring error variance and the measurement noise variance :

$$\begin{aligned} \Delta T^2 &= E\left[(\tilde{t}_i - T^*)^T(\tilde{t}_i - T^*)\right] \\ &= E\left[(\delta_i^C)^T\delta_i^C\right] + \sigma_M^2 && (13) \end{aligned}$$

By plugging expressions (7) and (9) into (13), and by using the angle, ϕ^*, associated with the rotation matrix R^*, the expression of the cost function can be rewritten as :

$$\begin{aligned} J^* = {} & \overline{\tilde{q}_i^T\tilde{q}_i} + \overline{\tilde{p}_i^T\tilde{p}_i} - \overline{\tilde{q}}_i^T\overline{\tilde{q}}_i - \overline{\tilde{p}}_i^T\overline{\tilde{p}}_i - \sigma_M^2 && (14) \\ & -2\cos(\phi^*)\left(\overline{\tilde{p}_{ix}\tilde{q}_{ix}} + \overline{\tilde{p}_{iy}\tilde{q}_{iy}} - \overline{\tilde{p}}_{ix}\overline{\tilde{q}}_{ix} - \overline{\tilde{p}}_{iy}\overline{\tilde{q}}_{iy}\right) \\ & -2\sin(\phi^*)\left(\overline{\tilde{p}_{iy}\tilde{q}_{ix}} - \overline{\tilde{p}_{ix}\tilde{q}_{iy}} - \overline{\tilde{p}}_{ix}\overline{\tilde{q}}_{iy} + \overline{\tilde{p}}_{iy}\overline{\tilde{q}}_{ix}\right) \end{aligned}$$

where $\tilde{p}_i = [\tilde{p}_{ix} \quad \tilde{p}_{iy}]^T$ and $\tilde{q}_i = [\tilde{q}_{ix} \quad \tilde{q}_{iy}]^T$. Taking the derivative to be equal to zero, we obtain:

$$\begin{aligned} \phi^* = {} & atan2\left(\overline{\tilde{p}_{ix}\tilde{q}_{iy}} - \overline{\tilde{p}_{iy}\tilde{q}_{ix}} - \overline{\tilde{p}}_{ix}\overline{\tilde{q}}_{iy} + \overline{\tilde{p}}_{iy}\overline{\tilde{q}}_{ix},\right. \\ & \left.\overline{\tilde{p}_{ix}\tilde{q}_{ix}} + \overline{\tilde{p}_{iy}\tilde{q}_{iy}} - \overline{\tilde{p}}_{ix}\overline{\tilde{q}}_{ix} - \overline{\tilde{p}}_{iy}\overline{\tilde{q}}_{iy}\right) && (15) \end{aligned}$$

The optimal rotation matrix can be deduced directly from this expression. Knowing R^*, we can find T^* by using equation (9).

Algorithm

1. Compute :

$$\overline{\tilde{p}_{ix}\tilde{q}_{iy}},\quad \overline{\tilde{p}_{iy}\tilde{q}_{ix}},\quad \overline{\tilde{p}_{ix}\tilde{q}_{ix}},\quad \overline{\tilde{p}_{iy}\tilde{q}_{iy}},$$
$$\overline{\tilde{p}_{ix}},\quad \overline{\tilde{p}_{iy}},\quad \overline{\tilde{q}_{ix}},\quad \overline{\tilde{q}_{iy}}$$

2. Find the angle ϕ^* by using equation (15) ;
3. Compute the rotation matrix $R^*(\phi^*)$;
4. Compute the translation vector T^*(9).

3.3 Evaluation of the Algorithm Complexity

The first step of the algorithm is related to the computation of averages. These operations have a complexity O(n). Steps 2, 3 and 4 are not dependent upon data size. Hence, the overall complexity is O(n).

3.4 Quality of the Homogeneous Transformation Estimation

In order to assess the quality of the result, the accuracy of the optimal translation T^* and angular error ϕ^* must be estimated for ΔT and $\Delta\phi^*$. These values are indispensable for estimating the measurement noise matrix of the Kalman Filter and (if used with Correlation-based approaches) for providing a correlation measure between the both point sets, since ΔT and $\Delta\phi^*$ increase accordingly to the inconsistency between sets by taking into account the matching error expectation.

Translation Vector Covariance Matrix

The covariance of the translation vector, Ω_T^2, is defined as :

$$\Omega_T^2 = E\left[\left(\tilde{t}_i - T^*\right)\left(\tilde{t}_i - T^*\right)^T\right] \qquad (16)$$

and can be expressed as:

$$\Omega_T^2 \quad = \quad \begin{bmatrix} \sigma_x^2 & \sigma_x\sigma_y \\ \sigma_y\sigma_x & \sigma_y^2 \end{bmatrix} \qquad (17)$$

The covariance matrix of the translation error corresponds to the covariance of the pairing error added to the covariance of the measurement noise reoriented so as to minimize the covariance of the translation vector, as followed :

$$\Omega_T^2 = E\left[\delta_i^C(\delta_i^C)^T\right] + R^* E\left[\delta_i^M(\delta_i^M)^T\right](R^*)^T \qquad (18)$$

Orientation Estimation Quality

Unlike for the translation, there is only one optimal rotation matrix for all pairs of 2D point. Moreover, as the optimal orientation minimizes ΔT, we can use ΔT to assess the orientation accuracy with an empirical formula. Considering that the maximum error of the angle is $\pm\pi$, we have:

$$lim_{\Delta T\to 0}\Delta\phi^* \quad = \quad 0 \qquad (19)$$
$$lim_{\Delta T\to\infty}\Delta\phi^* \quad = \quad \pi \qquad (20)$$

The following empirical formula provides a reasonable estimate of the angle accuracy :

$$\Delta\phi^* = \frac{\pi\Delta T}{\Delta T + \Delta T_{hight}} \qquad (21)$$

where ΔT_{hight} is a large translation error necessarily involving bad pairing.

3.5 Robustness Improvement

The previous results are based upon the assumption that the average pairing error is null and that the paring and measurement error are gaussian process. In particular, the presence of unknown elements in the environment entails a systematic pairing error with a non-zero average. Therefore the gaussien process assumption may not hold. Nevertheless, assuming that the proportion of unknown object points is low, equation (6) gives a good approximation of the correlation error for each pair.

In this way, it is reasonable to give more weight to pairs whose pairing error expectation δ^C is lower. To do this, we compute as first step, the transformation matrices that minimize the overall expectation of pairing error by using equations (15) and (9). Then, the correlation error of each pair can be estimated through (6), which, ignoring the measurement noise, leads to:

$$\tilde{\delta_i^c} = -T^* - R^*\tilde{p}_i + \tilde{q}_i \qquad \forall i \in \{1\cdots,N\}$$

From this result, an uncertainty coefficient can be derived as:

$$(\sigma_i^c)^2 = \tilde{\delta_i^c} * \tilde{\delta_i^c} \qquad (22)$$

The weight, γ_i, of each pair can then be defined as:

$$\gamma_i(\sigma_i^c) = \exp\{-\beta(\sigma_i^c)^2\} \qquad (23)$$

where the parameter β, which can be set empirically, determines the rejection rate for erroneous pairs. With appropriate β, this expression allows good weighting distribution among pairs .

Next, R^* and T^* can be updated by minimizing the weighted paring error expectation. The solution is identical to the one presented at (3.2), except that the following weighted averages are used at steps 1 and 2:

$$\overline{\tilde{p}_{ix}\tilde{q}_{iy}\gamma_i},\quad \overline{\tilde{p}_{iy}\tilde{q}_{ix}\gamma_i},\quad \overline{\tilde{p}_{ix}\tilde{q}_{ix}\gamma_i},\quad \overline{\tilde{p}_{iy}\tilde{q}_{iy}\gamma_i},$$
$$\overline{\tilde{p}_{ix}\gamma_i},\quad \overline{\tilde{p}_{iy}\gamma_i},\quad \overline{\tilde{q}_{ix}\gamma_i},\quad \overline{\tilde{q}_{iy}\gamma_i}$$

3.6 Algorithm Summary

From a real points set $\tilde{P} = \{\tilde{p}_i, i = 1,\ldots,N\}$ with M candidates for each real point, such as $C_i = \{c_{ij}, j = 1,\ldots,M\}$, find a set $\tilde{Q} = \{\tilde{q}_i, i = 1,\ldots,N\}$ that depicts the same physical points of the environment linked to the navigation map.

Then,

1. Choose the nearest neighbour of set P :
$$\tilde{q}_i = arg \min_{1\leq j\leq M} dist\left(\tilde{p}_i, c_{ij}\right)$$
where $dist$ is the Euclidean distance.
2. Find R^* and T^* by geometric matching (3.2).
3. Weighting each pair with (23), recalculate R^* and T^* with weighting averages (see 3.5).

Algorithm Complexity

In sub-section (3.3), we showed that geometric matching is of O(n) complexity. As each additional point involves M extra comparisons, then the complexity of *step 1* is directly proportional to the number of points. Therefore, this step is complexity $O(n\cdot m)$. *Step 3* involves one weight computation per point, so the complexity is O(n). Hence, the total method remains complexity $O(n\cdot m)$.

4 EXTENDED KALMAN FILTER FOR LOCALIZATION

In this section, we demonstrate how the proposed points matching method can be used to enhance the robustness of platform localization based on extended Kalman filter.

4.1 Platform Dynamic Model

The platform dynamic model is usually obtained by using the dead reckoning. Assume that $[x_k, y_k, \theta_k]'$, v_k and ω_k represent respectively the configuration X_k, the translation speed and the angular speed at time k. Assume that the integration constant step (which is also the sampling period) is δt. We can then write the following dynamic equations:

$$\begin{aligned}\theta_{k+1} &= \theta_k + \delta t\omega_k\\ x_{k+1} &= x_k + v_k\delta t\cos\theta_k\\ y_{k+1} &= y_k + v_k\delta t\sin\theta_k\end{aligned}$$

It must be noted that the dynamic equation may change according to the choice of platform model. This is not an issue of the proposed approach since this methodology focuses only on the observation model of the Kalman theory.

4.2 2D Point-based Observation Model

This simplest observation model that could be used in conjunction with Kalman filtering is represented by the following expression:

$$\begin{bmatrix} x_{k|k}\\ y_{k|k}\\ \theta_{k|k}\end{bmatrix} = \begin{bmatrix} x_{k|k-1}+T_x^*\\ y_{k|k-1}+T_y^*\\ \theta_{k|k-1}+\phi^*\end{bmatrix} \tag{24}$$

where $\{x_{k|k-1}, y_{k|k-1}, \theta_{k|k-1}\}$ is the configuration of the platform in the navigation map coordinate system, given all observations up to time $k-1$ and $\{x_{k|k}, y_{k|k}, \theta_{k|k}\}$ is the correction based on optimal homogeneous transformation parameters, $\{T_x^*, T_y^*, \phi^*\}$, obtained at time k. The measurement noise covariance matrix is obtained by using equations (17) and (21)

$$\Xi = \begin{bmatrix} \sigma_x^2+\varepsilon & \sigma_x\sigma_y & 0\\ \sigma_y\sigma_x & \sigma_y^2+\varepsilon & 0\\ 0 & 0 & \Delta\phi^{*2}+\varepsilon\end{bmatrix} \tag{25}$$

where the added ε is required in order to keep the filter stable when the value of the translation vector covariance and the uncertainty on the estimated angle are too small. If points are defined in the robot reference frame, transformation matrices to convert them into the map frame must be applied.

When the presence of a large obstacle or multiples unknown obstacles results in severe erroneous pairings between sets P and Q, the elements on the diagonal of the noise covariance matrix Ξ increase accordingly (see expression 18). This will reduce the confidence in the observation model and increase the confidence in the dynamic model. The impact on the configuration estimate will be limited provided the disturbance is not too prolonged. Otherwise, bad observations can cause, with time, the divergence of filter. The combination *large unknown obstacle - no relative motion* is the worst situation for the localization system.

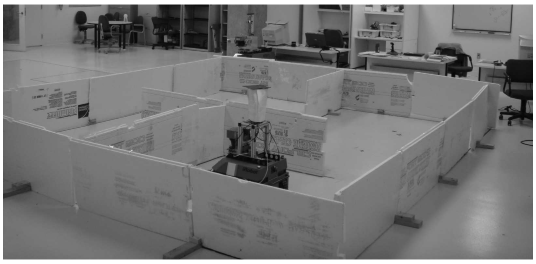


Figure 1: Controlled Navigation Environment.

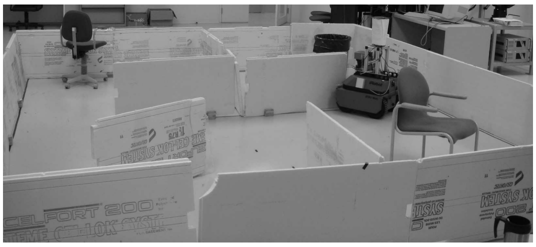


Figure 2: Controlled Navigation Environment with Unmodelled Obstacles.

5 EXPERIMENTAL RESULTS

5.1 Experimentation Setup

Hardware and Software

The approach described in this paper has been implemented in C++ and many simulations have been performed in Matlab and Acropolis (Zalzal, 2006). For real experiments, Acropolis and Player-Stage(Matthias Kranz and Schmidt, 2006) have been used as the robotic framework. The mobile platform hardware is an iRobot Mini ATRV with a differential driving mode and it is equiped on the front with a Sick LMS-200 laser range finder. Laser range data has been down-sampled in order to provide 18 measurements per scan.

Navigation Environment

In order to assess the proposed methodology, a navigation environment has been built. The workspace is delimited by walls as shown on Figure (1). A map of this environment is stored on the platform on-board computer. Figure (2) shows the same navigation environment with additional unmodelled obstacles.

Localization Parameter Settings

- The typical noise magnitude on the translation and angular speeds are set empirically to $0.06m/s$, and $0.06rad/s$ respectively. Note that this noise is inflated in order to reduce the negative impact of slippage on the platform dynamic model. Since this noise reflects confidence toward the dynamic system, an inflated noise magnitude value increases the confidence toward the observation model. In situations where the platform is resistant to slippage, the dynamic model is unbiased and in this case a realistic estimate of noise gives better results.
- The rejection rate β of erroneously paired points, used in equation (23), is set empirically to 0.3.
- The value of the parameter ε, used for numerical stability of equation (25), is set to 0.0001.

External Platform Localization System

In order to obtain an accurate estimate of the platform position in the navigation environment, a second laser range finder, mounted on a fix platform is used. This device detects only the top part of the robotic platform. The positions given by this range finder are used to track the actual trajectory of the platform in the map.

5.2 Simulation for Complexity Assessment

Assessment of the pairing and registration approaches has been realized using Matlab simulation. The goal of the simulation is to find the optimal homogeneous transformation matrices corresponding to two sets of 2D points.

The size of the sets is increased from 2^2 to 2^{15} and set is randomly generated. Given $T = [2 \quad 4]'$ and $\phi = 0.333$ rad, the Q set is generated by applying this transformation matrices to set P. Using the proposed approach, given only P and Q-sets, one should recover exact values of R and T.

Figure (3) plots the execution time as a function of the set size. The average error on the estimation of T^* and ϕ^* is $4.5926e^{-15}$ and $3.2937e^{-16}$ respectively. Furthermore, the figure shows that there is a linear relationship between the set size and the execution time. This result reinforces the claim that the proposed matching method is of complexity O(n).

5.3 Experimental Tests

With the same geometrical trajectory executed repeatedly, the average tracking error is $0.12m$ with a standard deviation of $0.10m$. A typical result is illustrated on Figure (4). The solid line represents the trajectory

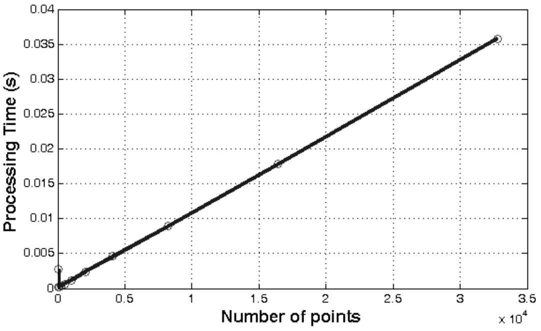


Figure 3: Processing Time as a Function of Data Size.

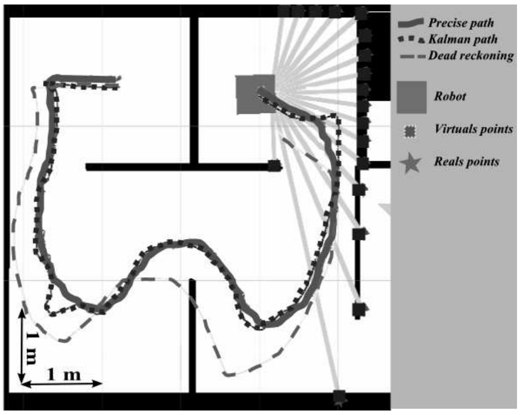


Figure 4: Configuration Tracking in The Navigation Environment.

of the platform reported by the external laser range finder. The dotted line is the trajectory computed by using the approach described in the paper. The dashed line corresponds to navigation with dead reckoning only.

Figure (5) shows an example of severe slippage of the platform during the first quarter of the trajectory. This slippage causes an increasing deviation between the actual trajectory, as reported by the external range finder, and the estimated trajectory based upon dead reckoning. By using the proposed observation model, our estimation is similar to what has been reported by the external range finder.

For the last scenario, several unknown obstacles (unmodelled objects in the navigation map) have been added in the environment. The same geometrical trajectory has been executed repeatedly. As long as the number of observed points corresponding to unknown obstacles remains smaller that the number of points from the known environment, the estimated position of the platform as reported by the approach is still reasonably good. Figure (6) shows a successful case of path following while Figure (7) illustrates a failure of the recovery method. In order to trigger this failure situation, the robot was forced to remain stationary in front of a large unknown obstacle so that the proportion of perceived 2D points from the obstacle become preponderant for a long duration. As such contexts normally involves important mismatches between the two point sets, the measurement noise covariance matrix (equation 25) should increase accordingly, giving priority to the dynamic model. Hence, such unmodelled objects should hardly cause the divergence when the platform moves continuously. Nevertheless, despite that this paper focuses on the localization method without addressing the map building, this kind of failures could be avoided by adding such objects on the map and the general accuracy of the configuration estimate should also get increased.

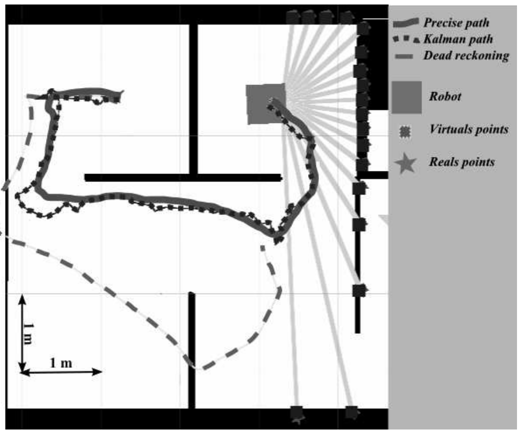


Figure 5: Configuration Tracking with Platform Slipping.

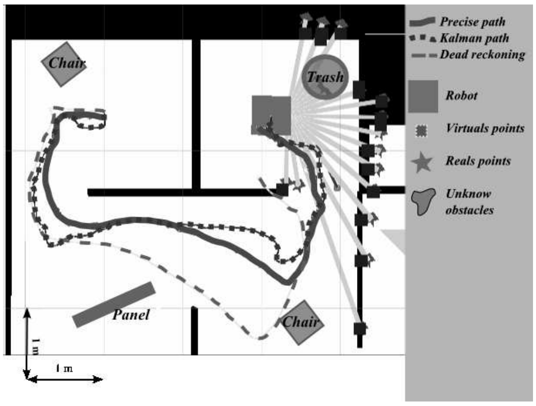


Figure 6: Configuration Tracking with Platform Slipping and Unknown Obstacles.

6 CONCLUSIONS

This article has presented a fast 2D points matching and registration algorithm of complexity $O(n \cdot m)$ that

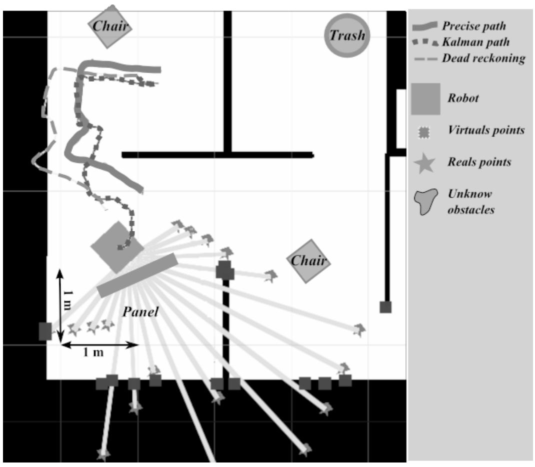


Figure 7: Configuration Tracking Failure.

exhibits robustness against erroneous point pairings. We have shown that this algorithm is easily integrable in a 2D point-based observation model for estimating a platform configuration. When used for robotic platform localization based on extended Kalman filtering, the algorithm provides an accurate estimate of the platform configuration, even in the presence of skidding and unknown obstacles in the environment. The observation model developed in this paper could be used in conjunction with other localization approaches, such as Particle Filter and Monte Carlo filtering.

ACKNOWLEDGEMENTS

This work has been supported by the Natural Science and Engineering Council of Canada (NSERC) through Grant No. CRD 349481-06. The authors wish to acknowledge the contribution of several members of the Perception and Robotics Laboratory during implementation and testing, H. Nguyen, V. Zalzal and R. Gava.

REFERENCES

Arun, K., Huang, T., and Blostein, S. (Sept. 1987). Least-squares fitting of two 3-d point sets [computer vision]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(5):698 – 700.

Carlson, J., Thorpe, C., and Duke, D. L. (2008). Robust real-time local laser scanner registration with uncertainty estimation. *Springer Tracts in Advanced Robotics*, 42:349 – 357.

Censi, A., Iocchi, L., and Grisetti, G. (2005). Scan matching in the hough domain. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 2739–2744.

De Laet, T., De Schutter, J., and Bruyninckx, H. (2008). A Rigorously Bayesian Beam Model and an Adaptive Full Scan Model for Range Finders in Dynamic Environments. *Journal of Artificial Intelligence Research*, 33:179–222.

Ho, J., Yang, M.-H., Rangarajan, A., and Vemuri, B. (2007). A new affine registration algorithm for matching 2d point sets. *Proceedings - IEEE Workshop on Applications of Computer Vision, WACV 2007*.

Horn, B. (1987). Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A (Optics and Image Science)*, 4(4):629 – 42.

Matthias Kranz, Radu Bogdan Rusu, A. M. M. B. and Schmidt, A. (2006). A player/stage system for context-aware intelligent environments. *To appear in Proceedings of the System Support for Ubiquitous Computing Workshop, at the 8th Annual Conference on Ubiquitous Computing (Ubicomp 2006)*.

Schmidt, S. F. (1970). Computational techniques in kalman filtering. *NATO Advisory Group for Aerospace Research and Development*.

Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press.

Walker, M., Shao, L., and Volz, R. (1991). Estimating 3-d location parameters using dual number quaternions. *CVGIP: Image Understanding*, 54(3):358 – 67.

Wei, P., Xu, C., and Zhao, F. (2005). A method to locate the position of mobile robot using extended kalman filter. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 3801 NAI:815 – 820.

Zalzal, V. (2006). *Localisation mutuelle de plates-formes robotiques mobiles par vision omnidirectionnelle et filtrage de Kalman*. PhD thesis, Ecole Polytechnique Montreal (Canada).

Zhang, Z. (1994). Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2):119 – 152.

TELEOPERATION OF A MOBILE ROBOT VIA UMTS LINK

Florian Zeiger, Markus Sauer, Lothar Stolz
Zentrum für Telematik e.V., Allesgrundweg 12, Gerbrunn, Germany
{florian.zeiger, markus.sauer, lothar.stolz}@telematik-zentrum.de

Klaus Schilling
Institute of Computer Science, University of Würzburg, Am Hubland, Würzburg, Germany
schi@informatik.uni-wuerzburg.de

Keywords: Networked robotics, Teleoperation, UMTS, Wireless teleoperation, Wireless networks of robots.

Abstract: Nowadays available broadband wireless communication technologies offer a broad spectrum of applications and the todays availability of UMTS technology has a great potential in the area of networked robotics. This work investigates teleoperation of mobile robots via UMTS communication links. First, several scenarios with artificially generated command traffic between mobile robot and operator are analyzed in order to gain knowledge of the link behavior. In a next step real mobile robot hardware is remote controlled via UMTS. Thereby, the robot transmits sensor data and a video stream of its onboard camera while receiving commands from the operator. The focus of this work is set on the measurement and analysis of the round trip time and the packet inter-arrival time of data in different scenarios like the connection of two UMTS nodes, the connection of an Internet PC and an UMTS node, or the behavior of several UMTS nodes generating a large amount of data. The results clearly show how applications for mobile robot teleoperation can efficiently use UMTS communication in order to allow seamless teleoperation between operator and hardware in distant locations.

1 INTRODUCTION

Communication is a very important issue in the area of mobile robotics. This is especially the case in networked multi-robot systems or joint human-robot teams. Although many advances in autonomy have been made in several applications like e.g. search and rescue still direct teleoperation might be needed. For direct teleoperation (Fong and Thorpe, 2001) the communication link characteristics are even more important than for other applications where data might not be as time-critical. From the human factors point of view situational awareness (Endsley, 2000) is a very critical aspect for robot teleoperation and can be significantly influenced by communication parameters e.g. available bandwidth, delays, packet inter-arrival time, and jitter for the payload data. For direct teleoperation, often the camera image from the robot is one of the major feedback elements for the human operator from the remote environment. Therefore, a relative high communication bandwidth, low delays and a small jitter is desirable for these connections in order to support maintaining situational awareness. Nowadays, in robotics research either wireless LAN according to 802.11 (Ollero et al., 2003) (Zeiger et al., 2008a) or proprietary communication systems are used (Musial et al., 2001) (Pezeshkian et al., 2007). The upcoming high-bandwidth networks for mobile phones or mobile Internet like UMTS offer a new widely used and commercially available technology with high potential for the application in mobile robot teleoperation. Up to now, the coverage of these networks has increased in a way that at least all bigger cities have access to broadband networks. This everywhere availability in large areas is a major advantage for any telematic application compared to a solution where infrastructure initially has to be built up and maintenance effort is necessary. In particular, the application area of service robotics can largely benefit from these characteristics. However , the mobile phone networks like UMTS are designed for different purposes and under different constraints. Therefore, it is important to investigate the critical parameters of a communication technology like UMTS in order to adjust the possible communication parameters on the application layer in a way to realize the optimum usage of this technology.

The remainder of this work is structured as follows. Section 2 gives a brief introduction to the UMTS technology. In Section 3, the test setup is described. In Sections 4 and 5 the evaluations of the tests are given. Also the results of the real mobile robot hardware teleoperation test is presented. This work concludes with a discussion of the results and an outlook in Section 6.

2 UMTS BASICS

The Universal Mobile Telecommunications System (UMTS) is a European standard for third generation mobile telephony (3G). It is based on W-CDMA technology, standardized by the 3GPP, and implemts European specifications of IMT-2000 ITU for third generation cellular radio systems. Although it is not a worldwide standard, UMTS will probably become the most popular one among its competitors. UMTS rigorously divides its infrastructure into two parts (cf. Fig. 1). The UMTS Terrestrial Radio Access Network (UTRAN) handles all tasks related to radio and wireless networking and the Core Network (CN) provides all user services running via UMTS.

The UTRAN maintains the radio connection to mobile operators for circuit and packet switched links. A key to UMTS radio technology is the use of the Wideband Code Division Multiple Access (W-CDMA) user multiplexing procedure. W-CDMA is responsible for ensuring that several participants can communicate simultaneously using the same frequency channel without interfering each other. The UTRAN radio provides two different modes of operation: using UTRA-FDD mode, up- and downlink of data run on separate frequencies. This mode supports data rates up to 384 kbit/s. In UTRA-TDD mode, up- and downlink take place at the same frequency, but are separated by time slots. UTRA-TDD supports data rates up to 2 Mbit/s but is quite uncommon compared to UTRA-FDD. Since demand for higher data rates emerged, UTRAN was enhanced by High Speed Downlink Packet Access (HSDPA) as well as High Speed Uplink Packet Access (HSUPA) to support data rates up to 14.4 Mbit/s on downlink and 5.76 Mbit/s on uplink. The UTRAN is composed of two basic architectural components. Base stations (Node Bs) provide the radio signaling. In addition, there are Radio Network Controllers (RNCs). Managing all radio resources, these RNCs administrate a group of Node Bs. The area covered by all Node Bs connected to one RNC form a Radio Network Subsystem (RNS). These RNSs are linked by cross connections of the RNCs. sThe RNC finally makes the connection to the Core Network (CN), too. Inside the CN, there are several nodes providing various high level services needed for mobile telecommunication. The Mobile Switching Center (MSC) serves as a switching node and gateway for circuit switched connections (phone calls). In case of packet switched data (IP data), these functionality is provided by the Serving GPRS Support Node (SGSN) and Gateway GPRS Support Node (GGSN). For more details on UMTS please refer to (Holma and Toskala, 2007).

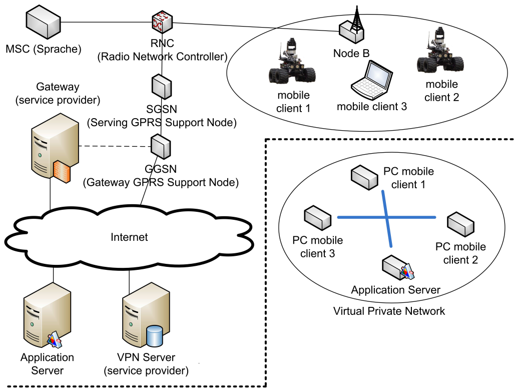


Figure 1: UMTS architecture.

3 SCENARIO SETUP

The focus of this work is set on teleoperation of a mobile robot via UMTS communication link and the characterization of the communication link in order to allow for a seamless teleoperation of the robot. Therefore, three different test setups are analyzed:

- The connection between one robot connected to UMTS and a PC connected to the Internet via DSL (16 MBit/s) (Mode 1).
- The connection of two mobile robots via UMTS (Mode 2). In this case, only the two robots generate traffic.
- The connection of two mobile robots via UMTS while a third UMTS node in the same cell is transmitting a large amount of data to the Internet (Mode 3).

The mobile clients are represented by mini PCs which are connected to the Internet via a USB UMTS device (Huawei 3G modem). This device supports HSDPA and HSUPA broadband data transmission besides the normal UMTS mode. This broadband communication is used for all tests presented in this work. Currently, all UMTS access providers do not provide public IP addresses or in case they do, only a small set of services is supported. Therefore, each UMTS node joins a virtual private network using openvpn (cf. Figure 1). Thus, a physical and hardware component is present, and a logical overlay is set on top of this (cf.

Fig. 1) which enables easy addressing of each mobile node. Also the provided data services can be defined according to the corresponding test. Usually, the UMTS specification promises a reliable packet transmission every 20 milliseconds. To get an idea of the link behavior, for each of the three above mentioned modes, data streams of different sizes are generated. As measurement categories, the packet inter-arrival time is analyzed as this is an important criterion for video and sensor data transmission and the round trip time (rtt) is investigated as this two-way delay is also very important in case the mobile robot is teleoperated directly by a human operator. For the packet inter-arrival time analysis, data is transmitted only one-way and the packet inter-arrival time is plotted at the destination node. Therefore, data is generated with a packet size of 2048 bytes and a packet inter-departure time of 10, 20, 50, and 100 milliseconds which results in packet streams of 20, 40, 100, and 200 kb/s. The round trip time measurements use ICMP ping packets with the size manually set to 2040 bytes of payload and 8 bytes ICMP header. Here, the send intervals for ping packets are also set to 10, 20, 50, and 100 ms which generates data streams comparable to the above described tests for the packet inter-arrival time. Of course, the data stream for the round trip time tests are transmitted in both directions.

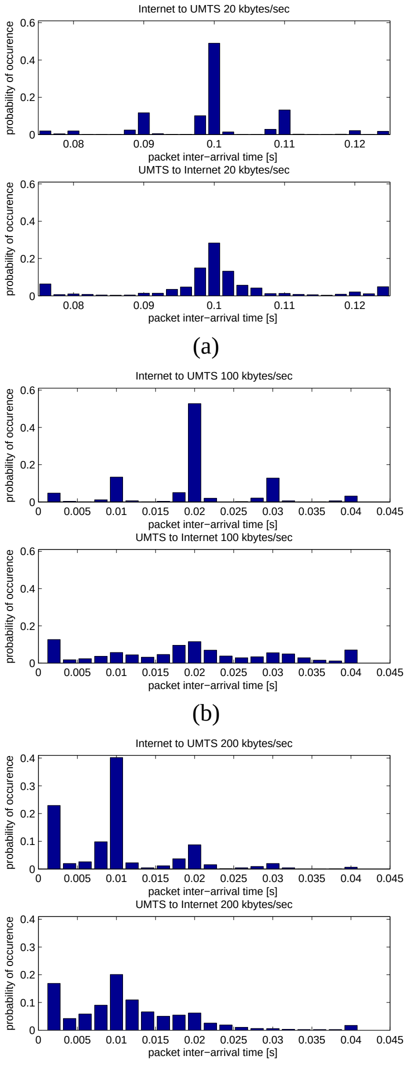
4 EVALUATION

4.1 Packet Inter-arrival Time

For this analysis, the packet inter-arrival time at the destination node is measured. Also the number of packets is counted. The results are displayed in histograms with a bin size of 2 milliseconds. The smallest bin holds all values between 0 and its own value, and the bin with the highest value holds also all larger values up to infinity. The x-axis shows the packet inter-arrival time in seconds and the y-axis shows the relative frequency of occurrence. As the data set of the measurements contains enough data to prove that the resulting distribution is stable, the relative frequency of occurrence can be considered as equal to the probability of occurrence of the corresponding packet inter-arrival time.

4.1.1 Connection between Internet PC and UMTS Node

For this setup, the packet inter-arrival time is measured in both directions for packet streams of 20, 100, and 200 kb/s. Figure 2a shows the results for the 20 kb/s stream. The packet inter-departure time for this stream is 100 milliseconds. The upper subplot of Figure 2a shows, that more than 50% of the packets arrive with an inter-arrival time of 100 ms. Approximately, another 10% arrive with an inter-arrival time of 90 ms and 110 ms respectively. For the opposite direction – from UMTS node to Internet PC (cf. lower subplot of Figure 2a)– the result is completely different. Here, the packet inter-arrival time is closely distributed around the expected packet value of 100 milliseconds.



(a)

(b)

(c)

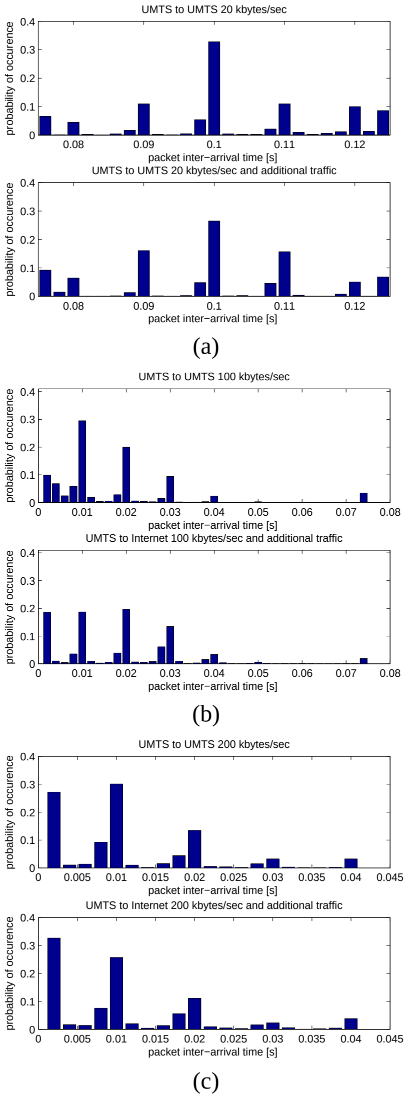
Figure 2: Packet inter-arrival time between Internet PC and UMTS node and a data bandwidth of 20 kb/s (Fig. 2a) , 100 kb/s (Fig. 2b) and 200 kb/s (Fig. 2c).

For the stream with a bit rate of 100 kb/s, the results between the both investigated transmission directions vary significantly (see Figure 2b). At the

UMTS node, more than 50% of the transmitted packets arrive with an inter-arrival-time of 20 ms and two other peaks with 10% each are visible at about 10 ms and 30 ms. For the packet inter-arrival time for the transmission direction from UMTS node to Internet PC, only about 13% of the packets have an inter-arrival time of 20 ms. Here, also small peaks at 10ms and 30 ms ($>$ 6% each), as well as a high amount of packets with 5 ms or less ($>$ 13%) and with 40 ms or more ($>$ 7%) are present. Also for the 200 kb/s stream, the results for both directions are quite different (cf. Figure 2c). For the transmission from Internet PC to UMTS node more than 40% of the packets have an inter-arrival time of 10 ms which corresponds to the packet inter-departure time. More than 20% of the packets have an inter-arrival time of 4ms and less. For the other transmission direction – from UMTS to internet PC – only about 20% of the packets arrive with the expected inter-arrival time of 10 ms. A second peak of about 18% is present for an inter-arrival time of 4 ms and less. Most of the other packets are distributed to inter-arrival times between 4 and 36 ms.

4.1.2 UMTS to UMTS Connection

This section shows the results of transmissions between two UMTS nodes. Each upper subplot of Figures 3a, 3b, and 3c show the packet inter-arrival times when traffic is transmitted only between the two involved nodes. The lower subplots of these Figures show the inter-arrival time of the packets while a third UMTS node transmits a large file to an Internet PC. Thus, this data stream must share the UMTS cell capacity with the measured data stream. Figure 3a shows the results for the 20 kb/s stream. Here, both setups show similar results. The majority of the packets has an inter-arrival time of 100 ms (33% without additional traffic and 28% with additional traffic) and almost all other inter-arrival times are distributed in 10 ms intervals at 80 ms, 90 ms, 110 ms, and 120 ms. For this stream, the additionally generated traffic reduces the amount of the packets with an inter-arrival time of 100 ms for approximately 5%. In Figure 3b the results are displayed for the 100 kb/s stream with a packet inter-departure time of 20 ms. In both situations, almost 20% of the packets have an inter-arrival time of 20 ms. In case of no additionally generated traffic, more than 55% of the packets arrive with an inter-arrival time of less than 10 ms. In case additional traffic is generated, only about 42% of the packets have an inter-arrival time of 10 ms or less. The remaining packet inter-arrival times are distributed at peaks around 30 ms and 40 ms. The results of the 200 kb/s stream displayed in Figure 3c are similar to the above described observations for the 100 kb/s stream.



(a)

(b)

(c)

Figure 3: Packet inter-arrival time between UMTS nodes and a data bandwidth of 20 kb/s (Fig. 3a), 100 kb/s (Fig. 3b), and 200 kb/s (Fig. 3c).

The additionally generated traffic reduces the amount of packets at the expected inter-arrival time of 10 ms for about 5% and the remaining packets are again located at the bins with 10 ms inter-bin distance and inter-arrival times of less than 4 ms, 20 ms, 30 ms, and 40 ms.

Finally, the effective receiving bit rates of the payload data is shown in Tables 1a and 1b. Table 1a shows the results of the test runs between UMTS node and Internet PC and Table 1b shows the results of the test between two UMTS nodes without additionally generated traffic (Mode 2) and between two UMTS nodes with additionally generated traffic (Mode 3). These two tables give an idea of the present packet

Table 1: Resulting effective payload bit rates at the receiving node between UMTS node and Internet PC (Table 1a) and for Mode 2 and Mode 3 (Table 1b).

sending data rate	receiving at UMTS node	receiving at Internet PC
20 kbytes/s	19.20 kbytes/s	19.58 kbytes/s
100 kbytes/s	98.09 kbytes/s	92.62 kbytes/s
200 kbytes/s	194.33 kbytes/s	162.82 kbytes/s

(a)

sending data rate	receiving Mode 2	receiving Mode 3
20 kbytes/s	16.19 kbytes/s	19.31 kbytes/s
100 kbytes/s	94.03 kbytes/s	92.53 kbytes/s
200 kbytes/s	141.80 kbytes/s	125.61 kbytes/s

(b)

loss during the test runs. Surprisingly good results are achieved for all data streams during the transmission from Internet PC to UMTS node. In the opposite direction, an increased packet loss is observed for the 200 kb/s stream. For Mode 2 and Mode 3, where two UMTS nodes communicated with each other, the packet loss was significantly higher.

4.2 Round Trip Times

Figure 4 shows the measured rtts for the generated packet streams. Mode 1 corresponds to the Internet-UMTS node scenario, Mode 2 is the transmission between two UMTS nodes without additionally generated traffic and for Mode 3, the data exchange of two UMTS nodes is measured while a third node transmits additional data to an Internet PC. These boxplots show the median of the measured values (horizontal line) and the box shows 50% of the values bounded by the lower 25% quartile and the upper 75% quartile. Furthermore, lines indicate the most extreme values within 1.5 times the interquartile range from the ends of the box and extreme values lying outside this borders are marked with crosses. For the three streams with 20 kbyte/s, 50% of the measured rtt values are distributed very close to the corresponding median. A similar observation can be made for the 40 kbyte/s stream in Mode 1 and Mode 3, and for the 100 kbyte/s stream in the Mode 1 scenario. As expected, the largest variations appear for the high bandwidth streams with 200 kbyte/s and the 100 kbyte stream in Mode 3 with the additionally generated traffic. In general, 50% of the measured rtt values of each test run are located close to the corresponding median. Only few extreme values are measured below the lower 25% quartile border and sometimes, very high rtt values are measured above the upper border of the 75% quartile (e.g. for 100 kb/s in Mode 3). Thus, the later used application to mobile robotics must consider these high round trip times which occur sometimes and must be able to handle this large variability.

5 REAL HARDWARE TEST

To analyze the UMTS link under real conditions for mobile robot teleoperation, a Pioneer 3-AT is teleoperated from a Laptop via UMTS. The robot platform is equipped with a mini PC and a network video camera AXIS 221 and transmits a motion JPEG video stream with a resolution of 320x240 and a frame rate of 15 frames per second. The Player version 2.0.4 software is used to interface the hardware and for providing communication between the client and the hardware. The client is a Java program which provides a video image of the mobile robot's onboard camera and displays sensor data. During the test runs, the mobile robot generates a sensor data stream and sends it to the Laptop. Additionally, video data is sent each 66.66 ms. The Client program is sending control commands with a bit rate of 1.2 kb/s. In Figure 5, the behavior of the video data which arrives at the operator's PC is shown. The video source generates relatively large packets which must be fragmented. Thus, the send buffers are continuously filled and packets are sent as often as possible. On the receiver's side, the already above observed peaks of the packet inter-arrival times each 10 ms is also visible.

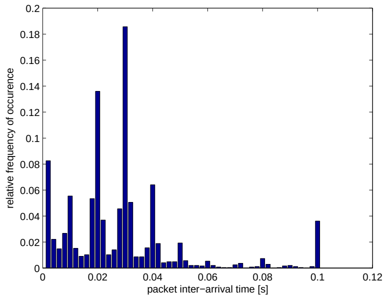


Figure 5: Packet inter-arrival of the video data coming from the mobile robot.

6 DISCUSSION OF THE RESULTS

The above presented results show clearly that UMTS is a well suited communication technology for the teleoperation of mobile robots. The results obtained in the initial test runs with the artificially generated command traffic give a detailed overview of the channel behavior. When data is transmitted from the UMTS node via the UMTS infrastructure and the Internet, the arriving packets at the PC have a higher variance in the packet inter arrival time as typically, the Internet data transfer induces such a variance. In the opposite direction, the packets arrive at the UMTS infrastructure via the Internet. Then the UMTS infrastructure is responsible for further data transmission

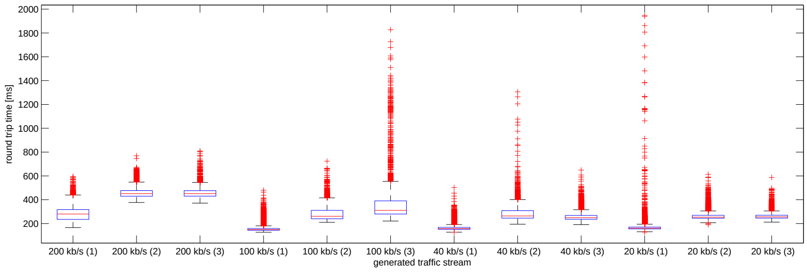


Figure 4: Round trip times in milliseconds (Mode (1): Internet-UMTS; Mode (2): UMTS-UMTS; Mode (3): UMTS-UMTS and additional traffic).

and takes care that packets are transmitted with defined packet inter-departure times. These packet inter-departure times are depending on the used bandwidth and the used HSDPA/HSUPA mode and some other aspects depending on the provider. Nevertheless, the measured packet inter-arrival times are now bound to fixed times which reduces the variance. In general, it can be seen that as soon as the broadband communication mode is entered, data is delivered at the mobile robot and at the operator's PC constantly. Noticeable outcomes are the frequent packet inter-arrival times at 10 ms and multiple of 10 ms with a relatively high peak at the 20 ms inter-arrival times. Also, the received effective payload data rates give a clear view of the present packet loss during the tests. Finally, the measured round trip times and their distribution lead to the following hints for implementing mobile robot teleoperation. The "just plug and try" setup which is described in Section 5 shows to be already usable. Nevertheless, a defined traffic shaping might be a suitable approach to use the characteristics of the UMTS link more efficient and to increase the quality of teleoperation (e.g. better video quality or less packet loss). Approaches for this idea are already published in another context (Zeiger et al., 2008c)(Zeiger et al., 2008b) and shows to be a useful technique. Nevertheless, UMTS is a promising technology to allow broadband communication for hardware teleoperation and will be in the research focus of networked robotics in near future. Future work will be set on analyzing more details of the UMTS link in combination with mobile robot teleoperation. Open issues are how to use the behavior of the communication channel in order to allow for high quality onboard video transmission together with reliable control data transmission in both directions.

ACKNOWLEDGEMENTS

This work is partially supported by the European Union and the Federal State of Bavaria in the frame of the EFRE program.

REFERENCES

Endsley, M. R. (2000). Theoretical underpinnings of situation awareness: A critical review. In Endsley, M. R. and Garland, D. J., editors, *Situation awareness analysis and measurement*, chapter 1, pages 3–26. Lawrence Erlbaum Associates.

Fong, T. and Thorpe, C. (2001). Vehicle teleoperation interfaces. *Auton. Robots*, 11(1):9–18.

Holma, H. and Toskala, A., editors (2007). *WCDMA for UMTS: HSPA Evolution and LTE*. John Wiley & Sons, Ltd, fourth edition.

Musial, M., Brandenburg, U. W., and Hommel, G. (2001). Success of an Inexpensive System Design: The Flying Robot MARVIN. In *16th Int. Unmanned Air Vehicle System Conf. (UAVs)*.

Ollero, A., Alcazar, J., Cuesta, F., Lopez-Pichaco, F., and Nogales, C. (2003). Helicopter teleoperation for aerial monitoring in the comets multi-uav system. In *3rd IARP Workshop on Service, Assistive and Personal Robots (IARP 2003), Madrid (Spain)*.

Pezeshkian, N., Nguyen, H. G., and Burmeister, A. (2007). Unmanned ground vehicle radio relay deployment system for non-line-of-sight operations. In *Proc. of the 13th IASTED Int. Conf. on Robotics and Applications, August 29-31, Würzburg, Germany, RA2007*.

Zeiger, F., Krämer, N., and Schilling, K. (2008a). Commanding mobile robots via wireless ad-hoc networks - a comparison of four ad-hoc routing protocol implementations. In *IEEE Int. Conf. on Robotics and Automation (ICRA 2008)*.

Zeiger, F., Sauer, M., and Schilling, K. (2008b). Intelligent Shaping of a Video Stream for Mobile Robot Teleoperation via Multihop Networks in Real-World Scenarios. In *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2008)*.

Zeiger, F., Sauer, M., and Schilling;, K. (2008c). Video Transmission with Adaptive Quality based on Network Feedback for Mobile Robot Teleoperation in Wireless Multi-Hop Networks. In *5th Int. Conf. on Informatics, Automation and Robotics (ICINCO 2008)*.

NEW METHODS FOR DISHWARE IDENTIFICATION AND INSPECTION

Trung H. Duong and L. L. (Larry) Hoberock
School of Mechanical and Aerospace Engineering, Oklahoma State University, Stillwater, OK 74078, U.S.A.
trungd@okstate.edu, larry.hoberock@okstate.edu

Keywords: Dishware identification, Dishware inspection, Partitioning, Adaptive thresholding, Global thresholding.

Abstract: We propose automatically identifying dishes in mixed batches by using statistics of shape descriptors of dish pieces. Experiments were conducted on 725 images of ceramic and plastic dishes taken in different lighting conditions using different positions of 84 separate dishes of 5 different styles and shapes. In order to find the minimum set of descriptors to produce fast, adaptable and efficient automatic dish recognition, we employed several shape-based properties, including area, perimeter, ratio of length to width, extension, and minimum bounding box, together with some properties based on gray level and color. For dish inspection, we propose a new technique using partitioning and adaptive thresholding, combined with global thresholding. For practical purposes, the algorithm should be fast, simple, and produce results invariant with lighting conditions and dish rotation about the camera-dish axis. Such an algorithm is described in this work. Matlab® R14 and Image Processing Toolbox V5.0 were used.

1 INTRODUCTION

Commercial dishwashing systems currently involve human loading, sorting, inspecting, and unloading dishes and silverware pieces before and after washing in hot and humid environments. In such difficult working conditions, leading to high turn-over of low-paid employees, automation is desirable, especially in large-scale kitchens of hospitals, navy ships, schools, hotels and other dining facilities. Our project is a part of developing an integrated machine vision sorting and inspecting system for mixed dish pieces and silverware exiting a flight-type commercial dishwashing machine, coupled with automatic loading and unloading.

Johnson (1993), working on the same dish set as used in this project, employing area and radius of the corner of the dish in using machine vision identification of dish pieces exiting commercial dishwashing machines. His method required an invariant position of a dish under the camera axis, which required a locating mechanism for each dish piece. Even with pre-location, he reported poor repeatability of results under small lighting variations, such as those due to normal small fluctuations in power supply voltage. For inspection of dishes for cleanliness, his algorithm used simple global thresholding, which is insufficient in accuracy for actual implementation. We present a much improved method for both dish identification and inspection, yielding much better results.

Other investigators studied identification and inspection of mixed silverware pieces exiting a commercial dishwashing machine. Yeri (2002) used blob analysis to recognize silverware pieces, using indirect illuminaiton in a light tent to weaken specular reflections. Lolla (2005) identified silveware objects by their perimeters, symmetric and asymmetric properties, and area moment of inertia measurements. He used edge detection algorithms together with template matching to inspect recognized silverware pieces. Zhou (2008) proposed an algorithm to recognize silverware pieces with incomplete (truncated) images and a fusion-based method for silverware inspection, producing very good results.

The objective of this project is to develop algorithms and procedures for on-line dishware identification and inspection of certain types of dish pieces exiting a commercial dishwasher. Figure 1 illustrates the overall process.

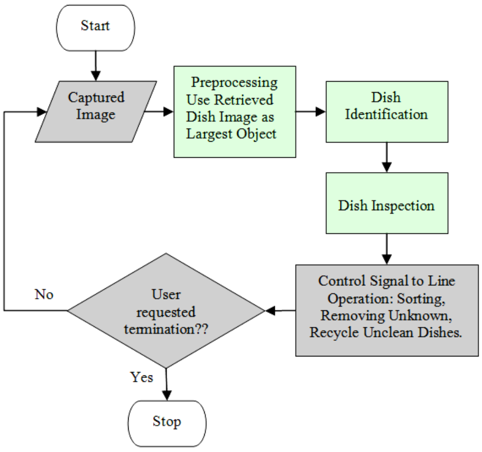

Start
Captured Image
Preprocessing Use Retrieved Dish Image as Largest Object
Dish Identification
Dish Inspection
Control Signal to Line Operation: Sorting, Removing Unknown, Recycle Unclean Dishes.
User requested termination??
No
Yes
Stop

Figure 1: Processing Flow Chart.

2 EXPERIMENTAL APPROACH

2.1 Experimental Setup

The experimental setup, inherited from Zhou (2008) is shown in Fig 2, with our modifications in lighting.

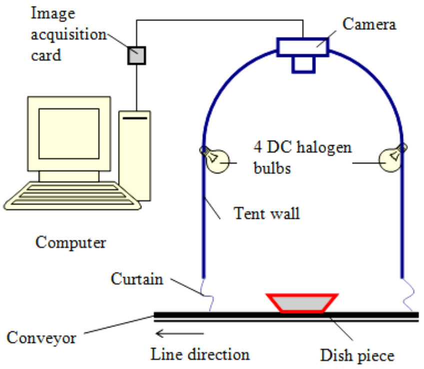


Figure 2: Experimental Setup.

After washing, it is anticipated that dish pieces will be automatically placed on the conveyor. However, in this project, dried dishes were placed manually. While wet dishes can be easily handled with this technique, dried dishes were much easier to handle experimentally. A dish image is captured by the camera when the dish is inside the light tent in the camera field of view. In full implementation, image taking will be triggered by appropriately placed sensors, but in the work herein, this triggering was done manually. This image is transmitted to a computer for image processing. The tent wall and curtains are used to eliminate uncontrolled illumination from the outside environment. In actual implementation, after processing, a signal identifying the type of dish piece will be sent to a sorting mechanism to sort the dish into a stack of like dishes. Unidentified objects will be automatically sent to a bin for such objects, and if the dish is determined by the vision system to be unclean, a signal will be sent to convey said dish into a bin for re-washing, as indicated in Figure 1.

The camera used in this project was an area scan, color digital industrial camera, Basler Co. model A102kc, directly connected to an image processing board in a personal computer, described later, for real time image processing. Sensor size in the camera was approximately 17 mm square. Resolution was set to 1392 by 1038 pixels, which is sufficient to discern a small dirty spot of SFS mm square. The lens was a Fuji model CF35HA-1, 35mm focus length, with 14°26’x10°46’ aperture view cone.

Let sensor size SS= 17mm, sensor resolution SR= 1038 pixels, and focal length FL= 35mm. Choose working distance WD= 600mm, which is large enough to avoid distortion when the lens is focused on an object within the working distance (Zhou, 2008). Then we calculate the smallest feature size, SFS, that the camera can detect (Edmund Industrial Optics) by:

$$SFS = \frac{2 \times FOV}{SR} \tag{1}$$

where FOV= 200mm is the field of view, given by

$$FOV = \frac{SS \times WD}{FL} \tag{2}$$

Subtituting FOV from (2) in (1) yields:

$$SFS = \frac{2 \times SS \times WD}{SR \times FL} \tag{3}$$

Using the above numerical values in (3) gives SFS= 0.38mm, which is sufficiently small for detection of food particles in our project.

For inspection purposes, we desire an even illumination across the dish piece, as well as a minimum of specular reflections and shadows. The white mat-finished cardboard forming the inside surfaces of our light tent provided some diffuse lighting of our dish pieces and reduction of specular reflections, while the choice and placement of our lights reduced shadowing. Acceptable lighting was achieved by trial and error. After considerable experimentation, we selected as our light sources four 12V-20W DC halogen light bulbs surrounding the dish piece and placed as indicated in Figure 2, which provided sufficient illumination for both identification and inspection. These lamps were powered by a Switch Mode Power Supply (SMPS) fed by 110V 60Hz building supply. Camera exposure time was set at 20 µs, which was found by experiment to yeild the best dish image details.

2.2 Dish Set

The dish set used in this project is shown in Fig 3.

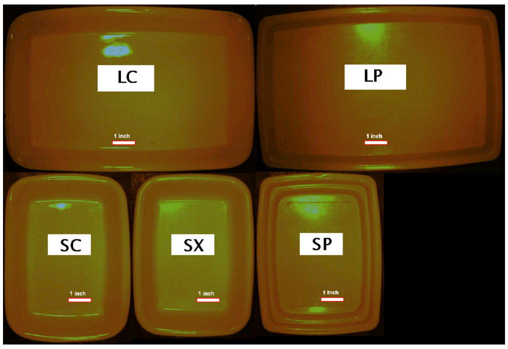


Figure 3: Dish Set Consisting of 5 types and 3 Colors of Dishes.

Our dish set was commercially available and used by a large, 700 bed hospital in Oklahoma. It consisted of 5 types and 3 colors of dishes. It was selected not only because it is in wide commercial use, but also because the colors, shape, and size of different types of dishes are very similar. However, each dish piece had uniform color, with no decorative markings (i.e. they were "plain"). Other commercially available "plain" dish sets present much lower challenges for the identification problem because their sizes are often easily distinguishable. For easy reference, we name each dish piece using size (large or small), and its material or function (ceramic, plastic, or spacer). Then LC and LP represent, respectively, the large ceramic dish and the large plastic dish, while SC, SP and SX represent, respectively, the small ceramic dish, the small plastic dish, and the small plastic spacer dish.

2.3 Pre-processing

We developed a pre-processing algorithm for thresholding, computing areas, and choosing the largest object (or particle) in a dish image, rejecting all other particles. This process removed noise and reliably retrieved a good dish image from the camera image, which was then ready for identification and inspection, illustrated in Figure 4.

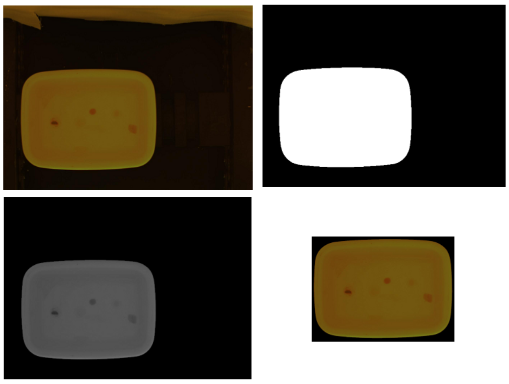


Figure 4: Example of SX Pre-processing. Original camera image (top-left); Binary mask for largest object (top-right); Gray image (bottom-left); and Dish retrieved image (bottom-right).

3 DISH IDENTIFICATION

The implemented automatic system should at least be comparable in performance to what can be achieved manually. Accordingly, from our observations in actual commercial dishwashing operations, our automatic system should be able to accurately recognize and inspect 5 types of dishes in real time at a minimum rate of 30 dishes per minute. While this task is easily accomplished manually, it poses a significant challenge for automation. We desire an algorithm that works flawlessly under varying dish positions beneath the camera and with varying illumination.

3.1 Possible Approaches

A human can quickly recognize each type of dish based on the weight, corner curvature, size, edge pattern, color, and/or a mix of these properties. In an attempt to imitate these capabilities, we experimented with several approaches based on edge detection, color recognition, and statistics of shape descriptors, such as area and perimeter. When using the area of a bounding box around the dish image axes, we found poor repeatability, due to slight dish non-alignment with the box (rotation), which produced pixel counting errors. Methods based on color or gray level intensity were found to be overly sensitive to small lighting variations. Edge detection methods, using Matlab and its Image Processing Toolbox, were not only computationally expensive and slow, but also proved difficult in selecting appropriate threshold values (Duong, 2008).

3.2 Identification Algorithm

We propose using statistics of shape descriptors of dish pieces to solve the identification problem. Three easily calculated statistics of shape descriptors are the dish image area, the ratio of dish image length to width, and the ratio of the dish image area to the area of the image oriented bounding box. The area of the dish image is already available from the pre-processing step.

In order to classify dish types, we used a training set of 500 images, with 100 images for each dish type, in varying position and orientation under the camera, to estimate the distributions of dish image properties. Examining the dish image area, we observe from Figures 5 and 6 that two groupings readily appear: large areas representing LC and LP, and small areas representing SC, SP and SX. By considering only dish area information, Table 1 shows that it is straightforward to identify SC, since none of the image areas of SC overlap with any other dish piece. However, there clearly is overlap in image areas of LP and LC, and of SP and SX.

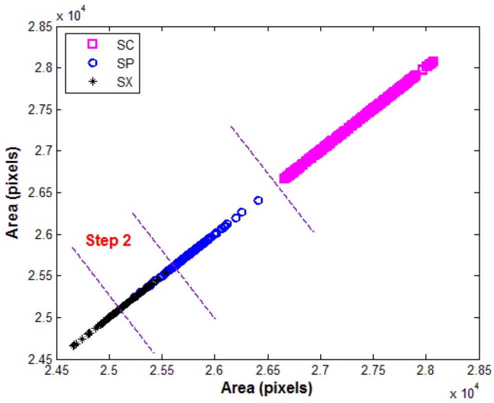


Figure 5: Small Dish Image Area Group.

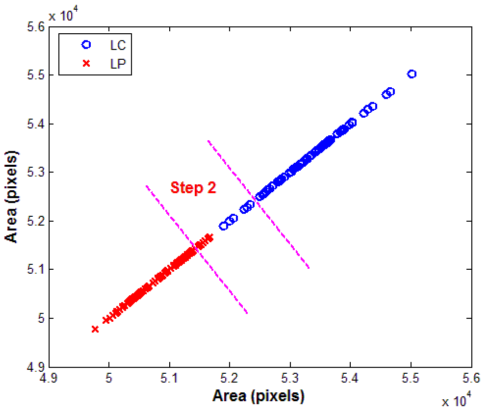


Figure 6: Large Dish Image Area Group.

Table 1: The area contribution.

Dish type	SX	SP	SC	LP	LC
Area (10^4 pixels)	2.45- 2.58	2.53- 2.66	2.69- 2.83	4.96- 5.28	5.16- 5.51
Overlap region	2.53- 2.58 **SX or SP**?		x	5.16- 5.28 **LP or LC**?	

Accordingly, we employ other properties to distinguish between them. Consider the ratio of dish image area to the area of the oriented image bounding box, called O_EXT, which can be thought of as indicating the curvature of the dish corner. This ratio is much faster to calculate, with more repeatable results, than calculating the radius of curvature of the dish corner, as attempted by Johnson (1993). Consider further the ratio of bounding box length to width, called O_REC, which can be calculated quickly with repeatable results. Using these two properties together, as indicated in Figure 7, we can easily distinguish SP from SX. Using area and length-to-width ratio, Figure 8 illustrates how LC is easily distinguished from LP.

The optimal lines to separate SP and SX, and LC and LP are given in Table 2. To save time, the two

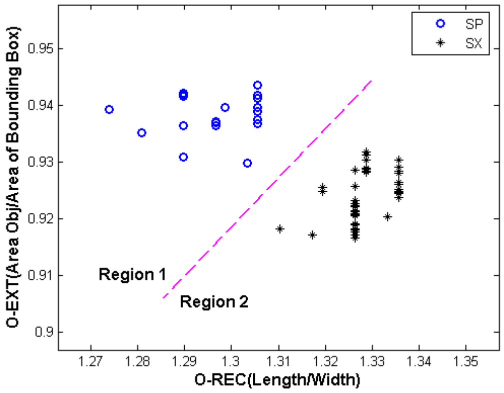


Figure 7: Separating SP and SX Dish Types.

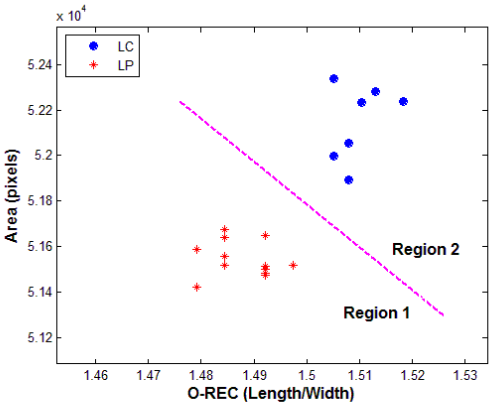


Figure 8: Separating LC and LP Dish Types.

properties, O_REC and O_EXT, are computed only if the area property is insufficient to make a reliable decision. Otherwise, the algorithm stops at Step 1 below.

Table 2: Lines separating SP and SX, LC and LP.

	Line to separate SP and SX		Line to separate LC and LP	
	Point 1	Point 2	Point 1	Point 2
Area (10^4 pixels)	x	x	5.22	5.13
O_REC	1.28	1.35	1.48	1.52
O_EXT	0.91	0.95	x	x

Hence, our identification algorithm is as follows:

- Pre-processing: retrieve dish image as the largest object in camera image.
- Step 1: Classify using dish area.
- Step 2: Separate SP and SX using O_REC and O_EXT. Separate LC and LP using O_REC and area.

3.3 Identification Results

Results were collected from 725 images of all types of dish pieces, not including any of the 500 training set images. All training and testing image sets were produced from 84 dishes of all types, clean and dirty, under different lighting conditions (produced by changing the exposure time of the camera) and under different dish positions and orientations under the camera axis.

The results in Table 3 show accurate identification for all images, with an average computation time of 0.21 sec. This is deemed acceptable to allow identification and inspection of dishes at our target dish processing rate of 30 dish pieces per minute. The variability from min to max computation time is explained because the amount of rotation among dish pieces varied with dish position, causing variability in times to compute classification parameters.

Table 3: Results of Dish Identification.

	No.	Correct	Time* (sec)		
			Min	Average	Max
LC	85	100%	0.18	0.22	0.57
LP	120	100%	0.18	0.33	0.59
SC	200	100%	0.17	0.18	0.23
SP	167	100%	0.17	0.20	0.49
SX	153	100%	0.16	1.24	0.48
All dishes	725	100%	0.16	0.21	0.59

(*) Matlab® R14, Image Processing Toolbox V5.0, Window Vista, dual core 1.6GHz, 2GB RAM.

4 DISH INSPECTION

Automated dish inspection following commercial dishwashing using image processing presents some unique challenges. First, the intensity of dish images is sensitive to changes in lighting, normal power fluctuations, and camera sensitivity drift (Lolla, 2005). Second, even with reasonable attempts to establish uniform illumination of dish pieces, uneven illumination persists in the camera field of view. This non-uniform color and gray intensity across a clean dish varies as the position of the dish varies in the field of view. Third, because of the non-flat geometry of the dish surface, the gray intensity of the image drops significantly at the dish side wall, especially for a deep dish with steep sidewalls, such as LC, SC and SX. Moreover, glare and shadows increase the difficulty of discerning clean from dirty dishes, even for human manual inspection. Fourth, food particle images vary in gray level, depending on food type, size, and location. Certain food particles, such as dried egg yolk, can be especially difficult to detect. Fifth, the definitions of a "clean dish" and a "dirty dish" are subjective and ill-defined (Zhou, 2008).

4.1 Previous Work

Zhou (2008) proposed a fusion based technique for silverware inspection. His key idea was based on the observation that shadows will move, but dirt will not, between two images of a silverware piece captured at two different positions under fixed illumination. Zhou's technique combines relevant information from two images, which reduces noise and recovers information in regions obscured by lighting glare and shadows. His method could be used in pre-processing before inspection, as long as the computation time is sufficiently small. After fusion of two images of one piece, Zhou applied simple global thresholding to the three color (R, G and B) channels. While this approach worked well for silverware, it will not work for dish inspection, because the gray level of a clean spot on the dish wall is comparable to or less than a dirty spot on the dish floor, and Zhou used only global thresholding.

Lolla (2005) used template edge matching for silverware inspection. This approach is not only time consuming, but also suffers from lacking the ability to deal effectively with glare and shadows.

One approach we considered was to create targeted illumination on the dish walls based on their inclination angle, and then apply global thresholding to the entire modified image. The problem was that modifying illumination of the wall was time consuming, and was difficult to adapt to inside corners. A more promising approach was to treat the dish floor region and dish wall region differently during image processing. This is the essence of our proposed method, in which we use partitioning and adaptive thresholding, which is much simpler and more efficient.

4.2 Inspection Algorithm

Our inspection method is inspired by observing how human eyes inspect a dish piece. Humans focus locally, and are very sensitive to relative changing in color or intensity. The human visual system also easily learns, and appears to eliminate glare and shadows, as well as adapts the background intensity.

In our inspection algorithm, a “dirty spot” is defined as a connected component that is (1) sufficiently dark, and (2) darker than the immediately surrounding area. The algorithm first separates the dish image into a dish floor region and a dish wall region. Then for each region, we automatically locate all components that are darker than the near surrounding area, employing adaptive thresholding, and discard those components that are not sufficiently dark, employing global thresholding. After these steps, we combine the floor and wall regions and carry out post-processing to remove small noise.

The steps of our inspection algorithm are as follows:

- Identify the dish piece, so that we have a template for partitioning.
- Start from a gray image of a dish piece which is the result of pre-processing.
- Partitioning: Detect and separate floor region and wall region of a dish image by using the appropriate floor template image.
- Adaptive and global thresholding: work with the floor region and wall region separately. Adaptive finds potential spots; global retains only those with gray levels less than the global threshold.
- Combine the two regions.
- Post processing: remove tinny spots that most likely are produced by noise rather than dirt.

Our experimental results showed that it is unnecessary to use a color image, such that even though we took color images with our camera, we employed only the corresponding gray scale image. A sample result of the partitioning process based on a dish floor template is shown in Figure 9.

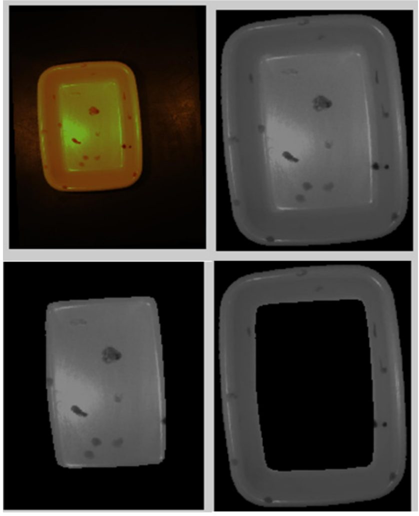


Figure 9: Partitioning process (SC Dish): Original Camera Image (top-left), Gray Scale Pre-processed Dish Image (top-right), Dish Floor Region (bottom-left), and Dish Wall Region (bottom-right).

For the dish in Figure 9, results of adaptive thresholding combined with global thresholding for floor region and wall region are shown in Figure 10.

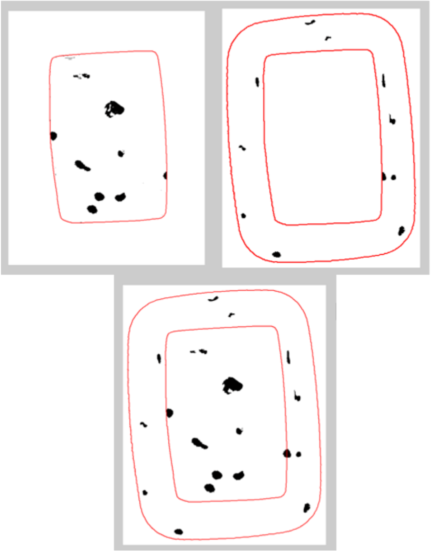


Figure 10: Adaptive and Global Thresholding for Floor Region (top-left) and Wall Region (top-right). Combination of the Two Regions and Post-processing (bottom).

For all tested images in our inspection process, the values of parameters for adaptive thresholding and global thresholding, found by trial and error, are given in Tables 4 and 5 respectively.

Table 4: Adaptive Thresholding Parameters.

	Dish floor region		Dish wall region	
	Window size (pixels)	Threshold value (%)	Window size (pixels)	Threshold value (%)
SC	60	3	30	5
SX	50	5	30	3
SP	50	5	12	3

Table 5: Global thresholding parameters.

	Dish floor region (gray level 0-255)		Dish wall region (gray level 0-255)	
	Lower bound	Upper bound	Lower bound	Upper bound
SC	20	86	20	70
SX	20	99	20	66
SP	20	99	20	54

4.3 Inspection Results

To obtain experimental results with our proposed inspection method, we manually applied real food particles to a variety of our dish pieces, varying them in type size, shape, and location of food. We used egg yolk, fruit juice, and a variety of sauces, including tomato-based sauces, all of which were allowed to air dry before inspection.

Figure 11 shows an example of results, with the original SC dirty dish image on the left and detected dirt spot boundaries (on the right) overlaid on the original dish image. All dirt spots with variation of gray intensity and location were correctly detected.

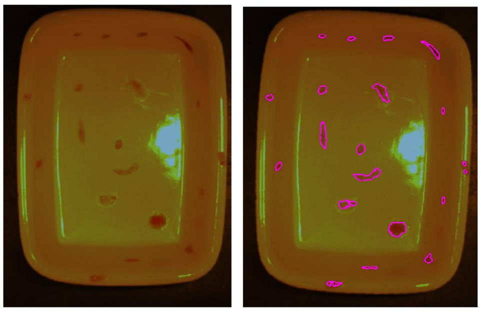


Figure 11: Origin SC (left) and all dirt spots detected (right).

Notice that the glare (specular reflection) due to the shiny surface of the ceramic dish did not produce spurious results.

Inspection results of a dirty SX dish and a clean SX dish are presented in Figures 12 and 13, respectively. Notice in Figure 12 the dirt spot exactly on the boundary of the floor region and wall region of the dish image (near the centre of the left image). This location of a dirty spot could be a significant detection challenge, yet our algorithm did detect it.

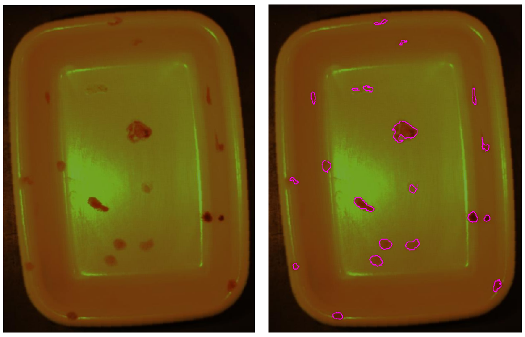


Figure 12: Original SX (left) and All Dirt Spots Detected (right).

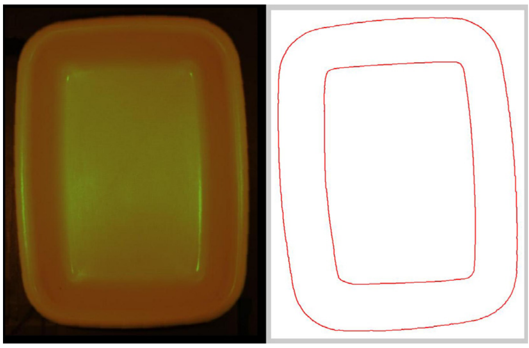


Figure 13: Origin Clean SX (left) and Correctly Detected Non-dirt (right).

One inspection result of a dirty SX dish is shown in Figure 14. All dirt spots, even some that are right on the inner edges of multilayers of the dish, are correctly detected. This type of dish with 3 shallow layers was the most difficult case to tune the parameters of the inspection algorithm.

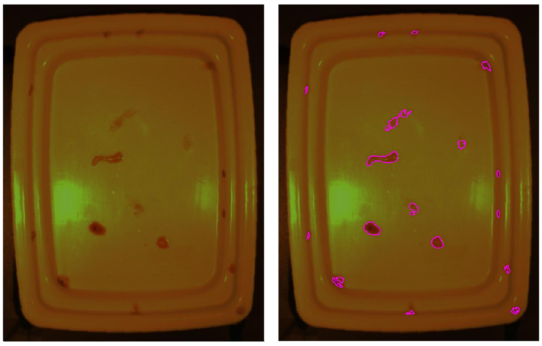


Figure 14: Original SP (left) and All Dirt Spots Detected (right).

A summary of inspection results for all types of dish pieces, SC, SP, SX, LC, and LP is given in Table 6.

Table 6: Summary of Inspection Results for Small Dishes.

	Total	Correct	Miss	Failed Alarm
Clean SC	5	4	-	1
Dirty SC	5	5	0	-
Clean SP	5	4	-	1
Dirty SP	5	5	0	-
Clean SX	5	5	-	0
Dirty SX	5	5	0	-
Clean LC	5	5	-	0
Dirty LC	5	5	0	-
Clean LP	5	4	-	1
Dirty LP	5	5	0	-
All dishes	50	47 (94%)	0 (0%)	3 (6%)

"Failed Alarm" in Table 6 means an incorrect result that a clean dish is classified as dirty. Such results would reduce production in a real inspection process line because the clean dish will be sent back to be re-washed when this was not necessary. On the other hand, such a result is deemed superior to a dirty dish that is classified clean, which would be unacceptable for hygienic and other reasons.

For the results in Table 6, the average computation time for the inspection process was 1.28 second per dish (using Matlab® R14, Image Processing Toolbox V5.0, Window Vista, dual core 1.6GHz, 2GB RAM). If we add this to the average identification time reported earlier, the total average time required for both identification and inspection processes for the small area dish group is approximately 1.47 second, which is acceptable for our target of processing 30 dish pieces per minute (2 seconds per piece).

5 CONCLUSIONS

In this study we successfully implemented new and novel dishware identification and inspection algorithms. The experimental results show that these algorithms work well under variation of lighting conditions caused by variation of dish position under the camera axis. The algorithms implemented on a standardly available PC were sufficiently fast for real time processing at a minimum rate of 30 dish pieces per minute. We also experimented with other dish sets, including plain circular and oval shaped plates, and small bowls. With small changes in few parameters, the algorithms work equally well.

On the other hand, partitioning and adaptive thresholding combined with global thresholding as presented here will not work for dishes that have colored or molded patterns on the surface. However, because most dish sets used in large scale dinning operations are non-textured and mono-colored with uniform background, our procedure should be widely applicable.

ACKNOWLEDGEMENTS

This work has been supported by a scholarship from Ministry of Education and Training of Vietnam (MOET- www.moet.gov.vn) and a research assistantship from Oklahoma State University.

REFERENCES

Bernsen, J., 1986. Dynamic thresholding of gray level images. In *Proc. Intl. Conf. on Pattern Recognition*, pp. 1251–1255.

Duong, H., T., Hoberock, L., L., 2008. Dish Recognition Based on Statistics of Shape Descriptors. *XXVIII Oklahoma AIAA/ASME Symposium, Tulsa, Oklahoma.*

Edmund Industrial Optics. *Fundamental Parameters of an imaging system.* http://www.edmundoptics.com/techsupport/DisplayArticle.cfm?articleid=287

Fisher, R., Perkins, S., 2003. *Adaptive Thresholding.* http://homepages.inf.ed.ac.uk/rbf/HIPR2/adpthrsh.htm

Johnson, K., A., 1993. *Machine Vision Sorting and Inspection in Commercial Automatic Dishwashing.* Oklahoma State University M.S. Thesis.

Lolla, V., 2005. *Identification and Inspection of Silverware Using Machine Vision.* Oklahoma State University M.S. Thesis.

Yeri, S., 2002. *Classification of Silverware Pieces Using Machine Vision.* Oklahoma State University Thesis.

Zhou, Y., and Hoberock, L. L., 2008. *Flaw Detection Using Image Registration and Fusion Techniques*, Proceedings of the ASME 2008 International Design Engineering Technical Conference & Computers and Information in Engineering Conference, Paper DETC2008-49703, August 3 – 6, 2008, Brooklyn, NY.

MODEL AND APPROACHES FOR TENSION OF PARALLEL STRUCTURES WITH ELASTIC JOINTS FOR MICRO AND NANO MANIPULATORS

D. Chakarov, K. Kostadinov and T. Tiankov

Institute of Mechanics, Bulgarian Academy of Sciences, Acad. G. Bonchev Str., bl. 4, 1113, Sofia, Bulgaria
{mit, kostadinov, tihomir}@imbm.bas.bg

Keywords: Piezo-Actuators, Serial-Parallel Micromanipulator, Elastic Joints, Stiffness Model, Preliminary Tension, Cell Injection.

Abstract: In this paper piezo actuated micromanipulators are considered with serial-parallel structure including elastic joints. Such structure allows a preliminary tension of the mechanical system in order to eliminate backlashes and to improve the performance of the piezo-actuators. A kinematics model of a serial-parallel structure for local micro manipulators is build here. A pseudo rigid body approach is used, where elastic joints are modelled as revolute joints. A stiffness model is created to estimate the general stiffness of the manipulator by means of reduction the stiffness of all elastic joints. Two approaches are presented here for preliminary tension of parallel manipulator structure: - deflection from the initial manipulator state by introducing of a driving joints motion during the assembly; - preliminary tensioning of the separate elastic joints. The two approaches considered are experimented on the manipulator for cell injection. The values of the mechanical parameters obtained by preliminary tension of the manipulator are pointed out.

1 INTRODUCTION

Micro and nano manipulators are mostly used in biological and microelectronics research, cellular technology, chemistry and investigation of thin films, in atomic force microscopes and scanning tunnelling microscopes.

There are known micromanipulators with piezo actuators (Fatikow, 1996; Kasper, 1998). Piezo actuated micromanipulators with parallel structure are also developed (Lee, 1999). Robots with parallel structure possess many advantages. Their small workspace in the case of cell manipulations is not a disadvantage, since it is enough for the application considered. Mechanisms with closed kinematic chains (Ionescu, 2002; Guergov, 2005, Prusak, 2009) are suitable for high-precision tasks in 3D space. The high accuracy of such mechanical systems is due to very high structural stiffness.

From the other hand in order piezo-ceramic structures to be with high stiffness they can be realized by parallel or a closed structure which has to be tensed. It is possible to use deformation in elastic joints or antagonistic redundant actuators to achieve tension in closed piezo-ceramic structures with desired degree of freedom (DoF).

To predict the displacements of compliant mechanisms with elastic joints the pseudo-rigid-body-model is commonly used (Zhang, 2002). As a rule it models an elastic joint as a revolute joint with a torsion spring attached. The pseudo-rigid-body method is effective and it simplifies the model of compliant mechanisms. To estimate the mechanism stiffness with elastic joints an analytical model is created out taking into account compliances of elastic joints in all axes. The analytical model is describing the relationship between input and output displacements of the mechanism, (Pham, 2005) or computing the stiffness matrix and estimating the stiffness performances of the robot (Carbone, 2006). To increase accuracy of the stiffness matrix identification alternative methodology is developed using advantages of analytical and numerical techniques (Pashkevich, 2009). Such analytical stiffness models of serial-paralel manipulators with elastic joints are analyzed in order to synthesize desired stiffness of the robot end-effector (Chakarov, 2004).

The objective of this paper is to create a stiffness model and to develop approaches for tension of serial-parallel structures with elastic joints for micro and nano manipulators with application in cellular technology, microelectronics, chemistry etc.

2 KINEMATIC MODEL OF PARALLEL STRUCTURES FOR LOCAL MICRO AND NANO MANIPULATORS

Investigated structures are serial-parallel structures including basic link 0 and some other links 1, … , n connected in between in a serial chain. The end-effector M is situated in the end link n of this chain, which moves in a ν operation space. The driving chains A_1 ,…, A_m, with number m, are attached to the basic link 0 and to the end link n forming parallel chains [Chakarov, 2007] as it is shown in Fig.1. The type of the kinematics joints is not shown in Fig.1, as they can be elastically or kinematically ones.

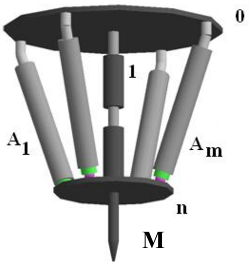


Figure 1: Generalized kinematic scheme of a serial-parallel manipulator.

All joints are modelled as kinematic joints with different number of restrictions, which give 6 DoF for each drive chain. In this way the number DoF of the structure is defined by the number DoF of the serial chain h. Let generalized parameters are accepted to be the parameters of the relative motions in all joints - elastic and non-elastic of the structure, presented by (k x 1) vector

$$\boldsymbol{\theta = [q;q^1]^T} \qquad (1)$$

Where

$$\mathbf{q = [q_1, \ldots., q_h]^T} \qquad (2)$$

is an (h x 1) vector of the generalized coordinates in the joints of the main serial chain with h DoF and

$$\mathbf{q^1 = [w; l]^T} \qquad (3)$$

is a (6m x 1) vector of coordinates in the joints of the actuator chains with number **m.**

Above

$$\mathbf{w} = [\mathbf{w}_1, \ldots, \mathbf{w}_{5m}]^T, \qquad (4)$$

is an (5m x1) vector of coordinates in the passive joints of the actuator chains, and

$$\mathbf{l = [l_1, \ldots, l_m]^T}, \qquad (5)$$

is an (m x 1) vector of coordinates in the motor linear joints of the actuator chains.

Let the Cartesian coordinates of the end effector M are denoted as

$$\mathbf{X = \left[X_1, \ldots, X_\nu\right]^T, \nu \leq 6} \qquad (6)$$

The relation between the parameters of the basic serial chain (2) and the parameters of the end effector (6) is known as a direct problem of the kinematics of the serial chain. This problem on the level of displacements and velocities is presented by the equations $\mathbf{X = \Psi(q)}$ and

$$\mathbf{\dot{X} = J\dot{q}} \qquad (7)$$

where $\mathbf{J} = \left[\partial\mathbf{X}/\partial\mathbf{q}\right]$ is the (ν x h) matrix of Jacoby.

In the parallel structure each closed loop implies the appearance of a connection between the generalized parameters (1). These connections are expressed by 6m scalar functions for the structure including m parallel loops: $\mathbf{\Psi_i(\theta) = 0}, i = 1,\ldots,6m$. The differentiation of above equations gives the relation

$$\mathbf{H_q \frac{d\,q}{d\,t} + H_w \frac{\partial w}{\partial q}\frac{\partial q}{\partial t} + H_l \frac{\partial l}{\partial q}\frac{\partial q}{\partial t} = 0} \qquad (8)$$

The matrix of partial derivations $\mathbf{H_q}$, $\mathbf{H_w}$ and $\mathbf{H_l}$ with size (6m x h), (6m x 5m) and (6m x m) allows to produce the summarized matrix of the partial derivatives

$$\mathbf{D} = \left|\frac{\partial \mathbf{q}}{\partial \mathbf{q}}; \frac{\partial \mathbf{w}}{\partial \mathbf{q}}; \frac{\partial \mathbf{l}}{\partial \mathbf{q}}\right|^{\mathbf{T}} = \left|\mathbf{E; W; L}\right|^{\mathbf{T}} \qquad (9)$$

where **E** is unitary (h x h) matrix, **W** is a (5m x h) matrix and **L** is a (m x h) matrix, or $\mathbf{D} = \left|\mathbf{E; H}\right|^{\mathbf{T}}$.

According to (8) we can reduce the (6m x h) matrix

$$\mathbf{H} = \left|\mathbf{W; L}\right|^{\mathbf{T}} = \frac{\partial \mathbf{q^1}}{\partial \mathbf{q}} = -\left|\mathbf{H_w; H_l}\right|^{-1} \mathbf{H_q} \qquad (10)$$

where $\left|\mathbf{H_w; H_l}\right|$ is a (6m x 6m) invertible matrix.

Using matrix (10) we have the relations between generalized velocities:

$$\dot{\mathbf{q}}_\mathbf{l} = \mathbf{H}\dot{\mathbf{q}} \tag{11}$$

$$\dot{\mathbf{w}} = \mathbf{W}\dot{\mathbf{q}} \tag{12}$$

$$\dot{\mathbf{l}} = \mathbf{L}\dot{\mathbf{q}} \tag{13}$$

The above equations allow determining the velocities $\dot{\mathbf{q}}_\mathbf{l}$ with dimension 6m as a function of the generalized velocities $\dot{\mathbf{q}}$ with dimension equal to the DoF h of the structure.

When the number of parameters (5) is equal to the DoF m = h, these parameters can be selected as independent parameters. In relations (13) **L** is a **(h x h)** matrix and inverse relation is possible:

$$\dot{\mathbf{q}} = \mathbf{L}^{-1}\dot{\mathbf{l}} \tag{14}$$

Equations (7) and (14) allow determining the velocities of end-effector, while equations (11) and (14) - the velocities of passive joints, as function of velocities of linear actuator joins $\dot{\mathbf{l}}$:

$$\dot{\mathbf{X}} = \mathbf{J}\mathbf{L}^{-1}\dot{\mathbf{l}} \tag{15}$$

and

$$\dot{\mathbf{q}}_\mathbf{l} = \mathbf{H}\mathbf{L}^{-1}\dot{\mathbf{l}} \tag{16}$$

By micromanipulations the above equations give the relations between small motions of microactuators Δl, small motions of the end-effector ΔX and small motions in passive joints $\Delta\mathbf{q}_\mathbf{l}$: $\Delta\mathbf{X} = \mathbf{J}\mathbf{L}^{-1}\Delta\mathbf{l}$ and $\Delta\mathbf{q}_\mathbf{l} = \mathbf{H}\mathbf{L}^{-1}\Delta\mathbf{l}$.

3 STIFFNESS MODEL OF SERIAL – PARALLEL STRUCTURES FOR MICRO AND NANO MANIPULATORS

Denote by $\mathbf{P} = [\mathbf{P}_1, \dots, \mathbf{P}\nu]^\mathbf{T}$ the (ν x 1) vector of the external forces and torques applied to the end-effector, corresponding to Cartesian coordinates (6). Denote by $\mathbf{Q} = [\mathbf{Q}_1,\dots, \mathbf{Q}_\mathbf{h}]^\mathbf{T}$ the (h x 1) vector of the generalized forces and torques in the joints of the main chain corresponding to the general coordinates (2). According to the principle of virtual work and equation (7), the connection between forces **P** and **Q** is as follows:

$$\mathbf{Q} = \mathbf{J}^\mathbf{T}\,\mathbf{P} \tag{17}$$

Denote by $\mathbf{F}_\mathbf{q} = \left|\mathbf{F}_{\mathbf{q1}};\dots;\mathbf{F}_{\mathbf{qh}}\right|^\mathbf{T}$ and $\mathbf{F}_\mathbf{w} = \left|\mathbf{F}_{\mathbf{w1}};\dots;\mathbf{F}_{\mathbf{w5m}}\right|^\mathbf{T}$ (h x 1) and (5m x 1) vectors of the forces and torques in the elastic joints, corresponding to coordinates (2) and (4). Denote by $\mathbf{F}_\mathbf{l} = \left|\mathbf{F}_{\mathbf{l}_1};\dots;\mathbf{F}_{\mathbf{l}_\mathbf{m}}\right|^\mathbf{T}$ the (m x 1) vector of the driving forces in the linear joints correspond to the coordinates (5). Above vectors can be summarized in the (h + 6m) x 1 vector of forces and torques, corresponding to the coordinates (1) **F**= [$\mathbf{F}_\mathrm{q}$; $\mathbf{F}_\mathrm{w}$; $\mathbf{F}_\mathrm{l}$]$^\mathrm{T}$.

According to the principle of virtual work and the equation (12), (13) the relation between forces **F** and generalized forces **Q**, using summarized matrix (9), is as follows:

$$\mathbf{Q} = \mathbf{D}^\mathbf{T}\,\mathbf{F} \tag{18}$$

$$\mathbf{Q} = \mathbf{F}_\mathbf{q} + \mathbf{W}^\mathbf{T}\,\mathbf{F}_\mathbf{w} + \mathbf{L}^\mathbf{T}\,\mathbf{F}_\mathbf{l} \tag{19}$$

Equations (17) and (19) produce

$$\mathbf{J}^\mathbf{T}\,\mathbf{P} = \mathbf{F}_\mathbf{q} + \mathbf{W}^\mathbf{T}\,\mathbf{F}_\mathbf{w} + \mathbf{L}^\mathbf{T}\,\mathbf{F}_\mathbf{l} \tag{20}$$

Differentiation of above equation with respect to parameters (2) and neglect the second partial derivatives, gives

$$\mathbf{J}^\mathbf{T}\,\frac{\partial\mathbf{P}}{\partial\mathbf{X}}\,\mathbf{J} = \frac{\partial\mathbf{F}_\mathbf{q}}{\partial\mathbf{q}} + \mathbf{W}^\mathbf{T}\,\frac{\partial\mathbf{F}_\mathbf{w}}{\partial\mathbf{w}}\,\mathbf{W} + \mathbf{L}^\mathbf{T}\,\frac{\partial\mathbf{F}_\mathbf{l}}{\partial\mathbf{l}}\,\mathbf{L} \tag{21}$$

Considering micromanipulator structure as a system with concentrated compliance in the joints [Chakarov, 2004] gives

$$\mathbf{K} = \mathbf{J}^{-\mathbf{T}}[\mathbf{K}_\mathbf{q} + \mathbf{W}^\mathbf{T}\mathbf{K}_\mathbf{w}\mathbf{W} + \mathbf{L}^\mathbf{T}\mathbf{K}_\mathbf{l}\mathbf{L}]\mathbf{J}^{-1} \tag{22}$$

where $\mathbf{K} = \partial\mathbf{P}/\partial\mathbf{X}$ is (ν x ν) matrix of the Cartesian stiffness of the end effector; $\mathbf{K}_\mathbf{q} = \partial\mathbf{F}_\mathbf{q}/\partial\mathbf{q}$ is diagonal (h x h) matrix of the shaft stiffness in the joints of the main serial chain; $\mathbf{K}_\mathbf{w} = \partial\mathbf{F}_\mathbf{w}/\partial\mathbf{w}$ is diagonal (5m x 5m) matrix of the shaft stiffness in the passive joints of the driving chains; $\mathbf{K}_\mathbf{l} = \partial\mathbf{F}_\mathbf{l}/\partial\mathbf{l}$ is diagonal (m x m) matrix of the shaft stiffness in the driving joints.

4 APPROACHES FOR PRELIMINARY TENSIONING OF PARALLEL MICRO MANIPULATORS WITH ELASTIC JOINTS

A preliminary tensioning of the mechanical micromanipulation system is necessary in order to eliminate the backlash and to improve the performance of the piezo-actuators. When only joints of class five are used for the modeling the mechanical system, the number of all joints is equal to the number k of the generalized system coordinates. In a case when number m of the driving joints is equal to the number of DoF h (m=h), then number of all the remaining joins is (k - h).

The following two approaches can be used for tensioning of the manipulator:

- deflection from the initial manipulator state by m = h driving joints motion introduced in the assembly;
- preliminary tensioning of the separate elastic joints with number j, $(k\text{-}h \geq j \geq h)$.

4.1 Tensioning by Means of Deflection from the Initial State

This can be achieved by means of an assembly deflection **δl** in the driving joints, which leads to deflection in all the system joints according to (14), (12) and deflection of the end-effector according to (7) defined by the equations:

$$\mathbf{\delta q = L^{-1}\delta l} \tag{23}$$

$$\mathbf{\delta w = W\delta q = WL^{-1}\delta l} \tag{24}$$

$$\mathbf{\delta X = J\delta q = JL^{-1}\delta l} \tag{25}$$

These deflections lead to elastic joints forces defined by the equations:

$$\mathbf{F_q = k_q\delta q = k_q L^{-1}\delta l} \tag{26}$$

$$\mathbf{F_w = k_w\delta w = k_w WL^{-1}\delta l} \tag{27}$$

where $\mathbf{k_q}$ and $\mathbf{k_w}$ are stiffness matrices of the passive joints of the basic serial chain and of the driving chains, respectively. The tensioned elastic system according to (19) is in a static equilibrium:

$$\mathbf{Q = F_q + W^T F_w + L^T F_l = 0} \tag{28}$$

The diagonal matrices $\mathbf{k_q}$ and $\mathbf{k_w}$ contain non-zero components, responding to elastic joints and zero components responding to kinematic joints. The number of non-zero components j must be bigger or equal to the DoF $j \geq h$ in order to achieve full degree of tension of all the actuators and limbs within the system. Equation (28) allows definition of the forces of the driving joints **F** in number h as a function of the forces $\mathbf{F_q}$, $\mathbf{F_w}$ in number $j \geq h$

$$\mathbf{F_l = -L^{-T} [F_q + W^T F_w]} \ . \tag{29}$$

4.2 Tensioning by Deformations in the Elastic Joints

In the manipulator structure with m driving joints there are k - m passive joints, which can be elastic. Because the driving joints m = h, by means of which the piezo-actuators are modelled are hundreds of times more rigid then the elastic manipulator joints, it is accepted that the system has 0 DoF. The tensioning of the elastic joints does not lead to a change in the manipulator position, but only in a change of the internal forces. For the actuator tensioning, the number of the elastic joints j must be bigger than the number of the DoF $j \geq h$. The preliminary joint deformations can be defined by the vectors:

$$\mathbf{\delta q^0 = [\delta q^0{}_1, \ldots., \delta q^0{}_h]^T} \tag{30}$$

$$\mathbf{\delta w^0} = [\mathbf{\delta w^0}_1,\ldots, \mathbf{\delta w^0}_{5m}]^T, \tag{31}$$

where the components of which corresponding to non-elastic (kinematic) joints are equal to 0**.**

The joint stiffnesses are represented by the diagonal matrices $\mathbf{k_q}$ and $\mathbf{k_w}$, which contain non-zero components corresponding to the elastic joints and zero components connected to the kinematic joints.

The preliminary deflections lead to appearance of forces in the elastic joints defined by equalities:

$$\mathbf{F_q = k_q \delta q^0} \tag{32}$$

$$\mathbf{F_w = k_w \delta w^0} \tag{33}$$

The driving elastic joints forces are in a static equilibrium (19):

$$\mathbf{Q = F_q + W^T F_w + L^T F_l = 0} \tag{34}$$

The upper equation defines the links among all the joint forces and allows the derivation of the driving joints forces in number h as a function of the elastic forces in number $j \geq h$

$$\mathbf{F_l = -L^{-T} [F_q + W^T F_w]} \ . \tag{35}$$

The forces in the driving joints can lead to small deflections in those joints due to the piezo-actuators high stiffness. Those deflections by the two approaches can be defined by the equality:

$$\mathbf{\delta l^0 = \frac{F_l}{k_l}} \tag{36}$$

where the diagonal matrix $\mathbf{k_l}$ includes the stiffness of the piezo- actuators, and $\mathbf{\delta l^0}$ are the deflections in the driving joints after the system tensioning. Resulting deflection of the end effector can be received according to equation (15).

5 NUMERIC EXPERIMENTATION OF TENSIONING APPROACHES IN ROBOTS FOR CELL INJECTION

A robot has been designed to perform automatic cell injection. The cells in the range of 10-30 [µm] are preliminary positioned in a matrix G. Local robot structure has serial-parallel structure with 3 DoF as shown in Fig.2. Base 0, elastic joint J_3, manipulator body 1, actuator $A_{3,}$ and working tool 2 with end-effector M form a serial chain. Actuators A_1 and A_2 are located perpendicularly to the manipulator body 1, and they are linked with the base 0 by means of elastic joints J_1 and J_2, thus forming parallel chains. The actuators are fixed to the body 1 via universal joints. Parallel structure comprising actuators A_1 and A_2 perform orientation motions, while the actuator A_3 performs injection through the pipette 2 attached to it. The main dimensions of the manipulator are $a_1=a_2=0.073$[m], $b_1=0.030$[m], $b_2=0.180$[m] as shown in Fig.2. Piezoactuators and elastic joints used have parameters specified in Table 1.

Table 1: Parameters of the elements used.

Elements	Travel [µm]	Resolu-tion [nm]	Axial stiffness [N/µm]	Reduced Axial stiffness [N/µm]	Angular stiffness [N/rad]
$A_{1,}A_2$	30	0.6	27	23.79	
A_3	60	1.2	15	13.95	
J_2,J_2,J_3			200		40

The end effector stiffness and the characteristics of the preliminary tensing of the parallel structure can be found. Since the manipulator under consideration is assembled with a special rectangle configuration we can easily derive the scalar equalities for the characteristic components using the matrix equalities (22), (23), (24), (25), (29). Software application based on these matrix equalities is developed using Microsoft Visual Studio.Net Express Edition and C++.

To find an estimation of a stiffness component along axis X, the software application substitutes the respective matrices in equality (22) for K_w=40[Nm/rad], J=b_2, K_l=23.788[N/m], W=b_1/a_2, K_q=40[Nm/rad], L=b_1 Performing the respective calculations, we find for the three additives in (22) that K_x=1235+209+660778=0.662 10^6 [N/m].

Similarly, the rest of the end-effector stiffness components can be found: K_y= 0.662 10^6[N/m], K_z=13.95[N/m]. As seen, the influence of piezo actuators to the joint stiffness is hundred times larger than the rest of the elements.

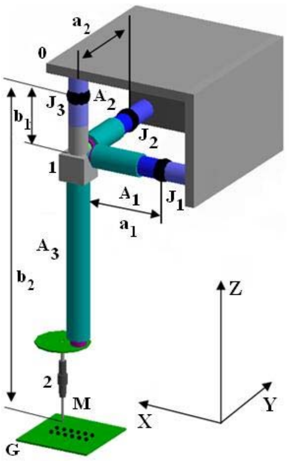


Figure 2: Microrobot for cell injection.

To apply preliminary tension by the actuator, following the approach outlined in Sub-paragraph 3.1., actuators A_1 and A_2 in the parallel structure should deflect by δl_1 and δl_2, so that the elastic joint deflection J_3 should not exceed +/-0.5^0, which is the admissible arbitrary rotation angle. Thus, $\delta q_1=\delta q_2=0.008726$[rad].

Using scalar equalities corresponding to (23), (24), (25), (29) as outlined above, we find the components of the elastic joint deflections and those of the actuator tension forces – see Table 2.

To attain preliminary tension in the manipulator as outlined in Sub-paragraph 3.2, the deflections of the three elastic joints J_1, J_2 and J_3 should be less than or equal to the admissible angles of rotation $\delta w^0{}_{11}$= =$\delta w^0{}_{21}$=0.008726[rad], $\delta q^0{}_1=\delta q^0{}_2$=0.008726 [rad].

Considering equality (35) and its scalar forms, we find the tension forces of actuators A_1 and A_2, Fl_1 = Fl_2 = -16.42[N]. These forces are larger than the tension forces found by applying the first approach, due to the tension of joints J_1, J_2 that is equal to the admissible limit. Both cases of tension yield small actuator deformations which can be found using equality (36).

Thus, considering the first case, those deformations are $\delta l^0{}_1$ = $\delta l^0{}_2$ =-0.572.10^{-6}[m], while for the second case we have $\delta l^0{}_1$ =$\delta l^0{}_2$ =-0.690.10^{-6} [m]. The deformations yield deflection of the end effector with respect to its initial position, equal to $\delta X^0=\delta Y^0$= -3.432.10^{-6}[m] in the first case and to $\delta X^0=\delta Y^0$= -4.142.10^{-6}[m] for the second case.

Table 2: Calculated values of the mechanical parameters.

[m]	δq_1 [rad]	δq_2 [rad]	δw_{11} [rad]	δw_{12} [rad]	δw_{21} [rad]	δw_{22} [rad]	δX [m]	δY [m]	Fl_1 [N]	Fl_2 [N]
$\delta l_1 = 0.263\ 10^{-3}$	$8.72\ 10^{-3}$	0	0	0	$3.589\ 10^{-3}$	0	$1.572.10^{-3}$	0	-13.6	0
$\delta l_2 = 0.263\ 10^{-3}$	0	$8.72\ 10^{-3}$	$3.589\ 10^{-3}$	0	0	0	0	$1.572.10^{-3}$	0	-13.6

6 CONCLUSIONS

Piezo actuated micromanipulators with serial-parallel structure including elastic joints are subject of this paper. A kinematic model of the micro manipulators is build using a pseudo rigid body method, where elastic joints are modelled as revolute joints.

A stiffness model is created to estimate the manipulator stiffness by stiffness reduction of all elastic joints. In order to eliminate the backlash and to improve the performance of the piezo-actuators the parallel structure of the manipulator must be tensed preliminary. Two approaches are proposed and presented here for preliminary tension of parallel manipulator structure:

1. Deflection from the initial manipulator state by driving joints motion implemented in the assembly;

2. Preliminary tensioning of separate elastic joints.

The first approach is easier for realization as the deviations are performed only in the driving joints. The elastic joints are tensed to a different degree.

Tensioning of all the elastic joints is realized to the admissible values by the second approach. Thus, the maximal values of the tensioning forces are achieved

The first approach is chosen for the robot developed for cell injection. The actuators A_1 and A_2 are mounted with deviations related to the values shown in the first column of Table 2. The robot with mechanical construction preliminary tensioned experimented achieves motion along the axes X, Y и Z as follows: ΔX=ΔY= 180 [μm], ΔZ=60 [μm]. Minimal displacement obtained experimentally is 30[nm]

Further numeric experiments and tests are under consideration for more stiff elastically joints and structures with arbitrary location of the actuators.

ACKNOWLEDGEMENTS

This work was funded by the European Commission through the FP6 Integrated Project HYDROMEL with contract No. FP6 NMP2-CT-2006-026622, to which the authors are expressing their acknowledgements.

REFERENCES

Fatikow, S., Munassypov, R., 1996, An Intelligent Micromanipulation Cell for Industrial and Biomedical Applications Based on a Piezoelectric Microrobot, *5th Int. Conf. on MicroElectro, Opto, Mechanical Systems and Components*, Berlin, 17-19 Sept., pp. 826-828.

Kasper, R.; Heinemann, W.; Wagner, A., 1998, Modelling and Control for Piezoelectric Actuators for High Speed Applications. *4th Inl. Conf. on Motion and Vibration Control*, Vol. 1, ETH Zürich, 231-236.

Lee, K., Arjunan, S., 1999, A three-degree-of-freedom micromotion in-parallel actuated manipulator, *IEEE Trans. on "Robot. and Autom.", vol. 7,* 238-247.

Ionescu Fl., Kostadinov, K., 2002, Piezo actuated micro robot for micro and nano manipulations, *ARA Journal, Vol. 2000-2002, Nr.25-27, Montréal,* 98-103.

Guergov, S., 2005, Multifunctional technological system with complex functions, reconfigurable structure, high level of integration and self-organizing elements. *Proc.of V Int. Congress"Computer Science for Design and Technology", STANKIN, Moscow*, 31-36.

Prusak, D., Tadeusz Uhl, 2009, Novel Type of Hybrid 3-DOF Micromanipulator with Piezoelectric Actuators - *Mechanical Construction and Simulations, Solid State Phenomena, Vols. 147-149*, pp25 -30.

Zhang, W., Zou, J., Watson, L., Zhao, W., 2002, The constant- jacobian method for kinematics of a three DOF planar micro-motion stage, *Journ.of Robotic Systems 19(2).* 63–72.

Pham, H., Chen, I., 2005, Stiffness modeling of flexure parallel mechanism, *Precision. Engineering 29*, 467–478.

Carbone, G., Lim, Hun-ok, Takanishi, A., Ceccarelli, M., 2006, Stiffness analysis of biped humanoid robot, Mechanism and Machine Theory, 41, 17–40.

Pashkevich, A., Chablat, D. ,Wenger, Ph., 2009, Stiffness analysis of overconstrained parallel manipulators, Mechanism and Machine Theory, Vol. 44, 966–982.

Chakarov, D., 2004, Study of the Antagonistic Stiffness of Parallel Manipulators with Actuation Redundancy, Mechanism and Machine Theory, Vol. 39/6, 583-601.

Chakarov, D., Abed Al-Wahab, M., Kasper, R., Kostadinov, K., 2007, Synthesis of tense piezo structures for lokal mikro & nano manipulations. 8th Magdeburg days of mechanical engineering, Oct.10-11, Magdeburg, pp.173-180.

A NOVEL VISION-BASED REACTIVE NAVIGATION STRATEGY BASED ON INVERSE PERSPECTIVE TRANSFORMATION[*]

Francisco Bonin-Font, Alberto Ortiz and Gabriel Oliver
Department of Mathematics and Computer Science, University of the Balearic Islands
Ctra de Valldemossa Km 7.5, 07122, Palma de Mallorca, Spain
francisco.bonin@uib.es, alberto.ortiz@uib.es, goliver@uib.es

Keywords: Mobile robots, Vision, Obstacle Avoidance, Feature Tracking, Inverse Perspective Transformation, SIFT.

Abstract: This paper describes a new vision-based reactive navigation strategy addressed to mobile robots, comprising obstacle detection and avoidance. Most of the reactive vision-based systems base their strength uniquely on the computation and analysis of quantitative information. The proposed algorithm combines a quantitative process with a set of qualitative rules to converge in a robust technique to safely explore unknown environments. The process includes a feature detector/tracker, a new feature classifier based on the Inverse Perspective Transformation which discriminates between object and floor points, and a qualitative method to determine the obstacle contour, their location in the image, and the course that the robot must take. The new strategy has been implemented on mobile robots with a single camera showing promising results.

1 INTRODUCTION

Visual techniques for detecting and tracking main scene features have been notably improved over the last few years and applied to robot navigation solutions. Zhou and Li (Zhou and Li, 2006) detected ground features grouping all coplanar points that have been found with the Harris corner detector (Harris and Stephens, 1988). Lowe (Lowe, 2004) developed the Scale Invariant Feature Transform (SIFT) method to extract highly discriminative image features, robust to scaling, rotation, camera view-point changes and illumination changes. Rodrigo *et al* (Rodrigo et al., 2006) estimated the motion of a whole scene computing a homography matrix for every different scene plane. Mikolajczyk and Schmid (Mikolajczyk and Schmid, 2005) compared the performance of different descriptors for image local regions showing that, for different region matching approaches SIFT yields the best performance in all tests. The Inverse Perspective Transformation (*IPT*) has been successfully used in obstacle detection procedures. Mallot *et al* (Mallot et al., 1991) analyzed variations on the optical flow computed over the Inverse Perspective Transformation of consecutive frames to detect the presence of obstacles. Bertozzi and Broggi (Bertozzi and Broggi, 1997) applied the *IPT* to project two stereo images onto the ground. The subtraction of both projections generate a non-zero pixel zone that evidences the presence of obstacles. Ma *et al* (Ma et al., 2007) presented an automatic pedestrian detector based on *IPT* for self guided vehicles. The system predicts new frames assuming that all image points lie on the floor, generating distorted zones that correspond to obstacles.

This paper addresses the problem of obstacle detection and avoidance for a safe navigation in unexplored environments. First, image main features are detected, tracked across consecutive frames, and classified as obstacles or ground using a new algorithm based on *IPT*. Next, the edge map of the processed frame is computed, and edges comprising obstacle points are discriminated from the rest of the edges. This result gives a qualitative idea about the position of obstacles and free space. Finally, a new version of the Vector Field Histogram (Borenstein and Koren, 1991) method, here adapted to systems equipped with visual sensors, is applied to compute a steering vector which points towards the areas into which the vehicle can safely move. The rest of the paper is organized as follows: the method is outlined in Section 2, experimental results are exposed and discussed in Section 3, and finally, conclusions and forthcoming work are given in Section 4.

[*] This work is partially supported by DPI 2008-06548-C03-02 and FEDER funding.

2 THE NEW METHOD

2.1 Inverse Perspective Transformation

The Direct Perspective Transformation is the first-order approximation to the process of taking a picture. The line that connects a world point with the lens intersects the image plane and defines its unique image point. The Inverse Perspective Transformation specifies the straight line upon which the world point corresponding to a certain image point must lie. (Hartley and Zisserman, 2003) describes the Direct and Inverse Perspective Transformation processes and both are also modeled in (Duda and Hart, 1973), as well as the expressions to calculate the world coordinates for points lying on the floor ($z = 0$):

$$x = X_0 - \frac{Z_0 x_p cos\theta + (y_p sin\varphi - f cos\varphi)(Z_0 sin\theta)}{y_p cos\varphi + f sin\varphi} \quad (1)$$

$$y = Y_0 - \frac{Z_0 x_p sin\theta - (y_p sin\varphi - f cos\varphi)(Z_0 cos\theta)}{y_p cos\varphi + f sin\varphi} \quad (2)$$

where (x_p,y_p) are the point image coordinates, (x,y) are the point world coordinates, (X_0,Y_0,Z_0) are the lens world coordinates at the moment in which the frame has been taken, f is the focal length, and θ and φ are the yaw and pitch angles of the camera, respectively.

2.2 Obstacle and Ground Points

Presuming that all image points lie on the floor (i.e. $z = 0$), their (x,y) world coordinates can be calculated using (1) and (2). This is an incorrect assumption for points of obstacles that protrude vertically from the floor. As a consequence, the (x,y) world coordinates (for $z = 0$) of an obstacle point are different when they are calculated from two consecutive images, and different to the obstacle point real (x,y) world coordinates. However, the (x,y) world coordinates (for $z = 0$) of a ground point, are equal when they are computed from two consecutive images, and equal to the real (x,y) ground point world coordinates. Hence, assuming $z = 0$ and analyzing the distance between the resulting (x,y) point world coordinates for $z = 0$, calculated across two consecutive images, one can distinguish if the point belongs to an object or to the floor:

$$D = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \Rightarrow \begin{cases} if D > \beta \Rightarrow \text{obstacle}, \\ if D \leq \beta \Rightarrow \text{ground} . \end{cases} \quad (3)$$

where (x_1,y_1) and (x_2,y_2) are the (x,y) feature world coordinates (for $z = 0$) at instants t_1 and t_2 respectively and β is the threshold for the maximum difference admissible between (x_1,y_1) and (x_2,y_2) to consider both as the same point. Ideally β should be 0.

Figure 1 illustrates the idea. Two frames of a scene are taken at instants t_1 and t_2. Point P_{2w} is on the ground. Its projection into the image plane at instants t_1 and t_2 generates, respectively, the image points P_{2i0} and P_{2i1}. The Inverse Transformation of P_{2i0} and P_{2i1} generates a single point P_{2w}. P_{1w} is an obstacle point. Its projection into the image plane at t_1 and t_2 generates, respectively, points P_{1i0} and P_{1i1}. However, the projection of P_{1i0} and P_{1i1} onto the ground plane (i.e. Inverse Transformation assuming $z = 0$) generates two different points on the ground, namely, P'_{1w} and P''_{1w}.

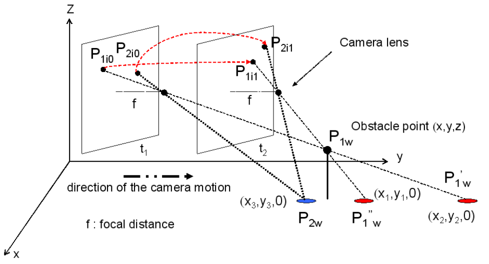


Figure 1: The IPM-based obstacle detection principle.

2.3 Feature Detection and Tracking

The first step of the obstacle detection algorithm is to find a sufficiently large and relevant set of image points, and establish a correspondence of all these points between consecutive frames. SIFT features (Lowe, 2004) have been chosen as the features to track because of their robustness to scale changes, rotation and/or translation as well as changes in illumination and view point. In order to filter out possible wrong correspondences between points in consecutive frames, outliers are filtered out using RANSAC and imposing the epipolar constraint. After the detection and tracking process, features are classified as ground or obstacle.

Small changes in the distance threshold β can alter the classification of those points which have a D value (3) close to β. In order to decrease the sensitivity of the classifier with regard to β, all these points are left unclassified. Additionally, in a previous training phase, and for each different scene, histograms of D values for well classified and misclassified points are built and analyzed. For every different scene, D values of ground points wrongly classified as obstacle are stored in a database. In the autonomous navigation phase, all object points with a D value included in that set of stored D values of the current scene, are neither classified. In this way, nearly all ground points classified as obstacles are eliminated, reducing the risk of detecting false obstacles, and although some true obstacle points are also removed, the remaining ones are sufficient to permit the detection of those obstacles.

2.4 Obstacle Profiles and the Navigation Strategy

SIFT features are usually detected at regions of high intensity variation (Lowe, 2004) and besides, commonly they are near or belong to an edge. Obstacles usually have a high degree of vertical edges and have one or some points in contact with the floor. All detected obstacle points are most likely to be contained or near a vertical edge which must belong to that obstacle. Hence, the next step of the algorithm is the computation of the processed images edge map, and the detection of all complete edges that comprise real obstacle points. This permits to isolate the obstacle boundaries from the rest of the edges and to get a qualitative perception of the environment. Obstacle points wrongly classified as ground can be reclassified if they are comprised in an edge that contains other obstacle points.

In order to combine a high degree of performance in the edge map computation with a relatively low processing time, our edge detection procedure runs in two steps (Canny, 1986): a) The original image is convolved with a 1D gaussian derivative, detecting zones with high vertical gradient from smoothed intensity values with a single convolution; b) A process of hysteresis thresholding is applied. Two thresholds are defined. A pixel with a gray level above the highest threshold is classified as edge pixel. A pixel with a gray level above the lowest threshold is classified as edge if it has in its vicinity a pixel with a gray value higher than the highest threshold.

The proposed navigation strategy has been inspired by (Borenstein and Koren, 1991). Only obstacles detected inside a *ROI* (Region of Interest) centered at the bottom of the image are considered to be avoided. This guarantees a virtual 3-D sphere of safety around the robot. The image *ROI* is in turn divided in angular regions. Those polar directions, corresponding to angular regions occupied by a real obstacle boundary are labeled as forbidden and those free of obstacle boundaries are included in the set of possible next movement directions. This process results in a polar map of free and occupied zones. Obstacle-free polar regions which are narrower than a certain threshold (determined empirically and depending on the robot size) are excluded from the possible motion directions. If all angular regions are narrower than the defined threshold, the algorithm returns a stop order. The next movement direction is given as a vector, pointing to the widest polar obstacle-free zone. Positive angles result for turns to the right and negative angles for turns to the left. The computed steering vector qualitatively points towards the free space and the complete algorithm gives a reasonable idea of whether this free space is wide enough to continue the navigation through it.

3 EXPERIMENTAL RESULTS

A Pioneer 3Dx mobile robot with a calibrated wide angle camera was programmed to navigate at 40mm/s in different environments to test the proposed strategy: environments with obstacles of regular and unregular shape, environments with textured and untextured floor, and environments with specularities or with low illumination conditions. Operative parameter settings: image $ROI = 85$ pixels; for the hysteresis thresholding: low level= 40 and high levels= 50; camera height= $430mm$; $\varphi = -9°$; initial $\theta = -2°$, and finally, $f = 3.720mm$. For each scene, the complete navigation algorithm was run over successive pairs of 0.56-second-separation consecutive frames so that the effect of *IPT* was noticeable. Increasing the frame rate decreases the *IPT* effect over the obstacle points, and decreasing the frame rate delays the execution of the algorithm. Frames were recorded and down-sampled to a resolution of 256×192 pixels, in order to reduce the computation time. All frames were also undistorted to correct the error in the image feature position due to the distortion introduced by the lens, and thus, to increase the accuracy in the calculation of the point world coordinates.

In order to assess the classifier performance ROC curves were computed, defining obstacle points classified as obstacle as true positives (TP), obstacle points classified as ground as false negatives (FN), ground points classified as floor as true negatives (TN) and ground points classified as obstacles as false positives (FP). The AUC (Area Under the Curve) were calculated as a measure of success classification rate, suggesting success rates greater than 93% (Bonin-Font et al., 2008). The β operational value (3) was obtained for every scene minimizing the cost function $f(\beta) = FP(\beta) + \lambda FN(\beta)$. During the experiments, λ was set to 0.5 to prioritize the minimization of false positives over false negatives. The value of f(β) was calculated for every pair of successive images, changing β. For a varied set of scenes differing in light conditions and/or floor texture, the optimum β had a coincident value of 20mm.

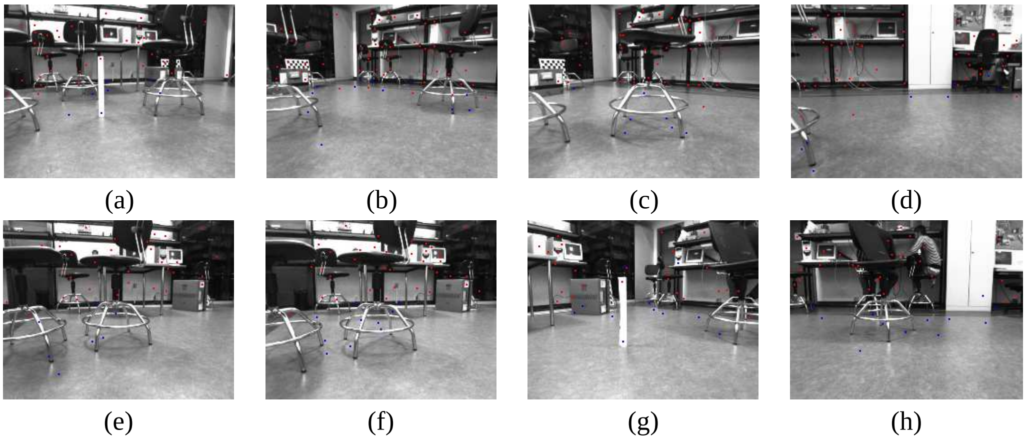


Figure 2: Scene 1. (a) to (d)- Experiment 1. (e) to (h)- Experiment 2: Object and floor points after the filter.

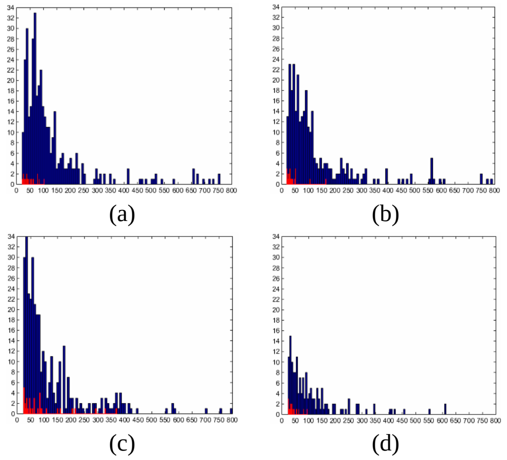


Figure 3: Histograms of D values: TP (blue) and FP (red).

Images (a), (b), (c) and (d) of figure 2 show the undistorted second frame of several pairs of consecutive images, recorded and processed on-line. Images show SIFT features classified as ground (blue) and classified as obstacles (red). Every image was taken just before the robot had to turn to avoid the obstacles it had in front. Notice that all four pictures present a few false positives on the floor.

Histograms of D values for TP (in blue) and FP (in red) are presented in figure 3. Plot (a) corresponds to scene 1 (figure 2), plot (b) to scene 2, plot (c) to scene 3 and plot (d) corresponds to scene 4. Scenes 2,3 and 4 are shown in figure 4. These histograms count false and true positives for different D values, in all frames recorded and computed by the algorithm during a complete sequence. Although histograms belong to environments with different lighting conditions or floor textures, and scenarios with inter-reflections or specularities, results were commonly similar: most of the true positives presented D values between 20mm and 300mm, and the majority of false positives had D values between 20mm and approximately 80mm. All positives with D values between 20mm and 80mm were filtered and left unclassified. This filtering process increases AUCs until the 96%, however, obstacle points near the floor have more probabilities of been miss-classified than others since their D value can be lower than 20mm.

Pictures (e) to (h) of figure 2 were taken during a second experiment through the environment of scene 1. In this experiment, the filter outlined in the previous paragraph was applied. Notice that all false positives have been eliminated. This reduces the risk of detecting false obstacles but maintains a sufficient number of true positives to detect the real obstacles. After the process of feature detection, tracking, and classification, the algorithm localizes every object point in the edge map of the second frame, and then searches for all edge pixels which are inside a patch window of 8×13 pixels, centered in the feature image coordinates. Every edge is tracked down starting from the object point position until the last edge pixel is found, and considering this last edge pixel to be the point where the object rests on the floor. This process results into the identification of the object vertical contours. The consecutive execution of the complete algorithm using successive image pairs as input results in a collection of consecutive steering vectors used as the continuous motion orders. After every robot turn, the value of the camera yaw angle is updated, adding the turn angle to the previous yaw value. The camera world coordinates are calculated composing the robot orientation and its center world coordinates obtained via dead reckoning, with the relative camera position respect to the center of the robot.

Pictures from (a) to (d), (i) to (l) and (p) to (s) in figure 4 show the second frame of different pairs of consecutive images, recorded and processed during the navigation through the scenarios 2, 3 and 4, respectively. Every image was taken before the robot had to turn to avoid the frontal obstacles, and show obstacle (in red) and ground points (in blue). Scene 2 presents inter-reflections, specularities, and a lot of obstacles with regular and irregular shapes. Scene 3 shows a route through a corridor with a very high textured floor, columns and walls. Scene 4 presents bad illumination conditions, a lot of inter-reflections on the floor, and some image regions (walls) with almost homogeneous intensities and/or textures, which results in few distinctive features and poorly edged obstacles. Walls with a very homogeneous texture and few distinctive features can present difficulties for its detection as an obstacle. In all scenes, all obstacle points with a *D* value between 20mm and 80mm were left unclassified, except in scene 4, where, only those obstacle points with a *D* value between 20mm and 45mm were filtered out. Pictures (e) to (h), (z) to (o) and (t) to (x) of figure 4 show the vertical contours (in orange) comprising obstacle points. See attached to every picture the angle of the computed steering vector. For example, in picture (x) objects are out of the *ROI*, then, the computed turn angle is 0° (follow ahead). In picture (e) the obstacles are partially inside the *ROI*, so the robot turns to the right (40°). Despite scene 4 presents a poor edge map and few SIFT features, the resulting steering vectors still guide the robot to the obstacles-free zone. Plots (1) to (4) show an illustration of the environment and the robot trajectory (blue circle: the starting point; red circle: the final point) for scenes 1, 2, 3 and 4, respectively. In all scenes, all features were well classified, obstacle profiles were correctly detected and the robot navigated through the free space avoiding all obstacles. The steering vector is computed on the image and then it is used qualitatively to guide the robot.

4 CONCLUSIONS

This paper introduces a new vision-based reactive navigation strategy addressed to mobile robots. It employs an *IPT*-based feature classifier that distinguishes between ground and obstacle points with a success rate greater than 90%. The strategy was tested on a robot equipped with a wide angle camera and showed to tolerate scenes with shadows, inter-reflections, and different types of floor textures or light conditions. Experimental results obtained suggested a good performance, since the robot was able to navigate safely. In order to increase the classifier success rate, future research includes the evaluation of the classifier sensitivity to the camera resolution or focal length. The use of various β values, depending on the image sector that *D* is being evaluated, can also increase the classifier performance.

REFERENCES

Bertozzi, M. and Broggi, A. (1997). Vision-based vehicle guidance. *Computer*, 30(7):49–55.

Bonin-Font, F., Ortiz, A., and Oliver, G. (2008). A novel image feature classifier based on inverse perspective transformation. Technical report, University of the Balearic Islands. A-01-2008 (http://dmi.uib.es/fbonin).

Borenstein, J. and Koren, I. (1991). The vector field histogram - fast obstacle avoidance for mobile robots. *Journal of Robotics and Automation*, 7(3):278–288.

Canny, J. (1986). A computational approach to edge detection. *IEEE TPAMI*, 8(6):679 – 698.

Duda, R. and Hart, P. (1973). *Pattern Classification and Scene Analysis*. John Wiley and Sons Publisher.

Harris, C. and Stephens, M. (1988). Combined corner and edge detector. In *Proc. of the AVC*, pages 147–151.

Hartley, R. and Zisserman, A. (2003). *Multiple view geometry in computer vision*. Cambridge University Press, ISBN: 0521623049.

Lowe, D. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110.

Ma, G., Park, S., Mller-Schneiders, S., Ioffe, A., and Kummert, A. (2007). Vision-based pedestrian detection - reliable pedestrian candidate detection by combining ipm and a 1d profile. In *Proc. of the IEEE ITSC*, pages 137–142.

Mallot, H., Buelthoff, H., Little, J., and Bohrer, S. (1991). Inverse perspective mapping simplifies optical flow computation and obstacle detection. *Biological Cybernetics*, 64(3):177–185.

Mikolajczyk, K. and Schmid, C. (2005). A performance evaluation of local descriptors. *IEEE TPAMI*, 27(10):1615–1630.

Rodrigo, R., Chen, Z., and Samarabandu, J. (2006). Feature motion for monocular robot navigation. In *Proc. of the ICIA*, pages 201–205.

Zhou, J. and Li, B. (2006). Homography-based ground detection for a mobile robot platform using a single camera. In *Proc. of the IEEE ICRA*, pages 4100–4101.

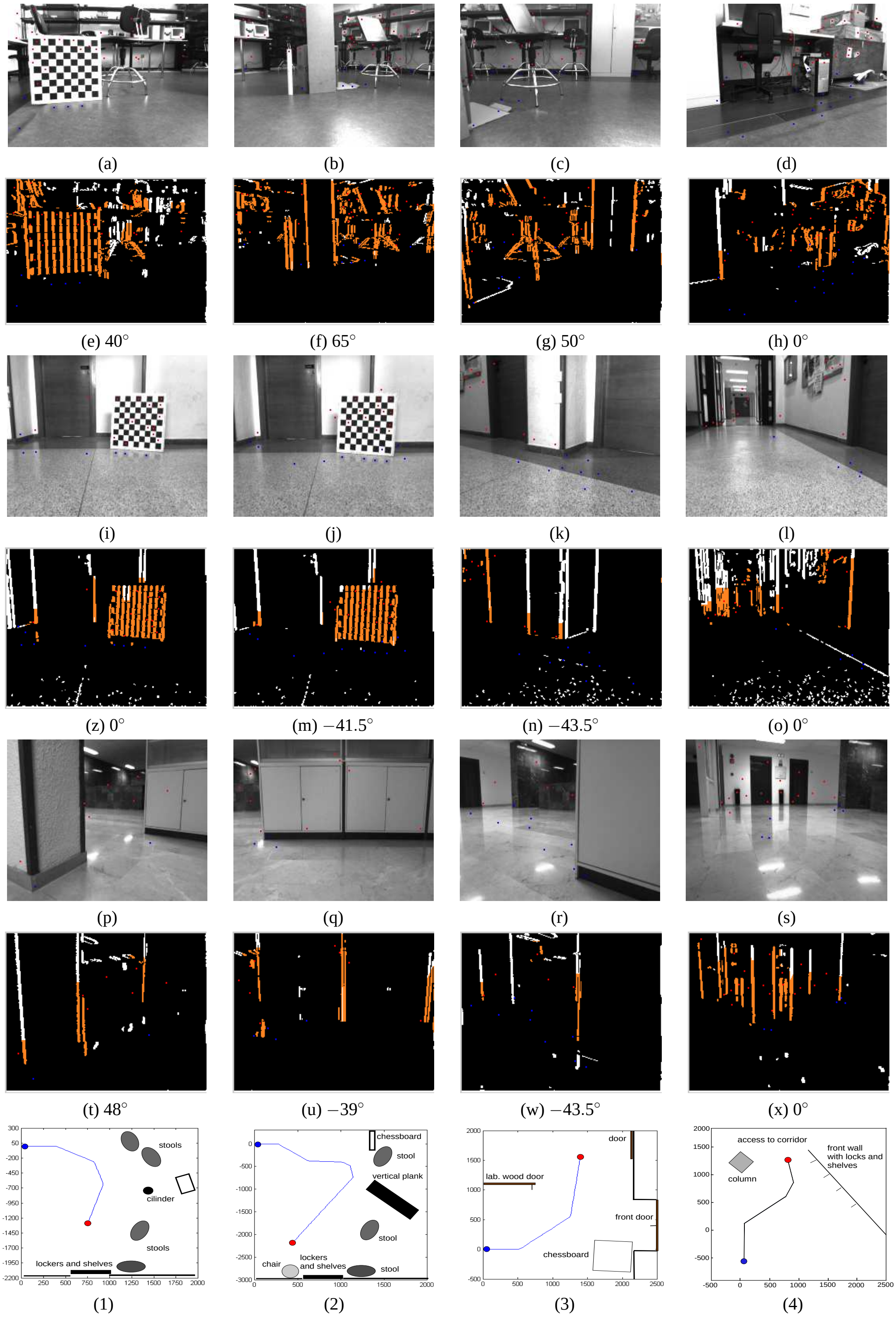


Figure 4: (a) to (d) Scene 2, (i) to (l) Scene 3, (p) to (s) Scene 4. (e) to (h), (z) to (o) and (t) to (x), vertical contours of Scene 2, 3 and 4, respectively. (1) to (4): robot trajectories.

VEHICLE ACCELERATION PREDICTION USING SPECIFIC ROAD CURVATURE POINTS

Aušra Vidugirienė, Andriejus Demčenko and Minija Tamošiūnaitė
Vytautas Magnus University, Vileikos st. 8, LT-44404 Kaunas, Lithuania
ausrine@gmail.com, andriejus.demcenko@gmail.com, m.tamosiunaite@if.vdu.lt

Keywords: Human-like driving, Intelligent driver's assistance, Longitudinal control, Curve-based parameters.

Abstract: In the work vehicle acceleration prediction issue is discussed. Three types of parameters are used for prediction system input: CAN-bus parameters – speed and curvature, derived speed parameters and newly offered specific curve point parameters, denoting changes in a curve. The real road data was used for predictions. Road curvature segments were divided into single and S-type curves. Acceleration was predicted using artificial neural networks and look-up table. The look-up table method showed the best results with newly offered specific curve parameters.

1 INTRODUCTION

Driving assistance systems are becoming a usual component of modern cars. Here we are developing an algorithm that could aid to driver's assistance on a curved country road. One way to develop such algorithms is through modelling driver's behaviour. Once we have a model that predicts driver's behaviour, we can compare actual behaviour with the prediction, and warn the driver if there is inconsistency.

In the field of driving action description several clear-cut situations have been studied exhaustively: lane following (Fenton, 1988; Mammar et al., 2006), car following at a safe distance (Gipps, 1981; Olstam et al., 2004), lane change (Gipps, 1986; Salvucci et al., 2007). For lane following on a curved road an extensive theory has been developed, mainly based on control engineering approaches (Hsu et al., 1998; Yuhara et al., 2001; Chen et al., 2006; Mammar et al, 2006). Yet speed control (so called longitudinal control), including speed on curves, has only been studied extensively from a car stability perspective (Jin et al., 2007; Hel et al., 2007; Song, 2008). Alternatively, we focus on predicting speed (or acceleration) profiles of individual drivers, where they are performing not at the limits of car possibilities, but rather in their comfort-driving modes. Speed prediction of an individual driver is a much more complicated problem as compared to steering prediction, because of much stronger influence of contextual information, and less constraint for a driver in choosing the actual speed profile. There are only singular investigations concerning speed prediction based on speed profiles of individual driver, e.g. (Partouche et al., 2007), and success of such work until now is quite limited.

In this study we apply learning techniques to predict individual driver's acceleration on a curve. Neural networks and look-up tables are employed for prediction. Real road driving data is used, and input parameters for driver's action prediction are analyzed.

Relatively long real road data sequences are required for predicting acceleration on a curve. This is because speed control process has a wider time scale than steering, i.e. for generating velocity control the driver reacts rather to future events, like upcoming curves, than immediate situations. E.g. it was observed in this study that deceleration in front of a curve starts 3-6 s or on some occasions even up to 10 s in advance. Consequently, multiple curve taking situations in the recordings are required to derive the algorithm that predicts an expected acceleration profile for a particular driver on a particular curve. This makes the problem of speed (or acceleration) prediction on a curve difficult to address, especially when using real-road data.

2 DATA FOR ACCELERATION PREDICTION

Two data sets were used for the study. The first data set was collected during November-December, 2006. The second data set was collected in December, 2007. Both data sets were obtained on country roads nearby Lippstadt, Germany; at day light, on a test car (Volkswagen Passat). In the data set from 2006, ten recordings, approximately six minutes length each were provided. Five of those recordings were obtained on the same road, using forward direction, and the other five were obtained using backward direction. The recordings were coming from two drivers: eight recordings of the first driver, and two recordings of the second driver. The second set of data (year 2007) consisted of six recordings. Those recordings were obtained on a different road as compared to the recordings from the year 2006. The recordings were again obtained in forward and backward directions, duration of ten minutes each. This set of recordings was repeated three times for three different drivers.

The test car control data were recorded using CAN-bus with a sampling interval of 0.06 s. The following signals were extracted from the CAN-bus and used in the study:

- velocity $v(t)$,
- acceleration $a(t)$,
- curvature of the road $c(t)$; curvature was measured using a gyroscope installed in the car.

3 METHODS

Curvature-based parameters combined with car velocity were employed to predict driver's acceleration. In this work gyroscopically measured curvature was used, as a shortcut proceeding towards further systems, where image processing or digital map information will be used to obtain the curvature in front of a car.

Neural networks and look-up tables were used as function approximation means for prediction. For neural network analysis a simple neural network with one hidden layer was used. There were from two to four neurons in the hidden layer, according to the number of input parameters. Separate learning data sets and test sets were employed. The average of prediction error from ten initializations was calculated to make results more reliable.

In the look-up table approach input parameter values obtained at discrete time moments were stored together with corresponding acceleration signal value. The predictions were made as follows: for the input parameter vector obtained at a specific time moment mean squared error (MSE) was calculated between that vector and every instance of the look-up table. The predicted acceleration was calculated as the mean of ten acceleration values, with the smallest MSE to input parameters. In addition, the acceleration signal was smoothed using 20 point moving average filter (corresponds to 1.2 s) from the previous predictions.

As part of the input vector raw CAN-bus signals: curvature and speed were used, but also a large set of derived parameters was introduced.

Among the derived parameters we used centrifugal acceleration (Hong et al, 2006):

$$a_c = \frac{v^2}{R} \qquad (1)$$

where R denotes the curve radius, and v is the speed. The centrifugal acceleration is considered to be a parameter influencing driving comfort and possibly driver’s actions (Hong et al, 2006).

We used speed differences $S_d=v(t)-v(t-\Delta t)$ over several second intervals (Δt=0.5, 1.0, 1.5, 2.0, 2.5, 3.0 s) to account for previous acceleration or deceleration actions. If a car decelerated, the speed difference was negative, and if the car was accelerating, the speed difference was positive.

For acceleration on a curve, features like the distance to a start of a curve or the distance to the end of a curve are important. We introduced a set of curve shape based points (see Fig. 1), that later were employed to derive features for acceleration analysis. All the parameters’ notations are listed in Table 1.

Table 1: Curve- and speed-derived parameters.

Parameter class	Parameter	Notation
CAN-bus derived parameters	Centrifugal acceleration	CA
	Speed difference (now-Xs back)	SD-X
Single curve parameters	Start	S
	Start peak	SP
	End peak	EP
	End	E
S-curve parameters	S-curve start peak	SSP
	S-curve zero crossing	S0
	S-curve end peak	SEP

Two different curve shapes were analyzed in this work:

- Single curve that has 4 specific points (start, start peak, end peak and end; see Fig. 1a),
- S-shaped curve that has 7 specific points (start, start peak, S start peak, S zero crossing, S end peak, end peak and end; see Fig. 1b).

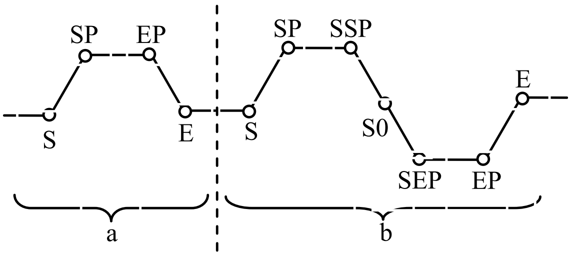


Figure 1: Specific curve-based points' scheme: a) single curve with 4 specific points: start (S), start peak (SP), end peak (EP) and end (E); b) S-shaped curve with 7 specific points: start (S), start peak (SP), S start peak (SSP), S zero crossing (S0), S end peak (SEP), end peak (EP) and end (E).

The features were described as distances from specific points. Examples of feature time series are provided in Fig. 2.

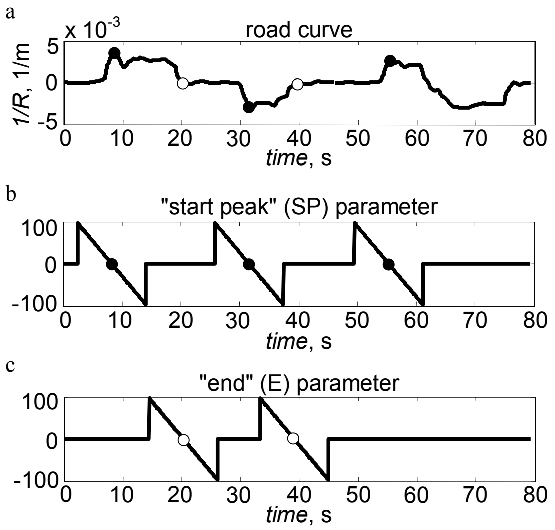


Figure 2: Curvature (a) and features describing distances to specific points on a curve (b and c). Features for the points 'Start peak' and 'End' are shown. The points 'Start peak' are marked by black points and the points 'End' in circles.

Before a specific point it is considered how much time is left to that point, and after the point it is pointed out how much time has passed since the specific point had been passed. A feature is started to be considered six seconds in advance before a specific point is reached and the point is "forgotten" six seconds after it has been passed. Before the point a feature is positive, at the point it is zero, and after the point it is negative.

An algorithm to derive feature values is as follows: first, the specific curve point t_p is determined and the feature value for that discrete time moment is set to zero. The feature values are calculated by adding 1 or -1 to the previous value when going through every discrete time step back and forward respectively. The calculations end when $t_{\text{back}}=t_{\text{p}}-100$ and $t_{\text{forward}}=t_{\text{p}}+100$ (100 discrete points corresponds to 6 s according to the signal discretization).

4 ACCELERATION PREDICTION RESULTS

4.1 Acceleration Predictions using Raw CAN-bus Signals

We used curvature $c(t+\Delta)$ where Δ = 4s (that is, four seconds ahead), and speed $v(t)$ to predict acceleration one step forward. The training set was composed of seven curve segments containing clear acceleration-deceleration patterns, and we predicted the segment that was not included into the learning data set. Examples of predicted signals are presented in Fig. 3.

As can be seen in the Fig. 3a, some acceleration events in the learning set are predicted accurately, but there are some other segments in the acceleration profile that the neural network fails to predict.

In the test sets (Fig. 3 b,c), if measured formally, the error between real and predicted signals would be high. Yet one can observe qualitative correspondence between real and predicted signals, and the presence of acceleration/deceleration events is predicted correctly with 1-2s precision. With slower acceleration dynamics it is a reasonable result. This could be enough for approximate detection of the moments when deceleration is required. Specifically, prediction of deceleration moment is important for driver assistance on a curved country road. However, we have observed that the results were varying a lot with different initializations of the artificial neural network. Consequently, we were looking for a method that could allow more stable acceleration predictions.

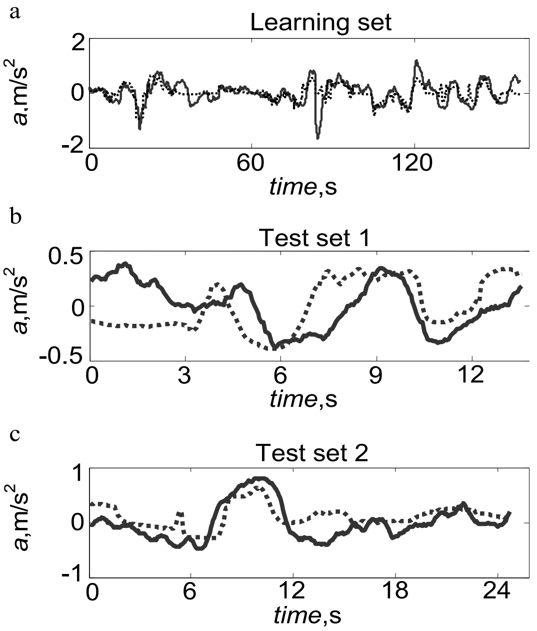


Figure 3: Two examples of acceleration prediction by ANN on a training set (a), and the test set (b and c). Input parameters: curvature $c(t+\Delta)$, where Δ=4s, and speed $v(t)$. Original signal is marked as solid curve; predicted signal is marked as dotted curve.

4.2 Acceleration Predictions using Specific Curve Features

We used specific curve point-based features to improve on acceleration prediction. A look-up table was used to map between features and actions.

For the current experiment for the learning set six minutes of driving of the same driver were used (recording from year 2007), and approximately 1.5 minute for each driver were used for testing. Data for testing were not included into the learning data set.

The resulting predictions (test sets) for two drivers are provided in Fig. 4 and 5.

In the top panel (Fig. 4 and 5) gyroscopically measured curvature is presented. Bigger details correspond to real road curvature, while smaller details at the top of the curve may be attributed to over-steering events. Acceleration (lower panel) shows much more details, as compared to curvature, but one can observe episodes of deceleration, performed as a sequence of several (usually 2-3) deceleration events in front of a curve. Speed usually starts increasing at the second half of the curve. Those rules can be derived for single curves (seconds approx. 50 to 90 in both plots), but for more complex curves the situation is difficult to specify.

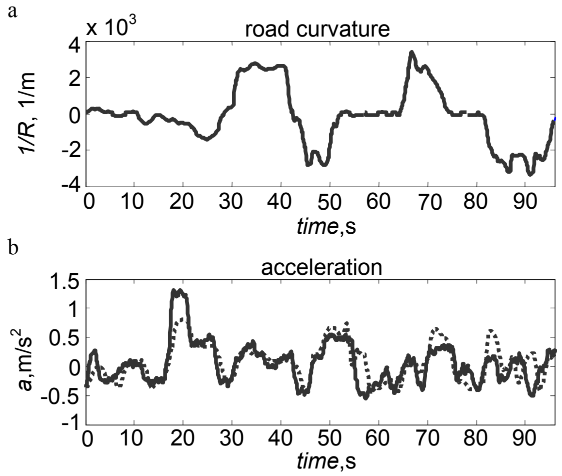


Figure 4: Gyroscopically measured curvature of the drive (a); original (solid curve) and predicted (dotted curve) acceleration signal (b); first driver. Input parameters: SP, E, CA, SD-2s.

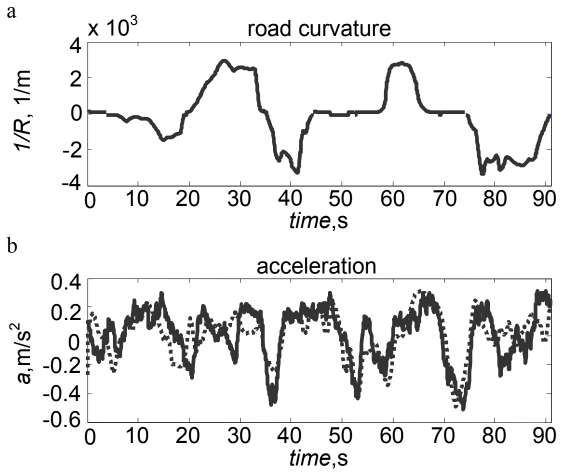


Figure 5: Gyroscopically measured curvature of the drive (a); original (solid curve) and predicted (dotted curve) acceleration signal (b); second driver. Input parameters: SP, E, CA, SD-2s.

In the first driver case (see Fig. 4b) the predicted signal corresponds to the original acceleration signal quite well. At the second 20 the predicted signal does not reach the real acceleration amplitude, but it starts to increase at the same moment as the true signal. At the intervals from 70 to 75 s and from 82 to 85 s the prediction gives bigger acceleration and decreases to the same level as original signal. The interval from 85 s to the end of test signal does not correspond to the real acceleration signal. That could be associated with over-steering that can be observed in Fig. 4a, (85 to 90 s).

With the second driver (Fig. 5) one can observe that the acceleration profile is reproduced less well between seconds 10 and 30, where there is a complex curve, but the profile is reproduced much better for single curves.

The interval from 77 s to the end of test signal does not correspond to the real acceleration signal as well. That could be also attributed to over-steering that is seen from Fig. 5a.

Summarizing the results it can be concluded that the algorithm grasps the moments of acceleration and deceleration on the curve well.

Selected parameter subsets have been analyzed to find out which parameter subset could serve best for acceleration prediction. Prediction error numerical values for various parameter combinations are listed in Tables 2 – 4.

Parameter combinations were investigated in the case when all curves were considered as single first. E.g. an S-shape curve was considered as a sequence of two single curves with appropriate single curve points. It was found that two points are most important for acceleration prediction: SP and E. When complementing curve shape features with centrifugal acceleration, and speed change from 1.5-2 seconds ago to a current moment, prediction improved for both drivers, but for driver B the result was still a small fraction better when adding point S (see Table 2).

Table 2: Prediction with look-up table considering complex curves as composed of single curves: mean squared error for various parameter combinations.

Parameter sets	Driver A	Driver B
SP, E	0.27	0.21
SP, E, CA	0.26	0.21
SP, E, CA, S, SD-2s	0.26	0.16
SP, E, CA, EP	0.29	0.20
SP, E, CA, SD-2s	0.20	0.17

The situation was improved by separately analyzing S-type curves (see Table 3). The best result for the data set was obtained when specific S curve parameters SSP, S0, SEP were not included into the input parameter vector (that is, even from S-type curves we were analyzing only the points SP and E, that are present both on a single and an S-type curve). This could possibly change when larger data sets are analyzed.

When analyzing which time window would tell the history of driver’s acceleration best (Table 4), and consequently allow to predict drivers next action with the smallest error, it was found that time windows of 1 s, 1.5 s and 2 s performed almost equally well, and longer as well as shorter time intervals performed worse for both drivers.

Table 3: Prediction with look-up table including S-curve parameters: mean squared errors for various parameter combinations.

Parameter sets	Driver A	Driver B
SP, E, CA, SD-2s	0.16	0.13
SP, E, CA, SD-2s, SEP	0.18	0.14
SP, E, CA, SD-2s, SEP, SSP	0.18	0.15
SP, E, CA, SD-2s, SSP	0.20	0.15
SP, E, CA, SD-2s, SEP, S0, SSP	0.22	0.16
SP, E, CA, SD-2s, S0, SSP	0.23	0.15

Table 4: Prediction with look-up table: mean squared error for various speed difference parameters.

Parameter sets	Driver A	Driver B
SP, E, CA, SD-3s	0.19	0.14
SP, E, CA, SD-2.5s	0.17	0.14
SP, E, CA, SD-2s	0.16	0.13
SP, E, CA, SD-1.5s	0.16	0.13
SP, E, CA, SD-1s	0.16	0.13
SP, E, CA, SD-0.5s	0.17	0.15

5 DISCUSSION

Two methods were introduced to predict individual driver's acceleration on a curve. The method employing only simple parameters: speed of the car and curvature at a single point in front of a car, failed to stably predict driver's acceleration. The other method introducing more complicated analysis of a curve shape, supplemented by centrifugal acceleration and history of driver's actions, provided promising results.

Driver's acceleration prediction on a curve is an important task on the way towards intelligent driver's assistance systems, as a big proportion of serious traffic accidents happen due to failure to properly reduce speed on curves (Comte et al, 2000). After developing adequate prediction methods one will have to define thresholds when acceleration profile is to be considered 'unusual' for a driver. However, examples of 'dangerous' speed profiles are difficult to obtain, especially in real road driving situations. Alternatively, one can perform experiments in driving simulators. Here one necessarily needs simulators imitating forces arising while driving a car, because with real road driving we observe much different speed (and acceleration) profiles on curves as compared to those obtained on a simulator with only visual feedback (Partouche et al, 2007) .

On the other hand, some practical tasks can be solved without analysing dangerous acceleration profiles. If one manages to predict with reasonable precision the moment of deceleration in front of a curve, then one can warn on the events where a driver failed to observed the curve, e.g. due to reduced visibility (warning in this case would be based on absence of deceleration event where it should appear).

One could argue that the curve shape features we are introducing are not practical, as stable visual analysis of a scene 6s in front of a car driving at motorway speeds (100 km/h or more) is not realistic to achieve. Our experience with visual analysis prompts the same. Yet with new developments, where interactive roads are foreseen (Jakubiak et al, 2008), or systems where map integrated into the car provides upcoming curvatures (Mammar et al, 2006) would solve the problem.

Turning to details of this study, good acceleration prediction results were obtained when curve shape parameters SP, E, CA, SD-1.5 or SD-2 were provided as input parameters and S-shape curve was analyzed separately. For the first driver the mean squared error of acceleration prediction was 16% and for the second driver the mean squared error was 13%. For the second driver adding parameter S allowed to reduce the error further. Although, those conclusions should only be taken as preliminary, and experiments with more data are required to refine parameter choice.

ACKNOWLEDGEMENTS

This work was supported in part by the European Commission project “Learning to Emulate Perception – Action Cycles in a Driving School Scenario” (DRIVSCO), FP6-IST-FET, contract No. 016276-2.

REFERENCES

Chen, L., Ulsoy, A., 2006. *Experimental evaluation of a vehicle steering assist controller using a driving simulator*. Vehicle System Dynamics, 44, 223-245.

Comte, S. L., Jamson, A. H., 2000. *Traditional and innovative speed-reducing measures for curves: an investigation of driver behaviour using a driving simulator*, Safety Science, vol. 36, issue 3, 137-150

Gipps, P., 1981. *A behavioral Car Following Model for Computer Simulation*. Transportation Research B., 15, 105-111.

Gipps, P., 1986. *A Model for the Structure of Lane Changing Decisions*. Transportation Research B, 20, 107-120.

Fenton, R., 1988. *On the optimal design of an automative lateral controller*, IEEE Transactions on Vehicular Technology, 37, 108-113.

He1, J., Crolla, D., Levesley, M., and Manning, W., 2007. *Coordination of active steering, driveline, and braking for integrated vehicle dynamics control*. Proc. IMechE Vol. 220 Part D: J. Automobile Engineering, 1401-1421.

Hong, I., Iwasaki, M, Furuichi, T, Kadoma, T. (2006) *Eye movement and driving behaviour in curved section passages of an urban motorway*. Proc. IMechE, 220 Part D: J. Automobile Engineering, 1319-1331

Hsu, J., Tomizuka, M., 1998. *Analysis of vision-based lateral control for automated highway systems*. Vehicle System Dynamics, 30, 345-373.

Jakubiak, J., Koucheryavy, Y., 2008. *State of the Art and Research Challenges for VANETs*. 5th IEEE Consumer Communications and Networking Conference, 912-916

Jin, Z., Weng1, J., and Hu, H., 2007. *Rollover stability of a vehicle during critical driving manoeuvres*. Proc. IMechE vol. 221 Part D: J. Automobile Engineering, 1041-1049

Mammar, S., Glaser, S., and Netto, M., 2006. *Time to Line Crossing for Lane Departure Avoidance: A Theoretical Study and an Experimental Setting*. IEEE Transactions on Intelligent Transportation Systems, 7, 226-241

Olstam, J., and Tapani, A., 2004. *Comparison of Car Following Models*. Swedish National Road and Transportation Institute.

Partouche, D., Spalanzani, D., Pasquier, M., 2007. *Intelligent Speed Adaptation Using a Self-Organizing Neuro-Fuzzy Controller*. IEEE Transactions on Intelligent Transportation Systems, 7, 846 - 851

Salvucci, D., Mandalia, H., Kuge, N., and Yamamura, T., 2007. *Lane-Change Detection Using a Computational Driver Model*. Human Factors, 49, 532-542

Song, J., 2008. *Enhanced braking and steering yaw motion controllers with a non-linear observer for improved vehicle stability*. Proc. IMechE, 222 Part D: J. Automobile Engineering, 293-303

Yuhara, N., Tajima, J., 2001. *Advanced Steering System Adaptable to Lateral Control Task and Driver's Intention*. Vehicle System Dynamics, 36, 119-158.

REAL-TIME BIOMETRIC EMOTION ASSESSMENT IN AN IMMERSIVE ENVIRONMENT

Vasco Vinhas, Daniel Castro Silva, Luís Paulo Reis and Eugénio Oliveira
Department of Informatics Engineering, Faculty of Engineering, University of Porto
Artificial Intelligence and Computer Science Laboratory, Rua Dr. Roberto Frias s/n 4200-465, Porto, Portugal
{vvm, dcs, lpreis, eco}@fe.up.pt

Keywords: Emotion Assessment, Biometric Readings, Immersive Digital Environments, Aeronautical Simulation.

Abstract: Both the academic and industrial worlds have increased investment and dedication to the affective computing area in the past years. At the same time, immersive environments have become more and more a reliable domain, with progressively cheaper hardware and software solutions. With this in mind, the authors used biometric readings to perform real-time user emotion assessment in an immersive environment. In the example used in this paper, the environment consisted in a flight simulation, and biometric readings were based on galvanic skin response, respiration rate and amplitude, and phalanx temperature. The detected user emotional states were also used to modify some simulation variables, such as flight plan, weather and maneuver smoothness. The emotion assessment results were consistent with user-described emotions, achieving an overall success rate of 78%.

1 INTRODUCTION

The presence of sensors, actuators and processing units in unconventional contexts is becoming consistently inevitable. This fact brings to both academic and industrial stages the term of Ubiquitous Computing as a regular one. In a parallel, yet complementary line, Affective Computing has recently gained the attention of researchers and business organizations worldwide. As a common denominator for these two concepts resides Emotion Assessment. Although this topic is no novelty by itself, it has been rediscovered in light of the mentioned knowledge areas breakthroughs, as it became theoretically possible to perform real-time minimal-invasive user emotion assessment based on live biosignals at economically feasible levels.

Having all that in mind, the authors envisioned an integrated interactive multimedia system where internal parameters are changed according to the user's emotional response. As the application example in this paper, an aviation environment was considered. The main reasons behind this decision are related to the human fascination for everything related to flying. Still, and as with most things, this attraction co-exists with the fear of flying, usually referred to as pterygophobia. According to a poll by CNN and Gallup for the USA Today in March 2006, 27% of U.S. adults would be at least somewhat fearful of getting on an airplane (Stoller, 2006).

The conducted experimental protocol was carried out in a quiet controlled environment where subjects assumed the pilot's seat for roughly 25 minutes. Internal variables were unconscientiously affected by the online assessed user emotions.

The project achieved rather transversal goals as it was possible to use it as a fully functional testbed for online biometric emotion assessment through galvanic skin response, respiration rate and amplitude and phalanx temperature readings fusion and its incorporation with Russell's Circumplex Model of Affect (Russell, 1980) with success rates of around 78%. Considering the aeronautical simulation, an immersive realistic environment was achieved, with the use of 3D video eyewear.

It was found that those without fear of flying found the experience rather amusing, as virtual entertainment, while the others considered the simulation realistic enough to trigger an emotional response – verified by biometric readings.

The present document is organized as follows: in the next section a broad, detailed revision of related work is depicted; in section 3, the project is described in a global perspective but also highlighting relevant system modules; in section 4 the conducted experimental session conditions are described and in the following section the results are presented; in the final section, conclusions are drawn and future work areas revealed.

2 STATE OF THE ART

This section is divided into two subsections: the first concerning automatic emotion assessment; the second regarding aeronautical simulation tools.

2.1 Automatic Emotion Assessment

Until a recent past, researchers in the domains related to emotion assessment had very few solid ground standards both for specifying the emotional charge of stimuli and also a reasonable acceptable emotional state representation model. This issue constituted a serious obstacle for research comparison and conclusion validation. The extreme need of such metrics led to several attempts to systematize this knowledge domain.

Considering first the definition problem, Damásio states that an emotional state can be defined as a collection of responses triggered by different parts of the body or the brain through both neural and hormonal networks (Damásio, 1998). Experiments conducted with patients with brain lesions in specific areas led to the conclusion that their social behaviour was highly affective, together with the emotional responses. It is unequivocal to state that emotions are essential for humans, as they play a vital role in their everyday life: in perception, judgment and action processes (Damásio, 1994).

One of the major models of emotion representation is the Circumplex Model of Affect proposed by Russell. This is a spatial model based on dimensions of affect that are interrelated in a very methodical fashion (Russell, 1980). Affective concepts fall in a circle in the following order: pleasure, excitement, arousal, distress, displeasure, depression, sleepiness, and relaxation - see Figure 1. According to this model, there are two components of affect that exist: the first is pleasure-displeasure, the horizontal dimension of the model, and the second is arousal-sleep, the vertical dimension of the model. Therefore, it seems that any affect stimuli can be defined in terms of its valence and arousal components. The remaining variables mentioned above do not act as dimensions, but rather help to define the quadrants of the affective space. Although the existence of criticism concerning the impact different cultures in emotion expression and induction, as discussed by Altarriba (Altarriba, 2003), Russell's model is relatively immune to this issue if the stimuli are correctly defined in a rather universal form. Having this in mind, the circumplex model of affect was the emotion representation abstraction used in the proposed project.

In order to assess Russell's model components, one ought to consider what equipment solutions were to be selected, considering, simultaneously, different features such as portability, invasiveness levels, communication integration and transparency and direct economical impact.

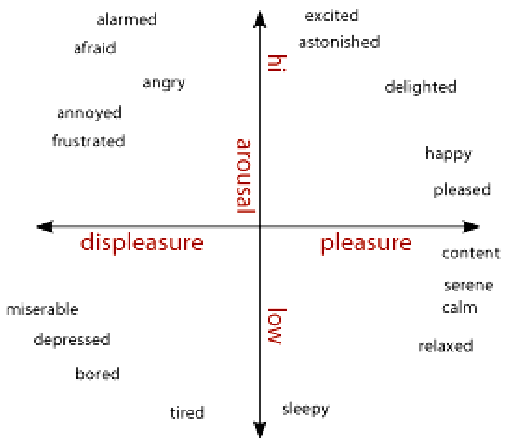


Figure 1: Russell's Circumplex Model of Affect.

Emotions assessment requires reliable and accurate communications with the subject so that the results are conclusive and the emotions correctly classified. This communication can occur through several channels and is supported by specific equipment. The invasive methods are clearly more precise, however more dangerous and will not be considered for this study. Conversely, non invasive methods such as EEG (Electroencephalography), GSR (Galvanic Skin Response), oximeter, skin temperature, ECG (Electrocardiogram), respiration sensors, amongst others have pointed the way towards gathering the advantages of low-cost equipment and non-medical environments with interesting accuracy levels (Benevoy, 2008).

Some recent studies have successfully used just EEG information for emotion assessment (Teixeira, 2008). These approaches have the great advantage of being based on non-invasive solutions, enabling its usage in general population in a non-medical environment. Encouraged by these results, the current research direction seems to be the addition of other inexpensive, non-invasive hardware to the equation. Practical examples of this are the introduction of a full set of non-invasive, low-cost sensors in several domains by Vinhas (Vinhas, 2008), Kim (Kim, 2008) and Katsis (Katsis, 2008). The usage of this kind of equipments in such diverse domains and conditions strongly suggests its high applicability and progressive migration towards quotidian handling.

For this study, the Nexus-10 hardware solution with temperature, GSR and Respiration Rate and Amplitude sensors shall be used and the data communication with the processing unit, fully described in section 3, shall be based on wireless Bluetooth technology.

2.2 Aeronautical Simulation Tools

There are two main simulator categories: Game Engines and Flight Simulators. In game engines, the most important aspect is an appealing visualization. Flight Simulators have a different approach – the main focus is on aerodynamics and flight factors present in real world, thus trying to achieve as realistic a flight as possible (Gimenes, 2008). The academic and business communities have already begun to use these cost-effective tools, benefitting from what they have to offer (Lewis, 2002).

The authors, after some consideration and analysis of available flight simulators, have chosen to use Microsoft Flight Simulator X (FSX) as the simulation environment. FSX not only provides a flexible, well-documented programming interface to interact with the environment, but also a very realistic visualization of the simulated world.

Several solutions are offered to treat pterygophobia, including medication, and some behavior therapies, together with virtual reality solutions. These are often used in conjunction with a more conventional form of therapy (Kazan, 2000), (da Costa, 2008). Though the authors are cautious regarding any conclusion about the psychological impact of this simulation tool, it is believed that this simulation, together with real-time emotional assessment, may have a positive impact in treating pterygophobia.

Summarizing, the proposed solution not only presents an online Russell's Model emotion assessment tool, based on minimal-invasive sensors, but also provides a cost-effective solution for a virtual reality simulation that can be used for treating fear of flying.

3 PROJECT DESCRITPION

This section is divided into three subsections, focusing the global architecture delineation, emotion assessment module description and aeronautical simulation component.

3.1 Global Architecture

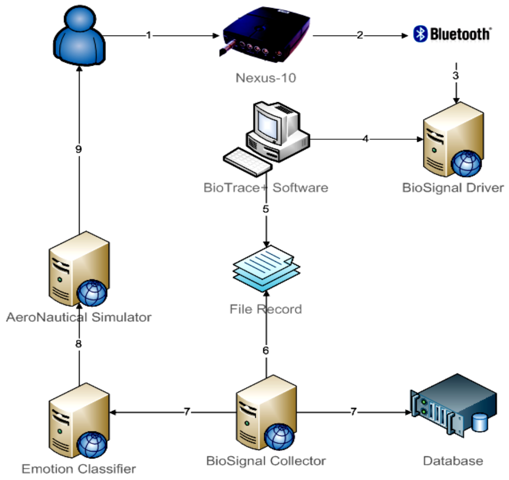


Figure 2: System's Global Architecture.

The system architecture is based on independent and distributed modules, both in logic and physical terms. As depicted in Figure 2, and following its enclosed numeration, biometric data is gathered directly from the subject by Nexus-10 hardware. In more detail, temperature, GSR and respiration sensors are used. As to reduce the number of wires presented to the user, the biometric data is transmitted by Bluetooth to a computer running the adequate data driver. The next step is of the responsibility of BioTrace+ software, and beyond providing a flexible interface for signal monitoring, it also records biometric data in a text file.

The BioSignal Collector software was developed to access the recorded data and make it available for further processing either by database access or TCP/IP socket connection. In the last case, lies the Emotion Classifier, responsible for user's emotion state assessment – this process is described below. The continuously extracted emotional states are projected into Russell's model and are filled as input to the Aeronautical Simulator. The simulation endpoint has a simple architecture. The main module communicates with the emotional endpoint, receiving data from the emotion assessment module, indicating which of the four quadrants of Russell's Model should be active. The module, in turn, communicates with FSX, changing its internal variables in order to match the desired quadrant, and as explained in section 3.3. This module also produces a permanent log file, with information collected from the simulator. The simulator interacts with the user through immersive 3D video hardware, allowing the user to control simulation visualization.

3.2 Emotion Assessment

The emotion assessment module is based on the enunciated 4-channel biometric data collected with Nexus-10 and accessed via text file readings at 10Hz sample rate – which for the analyzed features is perfectly acceptable. At the same rate, emotional states are assessed and its definition is continuously uploaded to a database for additional analysis and third-party tools access. Directly related to the aeronautical simulation, the GUI also provides an expedite method to define the session's emotional policy – force a specific quadrant, contradict or maintain the current state or tour the four scenarios. The remaining of this subsection is divided in three parts, devoted to emotion model description, calibration and data fusion, and dynamic scaling.

3.2.1 Base Emotion Model

As previously referred, the adopted emotion model was Russell's Circumplex Model of Affect. This bidimensional approach permits efficient, yet effective, online emotional assessment with none or residual historical data as it is based on single valence and arousal values. The key issue is not the determination of the subject's emotional state given a pair of valence/arousal values, but how to convert biosignals into valence/arousal pairs.

In order to anticipate the assessment of emotional data pair values, a normalization process is conducted, where both valence and arousal values are fully mapped into the [-1, 1] spectrum. With this approach, emotional states are believed to be identified by Cartesian points in a 2D environment.

3.2.2 Calibration & Channel Fusion

Having into consideration the referred normalization process, one ought to point out the importance of the calibration process. Although, the 2D point (-¾, ¾) represents a normalized defined emotional state, it can be achieved by an infinite conjugation of biosignals. This reality leads to the necessity of calibration and biometric channels fusion.

The first procedure consists in, for each subject and for each session, pinpoint directly in Russell's model, what is the predominant emotional state, through a self-assessment process. By performing this action, it is possible to define a normalized emotional baseline point. For each of the four channels taken into account for emotional state assessment an initial twenty percent variability is considered. Whenever overflow is detected, the dynamic scaling is activated as described below.

The three components were considered to have similar impact. For the valence values deviation, only galvanic skin response was considered. For this computation, the normalized baseline point is considered as reference. The conjugation of such weights determines the normalized values of arousal and valence and hence the current emotional state.

3.2.3 Dynamic Scaling

As a consequence of the emotional classification process, emerging issue concerns either biosignal readings' overflow or underflow, considering user-defined baseline and initial tolerance allowed.

To overcome this potential limitation, a fully dynamic scaling approach was considered, that consists in stretching the biometric signal scale whenever its readings go beyond the normalized interval of [-1,1]. This scale update is conducted independently for each of the analyzed biometric channels. During this process, a non-linear scale disruption is created, resulting in greater scale density towards the limit breach.

In order to better understand this approach, one shall refer to the set of formulas listed through Equation 1, depicting an overflow situation.

(a) $c_1Max = Math.Max(c_1Max, Sample[c_1Index])$

(b) $c_1ScaleUp = \dfrac{1 - baseLineNorm.Axis}{c_1Max - baseLineSample[c_1Index]}$

(c) $c = Sample[c_1Index] - baseLineSample[c_1Index]$

(d) $c_1Norm = baseLineNo\,rm.Axis + c_1ScaleUp \times c$

Equation 1: Dynamic Scaling Formulas.

First, c1 (any given biometric channel) maximum value is determined by comparing current reading with the stored value – Equation 1(a). If the limit is broken, the system recalculates the linear scale factor for values greater than the baseline neutral value, having as a direct consequence the increasing of the interval's density – Equation 1(b). Based on the new interval definition, subsequent values shall be normalized accordingly – Equation 1(c) (d). With this approach, and together with dynamic calibration and data normalization, it becomes possible for the system to perform real-time adaptations as a result of user's idiosyncrasies and signal deviations, thus assuring continuous normalized values.

3.3 Aeronautical Simulation

The desired emotional quadrant influences the simulation in three dimensions: weather, scenery and maneuvering.

The two quadrants characterized by displeasure are associated with worse climacteric conditions, ranging from thunderstorms, for the quadrant with high arousal levels, to foggy cold fronts, for the one with low levels. The two quadrants related to pleasure are coupled with fair weather, creating a more stable flight.

The chosen global scenery is an archipelago. For the two quadrants associated with high arousal levels, the itinerary takes the plane around an island, with many closed turns at low altitudes. For the two quadrants associated with low arousal levels, the path consists of an oval-shaped route around an island. The turns in this route have a superior radius and the altitude variations have smaller amplitude. As a result, the flight is experienced as a calmer one.

Closely related to the route description is maneuvering control. For the first route, typical auto-pilot controls are used, namely speed, heading and altitude controls. As for the second route, two additional features are applied – maximum bank and yaw damper, which limits the maximum roll during turns, and reduces rolling and yawing oscillations, making the flight smoother and calmer.

4 EXPERIMENTAL ACTIVITIES

The experiments were conducted using a variety of equipment, for both the biometrical emotion assessment module and the simulation module. As for the first module, sensors for skin temperature, galvanic skin response and respiration rate and amplitude were used. In order to present the user with an immersive experience, 3D video hardware was used in conjunction with the flight simulator, in the form of virtual reality video eyewear, which provides the user with a three degree of freedom head-tracker, allowing the user to experience the environment as if he was actually there.

The experiments were conducted among twenty subjects, 13 males and 7 females, aging between 21 and 56. Four of the subjects stated that they had some level of fear of flying, while the remaining declared not to.

After providing background information to characterize the sample, the subject was connected to the biometrical equipment, in order to establish an emotional baseline, as explained in section 3.2.2.

The experiment had three sequential stages. In the first, the plane takes off from an airport. After takeoff, a series of closed circuits was performed. Finally, in the landing phase, the plane lines up with the selected airport, makes the approach and lands.

After concluding the trial, the subjects described the experience, and reviewed an animation of the evolution of both simulation and emotional assessment, to confirm or refute those assessments.

5 RESULTS

The results are presented and analyzed in two main groups: emotion assessment and simulation.

In what concerns to emotion assessment, the validation model was based on user self-assessment, as previously described. These results were collected in two forms: concerning single emotions and specific regions on Russell’s model, and concerning only the four quadrants. For the first method, a success rate of 78% was achieved. For the second one, this number increases to 87%. Table 1 shows the confusion table with percentages of automatic assessment versus self-assessment for each quadrant.

Table 1: Emotion Assessment Confusion Table.

		Automatic Assessment			
		1st Quadrant	2nd Quadrant	3rd Quadrant	4th Quadrant
Users	1st Quadrant	**30,7**	1,8	0,3	1,2
	2nd Quadrant	3,1	**32,8**	1	0,1
	3rd Quadrant	0,2	1,7	**10,9**	1,2
	4th Quadrant	1	0,1	1,6	**12,3**

One additional result to consider is that the automatic emotion assessment has a lower rate of failure for opposite quadrants.

Concerning the simulation, users were asked to describe their experience, and to classify, on a scale of one to five, the level of immersiveness. The results show that the majority of the individuals considered the environment to be highly immersive, with an average classification of 4,2.

Takeoff and landing are traditionally associated with higher levels of apprehension and anxiety among passengers who suffer from pterygophobia, a fact confirmed by the experimental results. All subjects that are afraid of flying also stated that those are in fact the most stressful moments, and the collected data corroborates this fact. Figure 3 shows the average arousal levels measured during the experiments conducted among these individuals. As can be seen, higher arousal levels were registered during the initial and final stages of the simulation, which represent takeoff and landing.

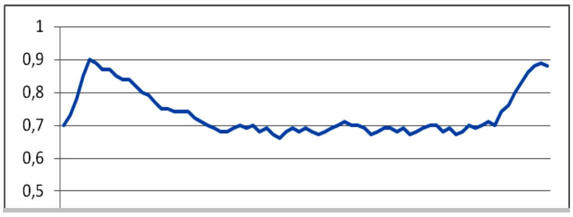


Figure 3: Average Arousal Levels During Simulation.

6 CONCLUSIONS

From an architectural standpoint, the distributed architecture with logic and physical module separation proved to be reliable and efficient. This approach enabled independence between biometric data collection, processing and simulation related computation. It also provided database collection of both raw biometric channel values and semantic emotional state information for future analysis and validation, improving system openness.

At a more significant level, the emotional assessment layer reached high accuracy levels. Through the detailed validation process, 78% of the classified emotional states were considered correct by the subjects. If simplified to Russell's four quadrants, this value reaches 87%, which supports the conclusion of an effective emotional assessment process. Still in this category, it is worth to mention the on-the-fly classification procedure that nearly suppresses the need to a long baseline data gathering and user identification as it is performed by the user at any time. Also, the dynamic scaling was valuable, as to correctly accommodate outsized signal deviations without precision loss.

In what regards the aeronautical simulation, all projected goals where completely fulfilled as users confirmed their immersion sensation, by both self-awareness and biological recorded response. It is believed that the use of 3D glasses as display device played a particularly important role in creating the appropriate environment.

Some improvement opportunities have been identified along the project. It is believed to be useful, for future system versions, to include additional biometric channels in the emotional assessment engine, such as ECG, BVP (Blood Volume Pulse) and even EEG. This signals integration would be fairly straightforward as the current data fusion process and emotional base model support that kind of enhancement. Still concerning this module, one shall mention the possibility to test Russell's model expansion to 3D by adding a dominance axis. Regarding the aeronautical simulator, it would be interesting to define and test more navigation scenarios. Still in this point, a more smooth transition between contexts, especially between quadrants characterized by high levels of arousal and those with low levels of arousal would be useful.

As a final project summary, one shall point that the proposed system has a dual application as a complete entertainment system with user emotional awareness that continuously adapts the multimedia content accordingly, and possibly a more solemn approach as a phobia treatment auxiliary.

REFERENCES

Altarriba, J., Basnight, D. M., & Canary, T. M., 2003. *Emotion representation and perception across cultures*. In W. J. Lonner, D. L. Dinnel, S. A. Hayes, & D. N. Sattler (Eds.), Readings in Psychology and Culture (Unit 4, Chapter 5), Center for Cross-Cultural Research, Western Washington University, Bellingham, Washington USA.

Benovoy, M., Cooperstock, J., Deitcher, J., 2008. *Biosignals Analysis and its Application in a Performance Setting - Towards the Development of an Emotional-Imaging Generator*. In Proceedings of the First International Conference on Biomedical Electronics and Devices, pp. 253-258.

da Costa, R., Sardinha, A., Nardi, A., 2008. *Virtual Reality Exposure in the Treatment of Fear of Flying*. Aviation, Space, and Environmental Medicine, 79(9), pp. 899–903.

Damásio, A. R., 1994. *Descartes error: Emotion, reason and human brain*. Europa-América.

Damásio, A. R., 1998. *Emotions and the Human Brain*. Iowa, USA: Department of Neurology.

Gimenes, R., Silva, D.C., Reis, L.P., Oliveira, E., 2008. *Flight Simulation Environments Applied to Agent-Based Autonomous UAVs*. In Proceedings of ICEIS 2008, pp. 243–246.

Kahan, M., Tanzer, J., Darvin, D., and Borer, F., 2000. *Virtual Reality-Assisted Cognitive-Behavioral Treatment for Fear of Flying: Acute Treatment and Follow-up*. CyberPsychology & Behavior, 3(3), pp. 387–392.

Katsis, C., Katertsidis N., Ganiatsas G. and Fotiadis D., 2008. *Towards Emotion Recognition in Car-Racing Drivers: A Biosignal Processing Approach*. IEEE Tran. on Systems, Man, and Cybernetics – Part A: Systems and Humans, Vol. 38, No 3, pp. 502-512.

Kim, J., André, E., 2008. *Multi-Channel BioSignal Analysis for Automatic Emotion Recognition*. In Proceedings of the First International Conference on Biomedical Electronics and Devices.

Lewis, M., Jacobson, J., 2002. *Game Engines in Scientific Research*. Communications of the ACM. Vol. 45, Nr. 1, pp. 27-31.

Russell, J. A., 1980. *A Circumplex Model of Affect*. In Journal of Personality and Social Psychology, 39, pp. 1161-1178.

Stoller, G., 2006. *Fear of Flying can Cripple Workers*, USA Today, March 2006. Available online at http://www.usatoday.com/educate/college/business/articles/20060326.htm. Consulted Dec. 2008.

Teixeira, J., Vinhas, V., Oliveira, E., Reis, LP., 2008. *Emotion Assessment Tool for Human-Machine Interfaces - Using EEG Data and Multimedia Stimuli Towards Emotion Classification*. In SIGMAP, pp. 185-188.

Vinhas, V., Oliveira, E., Reis, LP., 2008. *Realtime Dynamic Multimedia Storyline Based on Online Audience Biometric Information*. In KES IIMSS, pp. 545-554.

NEW TYPE OF MULTI-DEGREE-OF-FREEDOM PIEZOELECTRIC ACTUATORS, BASED ON ACTIVE KINEMATIC PAIRS

Ramutis Bansevicius, Arunas Lipnickas
Kaunas University of Technology, The Mechatronics Center for Research, Studies and Information
Kestucio 27, Kaunas, Lithuania
{ramutis.bansevicius, arunas.lipnickas}@ktu.lt

Minvydas Ragulskis
Kaunas University of Technology, Research Group for Mathematical and Numerical Analysis of Dynamical Systems
Studentu 50-222, Kaunas, Lithuania
minvydas.ragulskis@ktu.lt

Keywords: Mechatronics, Piezoelectric Actuators, Active Kinematic, Active Bearing, Modelling.

Abstract: New type of multi-degree-of-freedom piezoelectric actuators based on active kinematic pairs is presented. The contact zone of this type of actuator is formed by two oscillating transducers in a form of rod, plate, disk or cylinder. Depending on a phase of both transducers in a contact zone and their amplitudes, either high frequency oblique impacts or periodic change of normal reaction in the contact zone are generated, leading to continuous motion of one of the links. Schematics of piezoelectric motors, using two active elements in the contact zone and comprising the number of degrees-of-freedom up to 5 are presented. Several applications for laser beam deflection and positioning devices in the plane are considered. The concept of active bearing is introduced; this type of support has no processing datum surface errors.

1 INTRODUCTION

It is worthwhile to introduce the concept of *active kinematic pair* in the design of multi-degree-of-freedom actuators and 3*D* positioning systems. Such concept is especially useful in the design of adaptive positioning systems (Ragulskis, 1988; Bansevicius, 2002a). The characteristic feature of active kinematic pair is that one or both elements of it are manufactured from active or smart materials such as piezoelectric, magnetostrictic or shape memory materials (Bansevicius, 2002a). Active kinematic pair can change its kinematic structure or parameters depending on external conditions or excitation characteristics (Bansevicius, 2002b). The multi-functionality of the mechanisms can be achieved applying direct or inverse piezoelectric effects. In other words, several different functionalities as motion generation, measurement of parameters of motion, control of friction forces in the contact zone can be implemented into one instrument (Bansevicius, 2000a; Ko and Kimb, 2006, Chu and Fan, 2006). Excitation of static or quasi-static deformations, multi-directional and multi-shape resonance oscillations, generation of motion in the contact zone, transformation of oscillations into continuous motion are just several examples of application of active kinematic pairs (Bansevicius, 2000b; Bansevicius and Ahmed, 2000ab; Bansevicius, 2001).

Active kinematic pairs enable:

- Control of the number of degrees-of-freedom of the kinematic pair by means of friction force control in the contact zone or generation static or quasi-static deformations of the element of the pair;
- Generate forces and moments in the contact zones;
- To effect additional functions – self-diagnostics, multi-functionality, adaptively, self-assembly;
- To implement two levels of degree-of-freedom. The first level comprises large deflections or displacements, produced by transformation of

resonance oscillations of pair's links into continuous motion. The second level deals with small displacements (in nanometre range), implemented by means of direct piezoelectric effect and specific sectioning of electrodes.

2 TWO ACTIVE ELEMENTS IN THE CONTACT ZONE

Applying two active elements in the contact zone of piezoactuator enables enlargement of the generated force or torque and transforming oscillations into continuous motion. High frequency oblique impacts are generated in the contact zone between two rod type transducers (Fig. 1). Only longitudinal resonance oscillations are generated in these rods. Specific phase differences between the oscillations in these rods enable variation of the continuous motion parameters. Zero phase difference generates direct continuous motion, 180 degree phase difference enables reverse continuous motion. Phase difference between 0 and 90 degrees (or between 180 and 270 degrees) changes the normal force component in the contact zone and helps to tune dynamical parameters of the whole system with the rheological parameters of the contact zone.

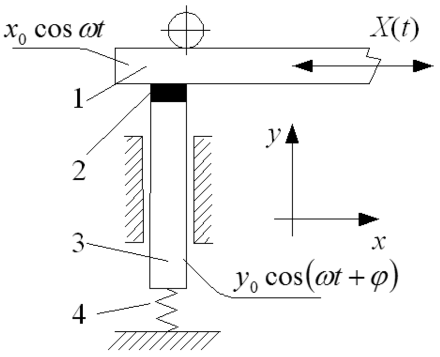


Figure 1: Piezoelectric motor. *1* – the first piezoelectric transducer (slider); *2* – contact element; *3* – the second piezoelectric transducer (active support); *4* – spring.

Several schematics of such piezomotors are presented in Figures 2, 4 and 6. Symmetric scheme (Fig. 2) enables sufficient increase of the generated force. Combination of different Eigen modes helps to achieve larger deflections or displacements of the sliding element. It must be noted that the reverse mode is symmetric in all schemes and can be realised by altering the phase of one of the transducers by 180 degrees. The scheme presented in Fig. 2b is implemented in the design of miniature longitudinal small stroke (2 mm) piezomotor and well illustrates the technological advantages of such an approach.

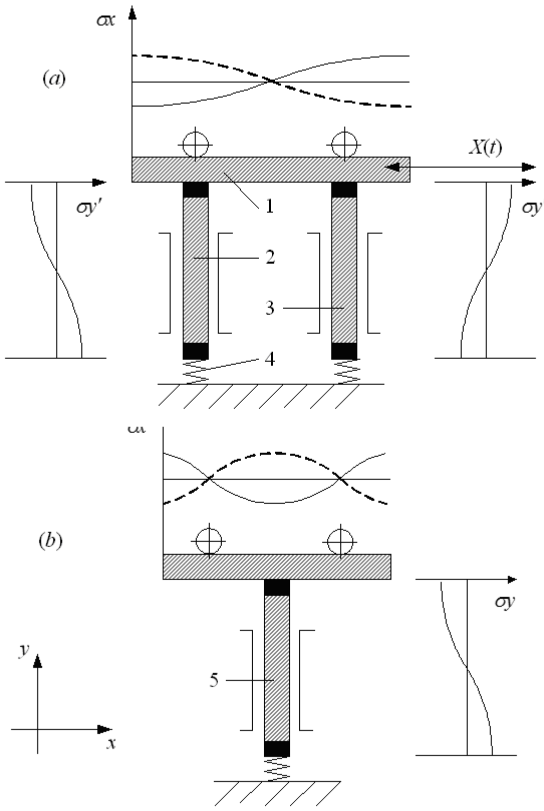


Figure 2: Two cases of piezoelectric motors with two active elements in the contact zone: *(a)* - symmetric scheme; only longitudinal first Eigen shape resonance vibrations are generated in all actuators (the node is located in the middle point of the actuator); *(b)* - application of different Eigen shapes to increase the stroke: longitudinal resonance oscillations (second Eigen shape) are excited in the slider (two nodal points); longitudinal resonance oscillations (first Eigen shape) are excited in the supporting actuator (one nodal point).

2.1 Numerical Analysis of Active Kinematic Pair comprising Two Active Elements in the Contact Zone

The active kinematic pair shown in Fig. 1 is modelled using finite element techniques (Zienkiewitcz and Taylor, 1991). We used hybrid elastic body – piezoelectric material finite element formulations, described in (Ragulskis, 1998). Non-adaptive uniform finite element meshes were used for the slider and the active support in order to secure the best stability and convergence of the numerical solution, while an adaptive mesh was used for the contact element (Fig. 1).

The effect of the pressing spring was assumed as constant forces, acting to nodes of the outside surface of the contact element. Limiters for the slider were modelled as kinematic constrains for the nodes of the external surface of the slider, permitting longitudinal but impeding transverse displacements.

Eigen shapes of the slider and the contacting element were calculated. Then the electrical excitation of the piezoelectric material was selected in such a way that the oscillations of the slider and the active support element would follow their Eigen shapes (and resonance Eigen frequencies) as close as possible.

The time step was selected to accommodate accurate integration of the fifteenth Eigen mode (the first fifteen Eigen modes were sufficient to represent complex dynamical processes taking even in the contact zone). Schematically, the piezoelectric excitation can be represented as:

$$
\begin{aligned}
F_S(i,j) &= \Phi_S(\delta x_i, \delta y_j)\sin(\omega t) \\
F_A(i,j) &= \Phi_A(\delta x_i, \delta y_j)\sin(\omega t + \varphi)
\end{aligned}
\qquad (1)
$$

where F_S is a matrix of dynamical forces acting to the nodes of the slider; F_A are forced acting to the nodes of the active support; Φ_S and Φ_A are first Eigen shapes of the slider and the active support (Eigen shape depends from the allocation of electrodes); ω is the resonance frequency (both for the slider and the active support); φ is the phase difference of the electric excitation. Numerical simulations produce a large dataset of results. Only the trajectory of the middle contact point of the slider is shown in Fig. 3 in the *x*-*y* phase plane in order to represent main dynamical features.

It can be seen that optimal longitudinal motion of the slider is achieved at φ = 0, or φ = 180 (the reverse mode). Other phase differences result into chattering mode when the oblique impact energy is not optimally transferred into the continuous longitudinal motion of the slider.

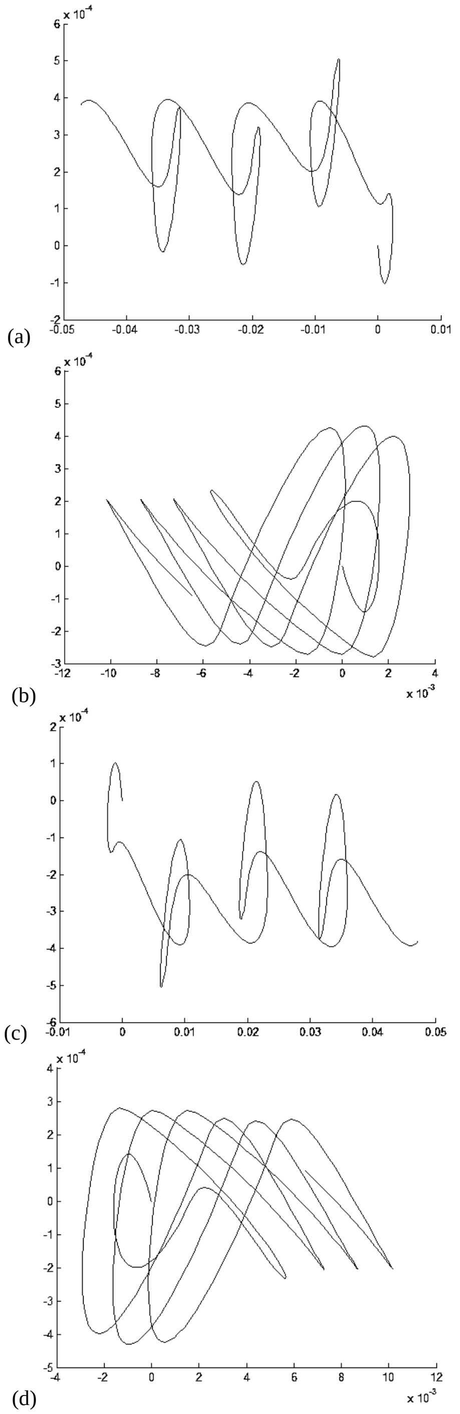


Figure 3: Numerically reconstructed transient dynamical processes taking place at different phases: (a) φ = 0; (b) φ = 90; (c) φ = 180; (d) φ = 270 degrees. Note different scales of *x*-axis.

3 APPLICATIONS OF MULTI-DEGREE-OF-FREEDOM PIEZOELECTRIC ACTUATORS

Piezoelectric actuators based on active kinematic pairs enable realisation of different types of motion. Schematic of piezoelectric rotary motor is presented in Fig. 4. Dynamical processes taking place in the contact zone are analogous to the basic type mechanism shown in Fig. 1 and depend on the phase difference and rheological properties of the contact surfaces. As in the previous schemes, the resonance frequencies of the radial oscillations of the disk type transducer and longitudinal oscillations of the rod piezotransducer can differ in the range of few percents. In fact, the range of tolerable differences depends on the damping in the transducers and in the whole system in general.

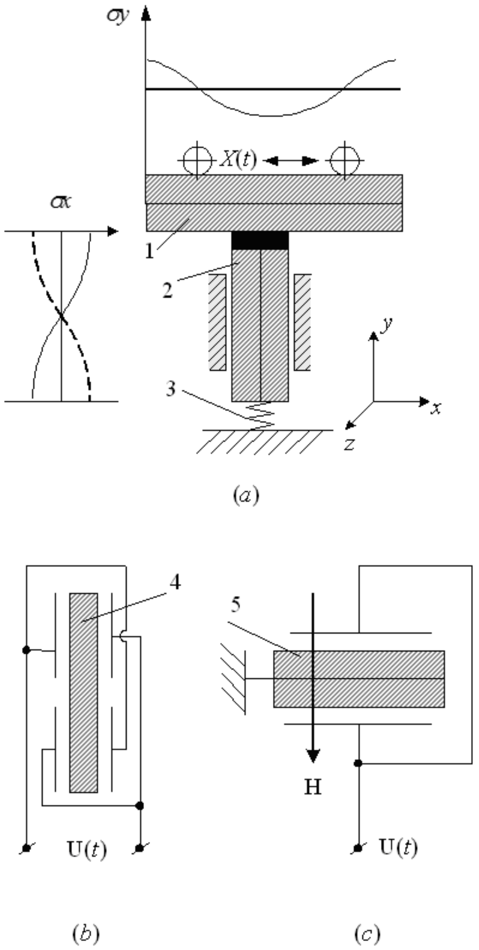


Figure 4: Increasing the velocity by using bimorphic transducers. *1*, *2* – bimorphic piezotransducers; *3* – spring; 4 – schematics of the electrodes; *5* – excitation wiring diagram; *H* – poling vector.

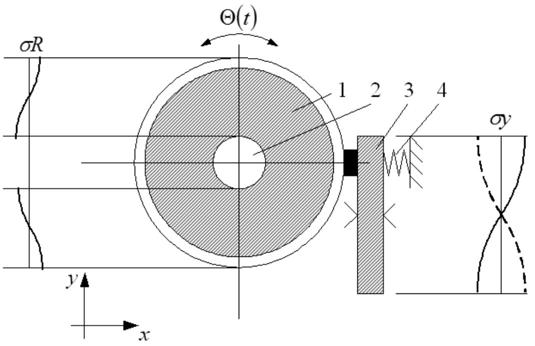


Figure 5: Schematics of angular motion piezomotors with increased torque. *1* – piezoceramic ring; *2* – fixing element; *3* – rod type piezotransducer; *4* – spring.

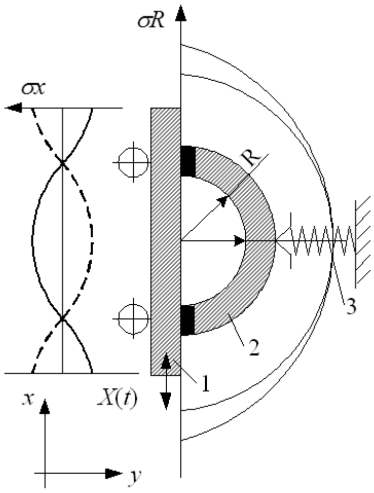


Figure 6: Schematic diagram of translational motion piezomotor analogous to symmetric scheme presented in Fig. 4.

Two new instrumentation schemes shown in Fig. 7 and Fig. 8 have been implemented practically. Initial experiments have shown their effectiveness for implementation small one-degree-of-freedom positioning systems dedicated for precision instrumentation. Such schemes can be easily manufactured using low cost piezotransducers.

Piezomotors with two active elements in the contact zone can be effectively applied in the design of optical beam reflection and scanning equipment. Two cases of such instrumentation are presented in Fig. 9 and 10. Oblique high frequency impacts are generated in the contact zone between the cylinder section and rod type transducer (Fig. 9).

The scheme showed in Fig. 10 enables implementation of two-degrees-of-freedom motion of the mirror by electric control of different disk or plate type piezotransducers' electrodes.

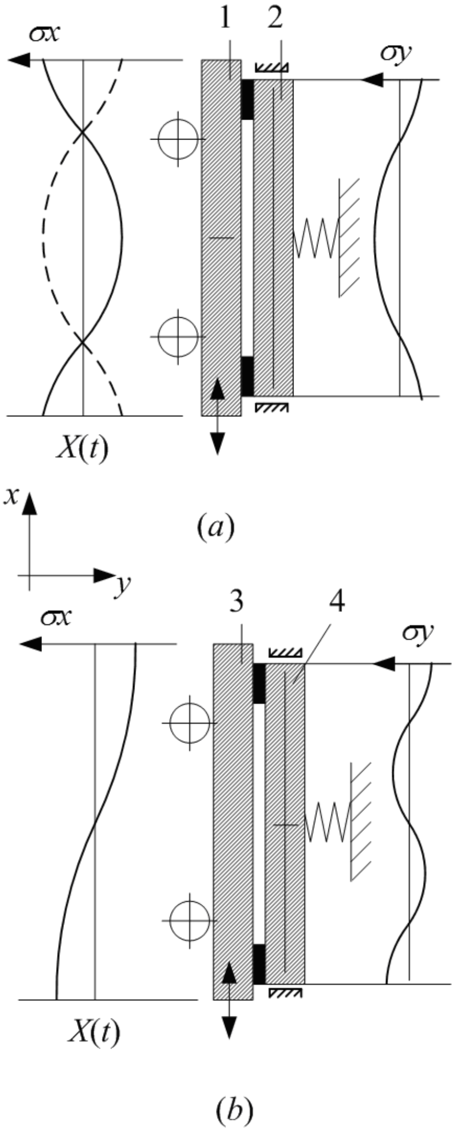


Figure 7: Schematics of translational motion piezomotors with two active elements in the contact zone. *(a): 1* – piezoceramic transducer (longitudinal resonance second Eigen shape oscillations; two nodal points); *2* – piezoceramic bimorphic bending resonance oscillation transducer (first Eigen shape; two nodal points). *(b): 3* – longitudinal vibration transducer (first Eigen shape; one nodal point); *4* – bending vibration transducer (second Eigen shape; three nodal points).

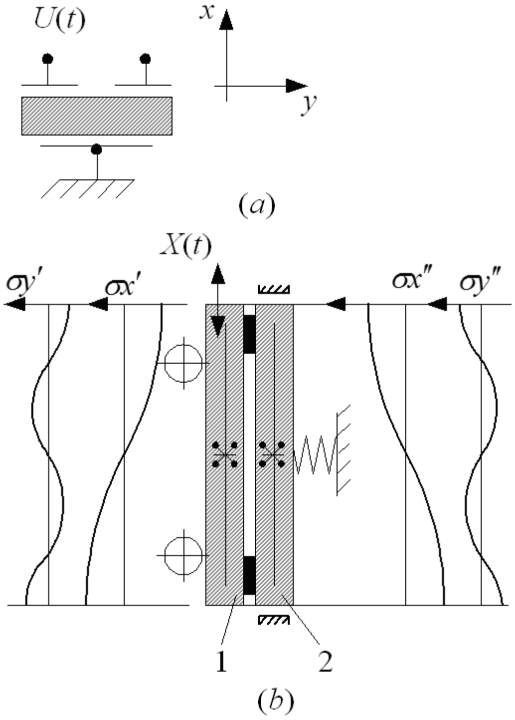


Figure 8: Symmetric translational motion piezomotor with two active elements in the contact zone. *(a):* sectioning of the electrodes; *(b): 1, 2* –longitudinal resonance oscillations (first Eigen shape) and in-plane bending oscillations (second Eigen shape) are generated in two identical piezotransducers.

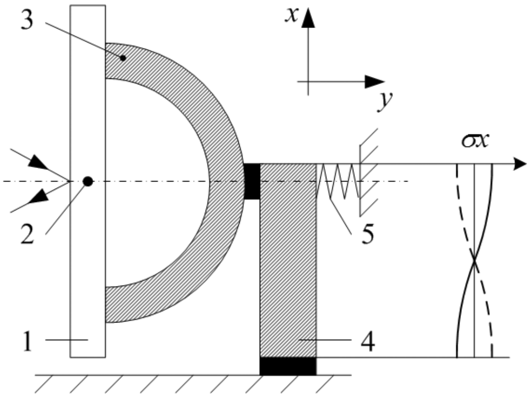


Figure 9: Optical beam reflector. *1* – mirror; *2* – fixing elements; *3* – piezoceramic sector; *4* – piezoelectric rod, generating longitudinal resonance first Eigen shape oscillations.

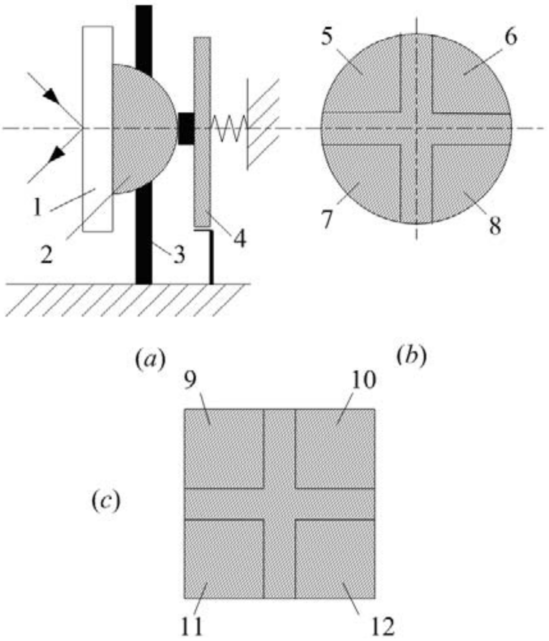


Figure 10: Two-degree-of-freedom optical beam reflector/scanner. *1* – mirror; *2* – segment of spherical piezotransducer; *3* – fixing element (scheme is not specified); *4* – plate type piezotransducer; *5…12* – sectioned electrodes.

4 CONCLUSIONS

New type of multi-degree-of-freedom piezoelectric actuators, based on active kinematic pairs, is presented in this paper. Schematics of piezoelectric motors, using two active elements in the contact zone and comprising the number of degrees-of-freedom up to 5 are presented. Several applications for laser beam deflection and positioning devices in the plane are described. The concept of active bearing is introduced.

The contact zones of these actuators are formed by oscillating pairs of piezoelectric transducers. Control of the phase difference between the transducers enable transformation of oblique impacts into continuous motion. Such types of actuators are characterised by high resolution, low time constant, and are applicable in different areas of precision mechatronics.

ACKNOWLEDGEMENTS

Research has been financed by Lithuanian State Science and Studies Foundation (High-Tech Development program’s research project PjezoAdapt).

REFERENCES

Ragulskis, K., et al., 1988, Vibromotors for Precision Microrobots, Hemisphere Publishing Corp., New York, 310 p.

Bansevicius, R. et al., 2002a, *The Mechatronics Handbook*, CRC Press, Boca Raton.

Bansevicius, R., 2002b, Latest trends in the development of piezoelectric multi-degree-of-freedom actuators/ sensors. Responsive systems for active vibration control, NATO Science Series, Vol. 85, Kluwer Academic Publishers, Dordrecht, p.207-238.

Bansevicius, R., et al., B., 2000a, Design and control of precise robots with active and passive kinematic pairs, Mechatronics 2000: Proceedings of the 7-th Mechatronics Forum, Atlanta, Georgia, U.S.A., Pergamon, p.10-16.

Ko, H-P., et al., 2006, A novelty in ultrasonic linear motor using the radial mode of a bimorph, Sensors and Actuators A 125, p.477–481.

Chu, C-L. and Fan, S-H., 2006, A novel long-travel piezoelectric-driven linear nanopositioning stage, Precision Engineering 30, p.85–95.

Bansevicius, R., et al., 2000b, Active error compensation for precise machines, IMEKO 2000: XVI IMEKO World Congress Proceedings, Vienna, Vol. 8, p.21-26.

Bansevicius, R. and Ahmed, S., 2000a, Piezoelectric Active Supports, U.K. Patent 2313982, Patented by Smart Technology Ltd., U.K..

Bansevicius, R. and Ahmed, S., 2000b, Piezoelectric Actuators, U.K. Patent 2332090, Patented by Smart Technology Ltd., U.K..

Bansevicius, R. and Ahmed, S., 2001, Bearings and Support, USA Patent 62625114B1, Patented by Smart Technology Ltd., U.K..

Zienkiewicz, O.C. and Taylor, R.L. 1991, Finite Element Method – Solid and Fluid Mechanics: Dynamics and Nonlinearity, Vol. 2, McGraw-Hill: New York.

TWO-STAGE ALGORITHM FOR PATH PLANNING PROBLEM WITH OBSTACLE AVOIDANCE

Mustafa Dogan and Nizami Gasilov
Baskent University, Faculty of Engineering, Eskisehir Yolu 20. km, Baglica, 06810 Ankara, Turkey
mustafa.dogan@gmail.com, gasilov@baskent.edu.tr

Keywords: Path planning, Obstacle avoidance, Graph theory, Dijkstra’s algorithm.

Abstract: The path-planning problem is considered for mobile robot inside environment with motionless circular obstacles in different sizes. The robot is expected to reach a given target by following the shortest path and avoiding the obstacles. The two-stage algorithm is proposed to solve the problem numerically. In the first stage a line-arc based path is found by using geometric techniques. This path cannot be minimal. However, its length can be used to restrict search space to an ellipse, which contains the minimal path. Thus, the reduced search space makes the next stage more efficient and endurable for real-time applications. In the second stage of the algorithm, by discretization of the restricted elliptic region the problem results in finding the shortest path in a graph and is solved by using the Dijkstra’s algorithm. The proposed two-stage algorithm is verified with numerical simulations. The results show that the proposed algorithm is successful for obtaining an optimal solution. The applicability of the proposed algorithm is validated by practical experiment.

1 INTRODUCTION

Various methods have been proposed for the solution of obstacle avoidance problem. One of the real-time methods that has been developed for navigation of mobile robots is potential field approach (Connoly *et al.*, 1991; Rimon and Koditschek, 1992). The main advantage of this method is on-line efficiency as a result of the integration of the low-level robot control and path planning. However, its main disadvantage is that in some cases it could not escape from local minima that result in abnormal termination without reaching the target. Harmonic potential functions (Connoly *et al.*, 1991) and navigation functions (Rimon and Koditschek, 1992) are proposed to overcome these difficulties and in this way obstacle avoidance is succeeded, but optimal path finding cannot be achieved. Besides, navigation functions are difficult to calculate and impossible to be implemented in real-time, especially for robots that have many degrees of freedom (Kavraki *et al.*, 1996). Furthermore, navigation functions should be differentiable by the definition and therefore, they can cause problems in piece-wise continuous or saturated robot control applications (Rimon and Koditschek, 1992). Nevertheless, potential field method is improved by the recent advances in both theoretical and application aspects, e.g. 3-D extension (Chuang, 1998; Chuang and Ahuja, 1998; Ren *et al.*, 2006; Cowan, 2007).

Probabilistic roadmap for path planning is just another alternative method (Kavraki *et al.*, 1996, 1998). This method, in comparison with the previous ones, can be more reliable and applicable in more general cases. On the other hand, theoretic analysis becomes more complex, which is an important disadvantage of this method.

Some other efficient shortest-path algorithms for mobile robots are also proposed based on graph theory approach (Helgason *et al.*, 2001; Liu and Arimoto, 1992). Finally, dynamic programming (Hamilton-Jacobi-Bellman) methods are used extensively as well (Dreyfus, 1965; Moskalenko, 1967, Sundar and Shiller, 1997). For example, in (Sundar and Shiller, 1997), near-optimal solutions for the shortest path problem have been obtained by applying the geometric approach efficiently. The disadvantage of this method is that it lacks the minimal path in some cases.

In this research, inspired by the last three approaches (Helgason *et al.*, 2001; Liu and Arimoto, 1992; Sundar and Shiller, 1997) a new two-stage optimization algorithm is developed. At the first stage, near-optimal solution is provided by geometric incremental approach, and this solution is used to describe the elliptic region that contains the shortest path. Thus, the optimal solution can be readily searched after the completion of the first stage by Dijkstra's algorithm.

Different from the graph-based heuristic algorithms, e.g. A* (Dechter and Pearl, 1985; Hart *et al.*, 1968; Hart *et al.*, 1972; Nilsson, 1980; Bruce and Veloso, 2006), the proposed method does guarantee that the selected path is optimal. Furthermore, A* algorithm can result in the much longer path than Dijkstra's one, depending upon crucial choice of the heuristic function and world configuration.

The most important novelty of this work is that the initial search space is reduced a lot in order to find the shortest path efficiently. Therefore, two main disadvantages of Dijkstra's algorithm, namely large computational burden and difficulty with following the discrete paths (Helgason *et al.*, 2001), have been overcome by search space reduction and greedy path construction approach that explained in Section 4. These two properties are indispensable in real-time applications.

2 PROBLEM DEFINITION

Suppose, motionless circular obstacles located in rectangular domain (search space) are given in finite number. It is assumed that no obstacle cuts or touches any other obstacle. The motivating question behind this research is how point robot can navigate on the shortest path from a given starting point *S* to a given target position *F* with obstacle avoidance.

Note that, the condition about point robot is not a restriction for the problem. Let the robot be circular with radius ρ. If we enlarge all obstacles in the amount of ρ radius-wise, then the robot itself can be considered as point robot.

Also note that, the proposed approach can be easily extended for the case when other types of obstacles such as ellipses, convex polygons are considered together with circles.

Two-stage algorithm is proposed for numeric solution to the problem. The detailed explanations of these stages are given in the following sections.

3 INCREMENTAL METHOD BASED ON GEOMETRY

The method applied at the first stage is incremental since it is optimal just for one step. The method is realized by using geometric representations. The first obstacle on the straight line between the current position of the object and the target is assumed to be a single obstacle in each step of the method. In accordance with this, the tangential path is determined firstly from the initial point to this obstacle. Besides, extra obstacles are controlled whether they intersect the path or not. If not (refer to Section 3.1), this path is used to reach the obstacle. Then, the path is followed along the boundary of the obstacle until the point, where tangent from the target touches the obstacle. This point becomes the new starting point for the next step. If there are extra obstacles across the tangential path that connects the initial point *S* and the first obstacle (refer to Section 3.2), then the extra obstacle that is closest to *S* will be determined. This extra obstacle is reached along the tangential path closer to the baseline *SF* and avoided by following its boundary. Then, arrival point is determined as new starting point for next step. This process will be iterated until no obstacle on the way to target.

3.1 Single Obstacle Avoidance

Assume that on the path *SF*, there is only one circular obstacle with the radius *r* and centered at *C* as represented in Figure 1. Two pairs of tangent lines from points *S* and *F* can be drawn to the circle. We can choose the ones that have minimum angle with line *SF*, i.e. SA_1 and FB_1 in the figure. Therefore, according to geometrical rules the shortest path consists of line SA_1, arc A_1B_1 and line B_1F.

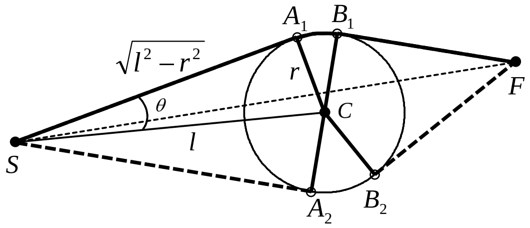


Figure 1: Optimal avoidance of a single obstacle.

In order to calculate coordinates of points A_1 and A_2, the following equation can be used:

$$\mathbf{SA} = \frac{\sqrt{l^2 - r^2}}{l} P_{\pm\theta}\ \mathbf{SC}$$

Where **SA** and **SC** are vectors; $\theta = \arcsin(r/l)$; P_θ is the rotation operator about point *S* through an angle θ. The sign of θ corresponds to choosing one of the points A_1 and A_2.

One of them, which is the closest point to baseline *SF*, is selected, either A_1 or A_2.

We can make a significant evaluation for proving convergence of approximate method based on geometry. Since circular obstacle centered at *C* crosses the line *SF*, we have: $d < r$ (Fig. 2). Hence

$$|SF| = |SH| + |HF| \ge |HF| = \sqrt{|CF|^2 - d^2} >$$
$$> \sqrt{|CF|^2 - r^2} = |BF| \quad \Rightarrow \quad |SF| > |BF|.$$

According to last inequality, direct distance to the target decreases by avoiding an obstacle.

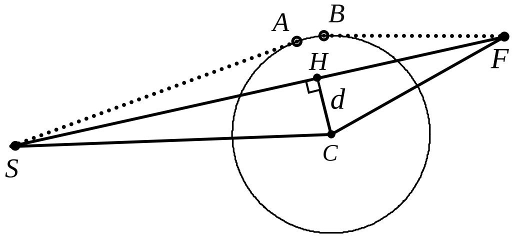


Figure 2: Schematic representation to prove that the direct distance to the target decreases with obstacle avoidance.

3.2 Extra Obstacle Avoidance

There could be some extra obstacles across the tangential path *SA* that is mentioned in Section 3.1. It is represented in Figure 3 how the path can be constructed in this case. In Figure 3, the obstacle centered at *C* is ordinary one on the path *SF*, and the obstacle centered at *E* is the extra one.

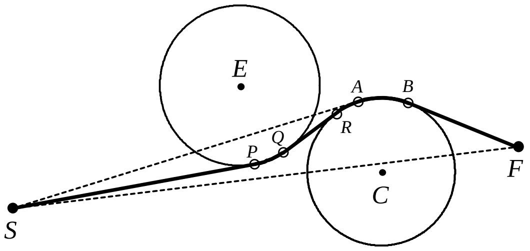


Figure 3: Avoidance of an extra obstacle.

The algorithm implemented for extra obstacle avoidance is explained briefly below.

Among extra obstacles crossing tangential path *SA* the obstacle that is the closest one to the base point *S* is determined, i.e. the obstacle centered at *E* in Figure 3. Direction *SF* will be our reference to avoid this obstacle. Tangential path *SP* close to line *SF* is determined. Subsequently, *QR*, common cross tangent of obstacles *E* and *C* with end point *Q* close to *P*, is calculated. The obstacle *E* has been avoided by following tangent line *SP* first, and then arc *PQ*. Then the question is considered whether there is any other extra obstacle on path *QR*, or not. If not, then by following tangent line *QR* and arc *RB* the ordinary obstacle *C* will be avoided. If there is an extra obstacle, new iteration on avoidance of extra obstacle is started with taking *Q* as the new initial point.

Since number of the obstacles is finite, extra obstacles will be eliminated after finite number of steps and an ordinary obstacle will be avoided next. Refer to end of the Section 3.1, the evaluations prove that direct distance to the target decreases by avoiding ordinary obstacle. There is finite number of obstacles by assumption and the distance to the target diminishes at each step, then approximate method based on geometry is convergent.

Geometric method implemented at the first stage of the main algorithm results in near-optimal solutions. The path obtained through this method might not be optimal. Such an example is given in Figure 4.

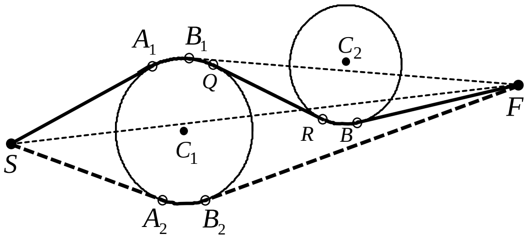


Figure 4: An example for which the path obtained by the geometric method is not optimal.

We can see how the method works for this example below. Circle C_1 is the first ordinary obstacle across the path *SF*. According to Section 3.1 this obstacle will be avoided following tangent path SA_1 closer to baseline *SF* and then arc A_1B_1. Taking B_1 as the new starting point, the next step of the method will be initiated. Circle C_2 is determined as the ordinary obstacle across path B_1F. In order to avoid it, the tangent, which is closer to the baseline B_1F, is calculated. This tangent line crosses C_1. Thus, in this time the circle C_1 becomes extra obstacle when ordinary obstacle C_2 is avoided. The procedure described in Section 3.2 is implemented to avoid the obstacle C_1. Since the starting point B_1 lies on C_1, the step to reach the extra obstacle will be eliminated. Only arc B_1Q is used to avoid C_1 (Here *Q* is the end point of *QR*, common cross tangent of circles). At the last iteration of the method, by following tangent line *QR* and arc *RB*, avoidance of the ordinary obstacle C_2 will be completed and by tangent path *BF* the target will be reached. Thus, the path calculated on proposed geometric method is *SA*$_1$*QRBF*. As it can be easily seen from Figure 4, this path is longer than the path *SA*$_2$*B*$_2$*F*, and consequently, is not optimal.

Thus, in general, solutions obtained through geometric method are only near-optimal. To find the optimal path the second stage of algorithm is applied, which is explained in the next section.

4 OPTIMAL PATH BY DIJKSTRA'S ALGORITHM

As it is mentioned above, the path obtained at the first stage might not be the optimal one. However, its length L_1^* gives an upper bound for optimal path length L^* such that $L^* \le L_1^*$.

The feasible region that contains the optimal path can be reduced with this inequality on purpose.

Let *X* be a point on optimal path. Then it can be claimed that $L^* = L_{SF}^* = L_{SX}^* + L_{XF}^*$.

Since the shortest path should be a line segment with no consideration for obstacles the following inequalities can be written: $|SX| \le L_{SX}^*$ and $|XF| \le L_{XF}^*$. Thus, we get $|SX| + |XF| \le L_1^*$.

Regarding this inequality, sum of distances from *S* and *F* to a point *X* lying in the feasible region cannot exceed the value L_1^*. Subsequently, the feasible region is inside the ellipse with focuses at *S* and *F*. Hence, based upon the value L_1^* the feasible region can be diminished and restricted to an ellipse. Thus, the reduced search space makes the second stage much more efficient and endurable for real-time applications.

In this stage, coordinate transformation is applied such that new origin will be the midpoint *M* of the line segment *SF*, and the new horizontal axis will be in the direction of ray *MF*. In this new coordinate system, the feasible region can be described simply as follows:

$$(x/a)^2 + (y/b)^2 \le 1$$

where $a = L_1^*/2$ and $b = \sqrt{(L_1^*)^2 - |SF|^2}/2$. In the mean time, changing the coordinate system is also beneficial such that the realizations of the following steps will be more efficient.

Discretization of the problem is the next step. For this purpose, a grid with equal squares is created over the region. The side length of a square, *h*, is complied with the minimum distance between obstacles, δ, such that $h \le \delta/3$. Intersection points of the grid, or nodes, are assigned as graph vertices. Thus the analyzed problem can be solved by graph theory approach. We can define two prohibited cases such that a) If the vertex *N* is out of feasible region, or b) If the square with side length *h* and centered at *N* intersects an obstacle. In both cases, the vertex *N* is marked as forbidden to pass. Graph edges can be constructed in two alternative ways such that:

1) 8-neigborhood vertices around any vertex *V*, which is not prohibited, are examined one by one. The edge is added between the vertex *V* and the one, which is permitted to pass.

2) All pairs of vertices *(U, V)* are to be examined one by one. If the vertices of a pair *(U, V)* are not prohibited and line segment *UV* does not intersect any obstacle, the edge with the length |*UV*| is constructed between *U* and *V*.

At the first alternative, discrete approach is also used to construct the edges. Therefore, the total number of edges is minimal and edge structure is easy to process. In the second alternative, which can also be characterized as greedy approach, edge structure is difficult to implement. However, it provides solution closer to the optimal solution than the first one does. In simulations, the results of which are represented in the next section, the second approach is applied.

Thus, the solution of the problem is reduced only to find the shortest path from vertex *S* to vertex *F* in the obtained graph. This new problem is solved by applying Dijkstra's algorithm (Anderson, 2004). Furthermore, some improvements have been done based upon the properties of the problem in order to make the Dijkstra's algorithm application more efficient. For instance, forbidden vertices are not included to the set of graph vertices. Let v be the number of graph vertices. If the first alternative mentioned above is realized then instead of weight matrix of size $v \times v$ a zero-one (or binary) matrix of size $8 \times v$ is used. Hence, this approach is suitable for real-time applications. For the second alternative, as weight matrix is symmetric, then only lower triangle matrix can be stored at memory.

Note, that in the first alternative the graph is sparse (number of edges $e \sim 8v$). In this case the complexity of Dijkstra's algorithm, implemented with a binary heap, is $O(e \log v) \sim O(v \log v)$.

5 SIMULATIONS RESULTS

The proposed two-stage algorithm is verified by many simulations. In simulations the obstacles are chosen randomly in a rectangular region. The target position is selected. Then the proposed algorithm is executed for different starting positions.

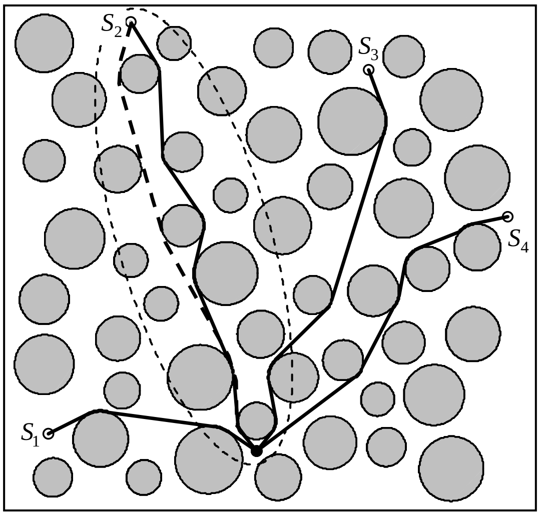


Figure 5: Near-optimal (solid lines) and optimal paths (dashed line) obtained from calculations in presence of 50 obstacles. Thin dashed line represents the boundary of elliptic feasible region used at the second stage of the algorithm.

The results of one simulation are represented in Figure 5. Here we take scene with size of $a \times b = 120 \times 120$ (unit length can be assigned arbitrarily). We randomly generate circles (x_c, y_c, r_c) with radius $r_c \in [4, 8]$. If next candidate circle don’t intersects an existing one, we add this circle to the list of obstacles. Otherwise the candidate one is rejected.

The paths that are obtained by the first stage have been represented with solid-line. For one of the starting points, (S_2), optimal path by the second stage has been shown as dashed-line in Figure 5. This optimal path has essential differences in comparison with the result of the first stage (solid-line starting from S_2). For other cases (S_1, S_3, S_4), the optimal paths, obtained at the second stage, have not been represented for the purpose of clarity of the figure, since they do not differ a lot from drawn ones.

For the case with starting point S_2, the boundary of feasible region, used at the second stage, is shown by an ellipse (thin dashed-line) in Figure 5. This ellipse envelops an area, which is about 1/5 of the whole search space (rectangle). Since the operation complexity of Dijkstra’s algorithm is $O(v^2)$ and v is proportional to covered area, the benefit of proposed algorithm is about 25 times better than the algorithm applied to whole region.

It has been verified by simulations that the proposed algorithm is useful to solve the optimization problem for obstacle avoidance. According to obtained results, in some cases only the first stage of the algorithm can be sufficiently used, especially considering robotic applications that require essential time and memory resources.

Although the proposed algorithm works well for circular obstacles, more efficient approximations for obstacles can be obtained by implementing the other convex figures, e.g. rectangles and ellipses. Therefore, the geometric method can be extended easily to cover these shapes. Fortunately, the second stage of the algorithm is independent from the shapes of obstacles.

6 EXPERIMENTAL RESULTS

Pioneer 3-DX mobile robot, which has embedded computer with C++ based ARIA (Advanced Robotics Interface for Applications) software and wireless communication capability, has been controlled by remote PC. Driving capabilities of the robot are 2-wheel drive, plus rear balancing caster with differential steering.

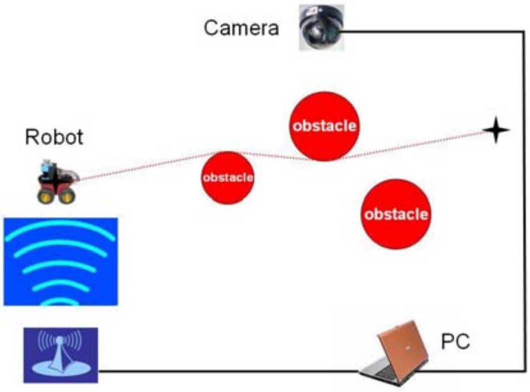


Figure 6: Experimental setup.

As shown in Figure 6, after extensive image processing, necessary path planning commands are produced by the proposed algorithm that all running at PC, and are transmitted through wireless network to the robot. Obstacles are chosen as circular shaped disks. Besides, their positions are selected in accordance with robot dimensions (swing radius is 32 cm), and minimum inter-distance requirements, see Section 4.

The first preliminary experiments are done with two obstacles to evolve the implementation. Finally, the last experiment is done with five obstacles. After image processing, the algorithm is implemented in an efficient way. At the end, the robot followed the prescribed path successfully as planned beforehand.

For future work, automatic identification and setting the robot orientation and pose will be an important achievement, since it took time to set the right orientation for the robot. Integrating both stages of the algorithm with image processing to work in real time while obeying the dynamic constraints will complete this research project.

7 CONCLUSIONS

Optimization problem for obstacle avoidance on the plane has been investigated. Two-stage algorithm has been proposed for solution to the problem and tested successfully with experiments. In the first stage, near-optimal solution is obtained through geometric approach. Using this solution, the feasible region is restricted to an ellipse. At the second stage the problem is reformulated as the shortest path problem in graph, and optimal solution is found by applying Dijkstra's algorithm in the reduced search space. Consequently, two main contributions of this research come out clearly at the last stage. The first one, the solution is optimal, and the second one, it is obtained through an efficient way with a significant reduction of search space. Simulation results have proved that the two-stage algorithm complies with theory and produces accurate solutions.

REFERENCES

Connoly, C. I., Burns, J. B., and Weiss, R., 1991. Path planning using Laplace's equation. *Proc. IEEE Int. Conf. Robot. Automat.*, vol. 1, pp. 2102-2106.

Rimon, E., and Koditschek, D. E., 1992. Exact robot navigation using artificial potential functions. *IEEE Transactions on Robotics and Automation*, vol. 8, pp. 501-518.

Kavraki, L. E., Kolountzakis, M. L., and Latombe, J. C., 1998. Analysis of probabilistic roadmaps for path planning. *IEEE Transactions on Robotics and Automation*, vol. 14, no. 1, pp. 166-171

Kavraki, L. E., Svestka, P., Latombe, J. C., and Overmars, M. H., 1996. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, 566-580.

Helgason, R. V., Kennington, J. L., and Lewis, K. R., 2001. Cruise missile mission planning: a heuristic algorithm for automatic path generation, *Journal of Heuristics*, Kluwer Academic Publishers, vol. 7, pp. 473-494.

Liu, Y. H., and Arimoto, S., 1992. Path planning using a tangent graph for mobile robots among polygonal and curved obstacles. *Int. J. Robot. Res.*, vol. 11, no. 4, pp. 376-382.

Dreyfus, S., 1965. *Dynamic Programming and the Calculus of Variations*. New York, London: Academic.

Moskalenko, A. I., 1967. Bellman equations for optimal processes with constraints on the phase coordinates. *Autom. Remote Cont.* (Translation of *Avtomatika i Telemekhanika*), vol. 4, pp. 1853-1864.

Sundar, S., and Shiller, Z., 1997. Optimal obstacle avoidance based on the Hamilton-Jacobi-Bellman equation. *IEEE Transactions on Robotics and Automation*, vol. 13, no. 2, pp. 305-310.

Anderson, J. A., 2004. *Discrete Mathematics with Combinatorics*, 2nd Ed. Prentice Hall.

Dechter, R., and Pearl, J., 1985. Generalized best-first search strategies and the optimality of A*. *Journal of the ACM***,** vol. 32, pp. 505 – 536.

Hart, P. E., Nilsson, N. J., and Raphael, B., 1968. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, vol. SSC-4, no. 2, pp. 100–107.

Hart, P. E., Nilsson, N. J., and Raphael, B., 1972. Correction to a formal basis for the heuristic determination of minimum cost paths. *SIGART Newsletter*, no. 37, pp. 28-29.

Nilsson, N. J., 1980. *Principles of artificial intelligence*. Palo Alto: Tioga.

Bruce, J. R., and Veloso, M. M., 2006. Safe multi-robot navigation within dynamic constraints. *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1398-1411.

Chuang, J.-H., 1998. Potential-based modeling of three-dimensional workspace for obstacle avoidance. *IEEE Transactions on Robotics and Automation*, vol. 14, no. 5, pp. 778-785.

Chuang, J.-H., and Ahuja, N., 1998. An analytically tractable potential field model of free space and its application in obstacle avoidance. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, vol. 28, no. 5, pp. 729-736.

Ren, J., McIsaac, K. A., and Patel, R. V., 2006. Modified Newton's method applied to potential field-based navigation for mobile robots. *IEEE Transactions on Robotics*, vol. 22, no. 2, pp. 384-391.

Cowan, N. J., 2007. Navigation functions on cross product spaces. *IEEE Transactions on Automatic Control*, vol. 52, no. 7, pp. 1297-1302.

DIFFERENTIAL-DRIVE STEERING SYSTEM USING PLANETARY GEARING FOR OMNIDIRECTIONAL MOBILE ROBOT

Hideo Kitagawa
Department of Electronic Control Engineering, Gifu National College of Technology
Kamimakuwa, Motosu, Gifu, Japan
hkita@gifu-nct.ac.jp

Takashi Ohno, Yuki Ueno, Kazuhiko Terashima
Department of Production Systems Engineering, Toyohashi University of Technology
Hibarigaoka, Tempaku, Toyohashi, Japan
{*ueno, tearsima*}*@syscon.pse.tut.ac.jp*

Keywords: Mobile robots, Movement, Robot kinematics, Vehicles, Wheels, Ride comfort.

Abstract: Holonomic omnidirectional mobile robot is useful with its high mobility in narrow or crowded area, and omnidirectional robot equipped with normal tires is desired for difference excess, vibration suppression and ride comfort. Caster-drive mechanism using normal tire has been developed to realize a holonomic omnidiredctional robot, however, there remains some problems. This paper presents effective systems to control the caster-drive wheels of omnidirectional mobile robot. Differential-Drive Steering System (DDSS) using planetary gearing is proposed to improve the operation ratio of motors. DDSS generates driving and steering torque effectively from two motors. Simulation results show the proposed system is effective for holonomic omnidirectional mobile robots.

1 INTRODUCTION

An omnidirectional robot is highly maneuverable in narrow or crowded area including residences, offices, warehouses and hospitals. It can be applied to an autonomous mobile robot in a factory, a wheelchair and so on. Several kinds of omnidirectional mobile robots and their applications have been developed by (West, 1992), (Pin, 1994), (Damoto, 2002) and (Kitagawa, 2008). However, these robots realized their omnidirectional motion by using special wheels such as mechanum wheels, ball wheels, omni-disks and omni-wheels.

To improve the ride comfort, vibration suppression, slippage reduction and ability of difference excess, omnidirectional robots equipped with normal tires have been strongly required. (Arai, 1981) proposed an omnidirectional vehicle equipped with normal tires. However, it was a non-holonomic vehicle which has to adjust the direction of wheels before changing the moving direction of vehicle.

Holonomic omnidirectional vehicles, which can move in any direction without changing the direction of tires beforehand, equipped with normal tires include dual-wheel type by (Wada, 2000) and caster-drive(active-caster) type by (Wada, 1996).

The dual-wheel type has problems as follows. Number of wheels is limited to two, and it is impossible to get high friction or to adapt a rough terrain by synchronous drive of many wheels. Moreover, a passive wheel is needed to stabilize the posture of vehicle.

The caster-drive wheel has offset between the steering axis and the center of wheel. The wheel can move in any direction by controlling the steering axis and the driving wheel independently by using two motors. A holonomic omnidirectional motion of a robot can be realized by using two or more caster-drive wheels.

However, the caster-drive wheel has a problem as follows. When the vehicle is in steady motion including straight motion and rotation with constant curvature, only the driving motor works and the steering motor becomes idle. When the vehicle changes its moving direction, high load is applied to the steering motor. Therefore, high power is required both for the driving and steering motor. It causes increase of mass.

The aim of our research is to develop a holonomic omnidirectional mobile robot with caster-drive wheel minimizing the motor power by using the interference of output of two motors. New gearing mechanism is proposed to realize the interference.

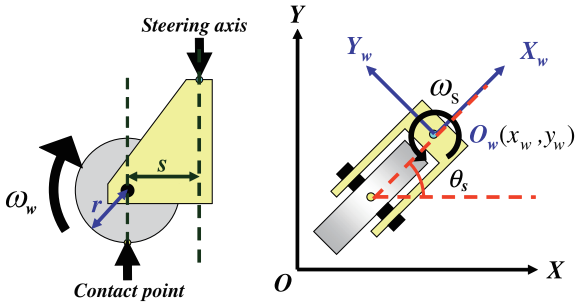


Figure 1: Caster-drive wheel.

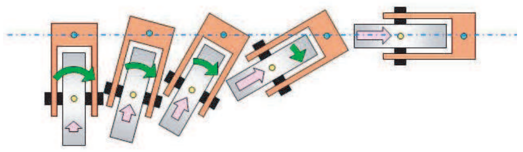


Figure 2: Lateral motion to right.

2 PRINCIPLE

2.1 Omnidirectional Motion using Caster-Drive Wheel

The principle of holonomic omnidirectional motion using caster-drive wheels is described in this section. Figure 1 shows the caster-drive wheel. The position and orientation of wheel can be represented by the position $O_w(x_w, y_w)$ of steering axis and the orientation θ_s from the contact point between the wheel and the ground to the steering axis with reference to the fixed frame $O-XY$.

By rotating the driving wheel with the angular velocity ω_w, velocity $\dot{x}_w = r\omega_w$ generates in the direction of X_w axis. Here, r is the radius of driving wheel. By rotating the steering axis with the angular velocity ω_s, velocity $\dot{y}_w = -s\omega_s$ would generate at the center of wheel in the direction of Y_w axis. Here, s is the offset between the steering axis and the center of driving wheel in the direction of X_w. However, reacting velocity $\dot{y}_w = s\omega_s$ generates at the steering axis in the direction of Y_w axis, because the position of the driving wheel is fixed by the friction with the ground. Therefore, the velocity ($\dot{x}_w$, $\dot{y}_w$) of caster-drive wheel can be controlled by changing ω_w and ω_s.

Figure 2 shows an example of motion. The initial orientation θ_s of the wheel is set to be $\theta_s = \pi/2$ in the frame $O-XY$. The motion as shown in Fig. 2 can be given by changing ω_w and ω_s appropriately. Even though the rotating wheel itself can not generate lateral motion to the right, the lateral motion of the robot, which is fixed to the steering axis, is realized.

Each wheel does not have to control the orientation of the robot by itself.

The direct kinematic equation is denoted by the state vector $x_w = [x_w, y_w]^T$ and the input vector $u_w = [\omega_w, \omega_s]^T$ as

$$\dot{x}_w = B_w u_w, \tag{1}$$

where

$$B_w = \begin{bmatrix} r\cos\theta_s & -s\sin\theta_s \\ r\sin\theta_s & s\cos\theta_s \end{bmatrix}. \tag{2}$$

The inverse kinamatic equation becomes

$$u_w = B_w^{-1}\dot{x}_w, \tag{3}$$

where

$$B_w^{-1} = \begin{bmatrix} \frac{1}{r}\cos\theta_s & \frac{1}{r}\sin\theta_s \\ -\frac{1}{s}\sin\theta_s & \frac{1}{s}\cos\theta_s \end{bmatrix}. \tag{4}$$

Holonomic omnidirectional motion ($\dot{x}$, $\dot{y}$, $\dot{\theta}$) of a mobile robot can be achieved by using two caster-drive wheels or more. Furthermore, synchronous drive with arbitrary number of wheels and rotation mechanism yields three dimensional holonomic omnidirectional motion by three motors.

2.2 Differential-Drive Steering System (DDSS)

In this section, we develop a useful method for constructing a caster-drive wheel using Differential-Drive Steering System (DDSS). The DDSS outputs driving and steering velocities from two motors using differential-drive gearing.

Figure 3 shows the principle of the DDSS. Just like a usual planetary gearing, it is composed of sun gear(A), outer ring gear(B), planet gear(C) and planet carrier(D). The planet carrier(D) holds the planet gear(C) and rotates relative to the sun gear(A) and the outer ring gear(B). However, unlike usual planetary gearing, the DDSS is 2-input 2-output system without fixing any component. A and B are independently driven by two motors. C and D provide output torque.

Figure 4 shows the mechanism of the DDSS. D, which is fixed to the chassis(E), provides the steering torque, and C, which leads to the driving wheel via the bevel gear, provides the driving torque.

Let ω_A, ω_B, ω_C and ω_D be the angular velocity of A, B, C and D in Fig. 3, and Z_A, Z_B and Z_C be the number of teeth of A, B and C, respectively.

When $\omega_D = 0$, the steering angular velocity ω_s becomes zero, and we obtain

$$\omega_A = -\frac{Z_B}{Z_A}\omega_B = -\frac{Z_C}{Z_A}\omega_C, \tag{5}$$

$$\omega_D = 0. \tag{6}$$

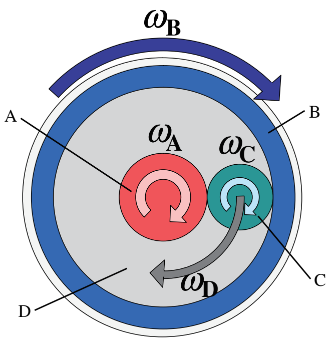


Figure 3: Principle of Differential-Drive Steering System (DDSS).

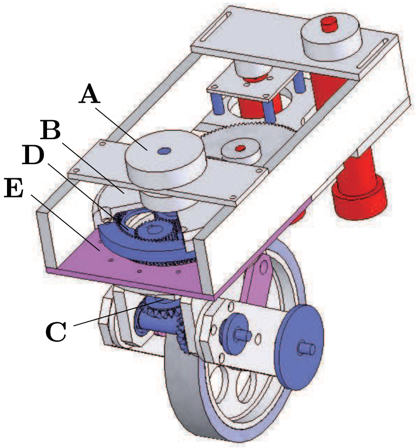


Figure 4: Mechanism of DDSS.

When $\omega_C - \omega_D = 0$, the driving angular velocity ω_w becomes zero because C does not rotate between A and B, and we obtain

$$\omega_A = \omega_B = \omega_C = \omega_D. \tag{7}$$

The direct kinematic equation, which derives driving and steering output $u_w = [\omega_w, \omega_s]^T$ from motor input $u_P = [\omega_A, \omega_B]^T$, can be described as

$$u_w = \begin{bmatrix} \omega_C - \omega_D \\ \omega_D \end{bmatrix} = B_P u_P, \tag{8}$$

where

$$B_P = \begin{bmatrix} -\frac{Z_A Z_B}{Z_C(Z_A+Z_B)} & \frac{Z_A Z_B}{Z_C(Z_A+Z_B)} \\ \frac{Z_A}{Z_A+Z_B} & \frac{Z_B}{Z_A+Z_B} \end{bmatrix}. \tag{9}$$

The inverse kinematic equation becomes

$$u_P = B_P^{-1} u_w, \tag{10}$$

where

$$B_P^{-1} = \begin{bmatrix} -\frac{Z_C}{Z_A} & 1 \\ \frac{Z_C}{Z_B} & 1 \end{bmatrix}. \tag{11}$$

Next, we derive the motor power ratio of the DDSS. Joint torques T_A, T_B, T_C and T_D of A, B, C and D, respectively, are given by

$$\begin{bmatrix} T_C \\ T_D \end{bmatrix} = \begin{bmatrix} -\frac{Z_C}{Z_A} & \frac{Z_C}{Z_B} \\ \frac{Z_A+Z_C}{Z_A} & \frac{Z_A+Z_C}{Z_B} \end{bmatrix} \begin{bmatrix} T_A \\ T_B \end{bmatrix}, \tag{12}$$

where positive direction of each torque is same as that of angular velocity in Fig. 3.

For an omnidirectional mobile robot with the DDSS, steady motion including straight motion and rotation with constant curvature is achieved by $\omega_s (= \omega_D) = 0$. When $\omega_s = 0$ $(T_D = 0)$, the joint torques are given from (12) by

$$\begin{aligned} T_A &= -\frac{Z_A}{Z_B} T_B, && (13) \\ T_C &= -\frac{2Z_C}{Z_A} T_A, && (14) \\ T_D &= 0. && (15) \end{aligned}$$

The power ratio of two motors is given from (5) and (13) by

$$\begin{aligned} P_A : P_B &= T_A\omega_A : T_B\omega_B \\ &= \left(-\frac{Z_A}{Z_B} T_B\right)\left(-\frac{Z_C}{Z_A}\omega_C\right) : T_B \frac{Z_C}{Z_B}\omega_C \\ &= 1:1 && (16) \end{aligned}$$

On the other hand, when $\omega_w = 0$ $(T_C = 0)$, the joint torques are given from (12) by

$$\begin{aligned} T_A &= \frac{Z_A}{Z_B} T_B, && (17) \\ T_C &= 0, && (18) \\ T_D &= \frac{2(Z_A+Z_C)}{Z_A} T_A. && (19) \end{aligned}$$

The power ratio is given from (7) and (17) by

$$\begin{aligned} P_A : P_B &= T_A\omega_A : T_B\omega_B \\ &= \frac{Z_A}{Z_B} T_B\omega_B : T_B\omega_B \\ &= Z_A : Z_B. && (20) \end{aligned}$$

Letting the diameter of sun gear and planet gear be the same in Fig. 1 yields

$$P_A : P_B = 1:3. \tag{21}$$

2.3 Operation Ratio of Motors

In this section, we discuss the operation ratio of motors by comparing the DDSS to a conventional caster-drive wheel. We define the operation ratio δ of motors as

$$\delta = \frac{(Sum\ of\ motor\ power\ in\ motion)}{(Sum\ of\ rated\ power\ of\ motors)}. \tag{22}$$

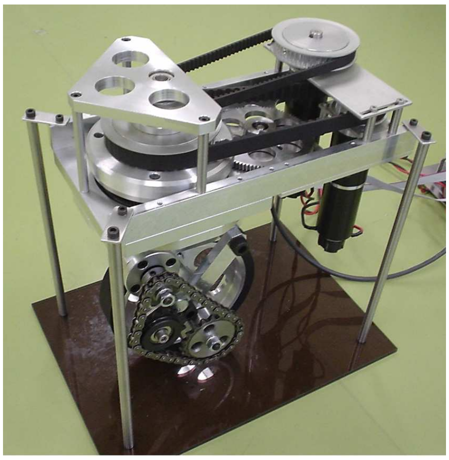


Figure 5: Prototype DDSS.

The ratio ($P_{A0} : P_{B0}$) of rated power of two motors used in the DDSS is set to be 1:1. The ratio of rated power used in conventional method is also set to be 1:1 as denoted in (Wada, 1996).

We calculate the operation ratio δ in case of driving motion ($T_D = 0$). Let P be the sum of motor output power needed to achieve the motion. The result of the conventional method is $\delta = \frac{P}{P_{A0}+P_{B0}} = 0.5$ from $P_{A0} = P_{B0} = P$. The result of the DDSS is $\delta = \frac{P}{P_{A0}+P_{B0}} = 1$ from $P_{A0} = P_{B0} = \frac{P}{2}$.

Next, we calculate δ in case of steering motion ($T_C = 0$). The result of conventional method is $P_{A0} = P_{B0} = P$ and $\delta = 0.5$. The result of the DDSS is $P_{A0} = P_{B0} = \frac{3}{4}P$ and $\delta = 0.67$, because $P_{B0} = \frac{3}{4}P$ from (21).

The output power of motors can be decreased by using the DDSS as a caster-drive wheel because of its high operation ratio of motors. It means that the size of robot become smaller by using the DDSS.

3 CONSTRUCTION OF OMNIDIRECTIONAL MOBILE ROBOT

We constructed a prototype model of the DDSS to check the availability of proposed mechanism as shown in Fig. 5. Torques of two motors are transmitted to the sun gear(A) and the outer ring gear(B) by the timing belts, and driving and steering torques are generated. Effectiveness of the proposed DDSS was confirmed by this apparatus.

Figure 6 and Table 1 show picture and specification of an omnidirectional mobile robot with four DDSS wheels, respectively. The proposed omnidirectional robot has capability of climbing slope of 10 [deg], accelerating 0.5 [m/s^2] and exceeding difference of 60 [mm] with carrying load of 100 [kg].

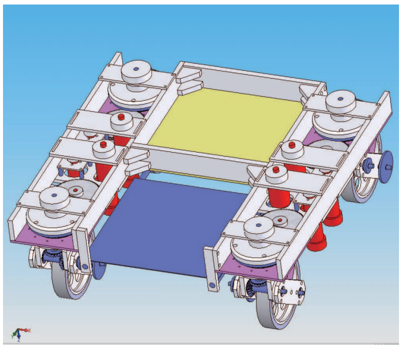


Figure 6: Omnidirectional Mobile Robot.

Table 1: Specification of Omnidirectional Mobile Robot.

Size (D × W)	600 × 600 [mm]
Weight	70 [kg]
Motor power	150 [W] × 8
Max. velocity	6 [km/h]
Max. acceleration	0.5 [m/s^2]
Max. slope angle	10 [deg]
Max. step difference	60 [mm]
Max. loading weight	100 [kg]

4 SIMULATION

4.1 Simulation Method

To show the performance of the DDSS, simulations are conducted. The simulation model is constructed by SolidWorks and DADS. The radius r of the wheel and the offset s shown in Fig. 1 were given as $r = 80$ [mm] and $s = 50$ [mm], respectively.

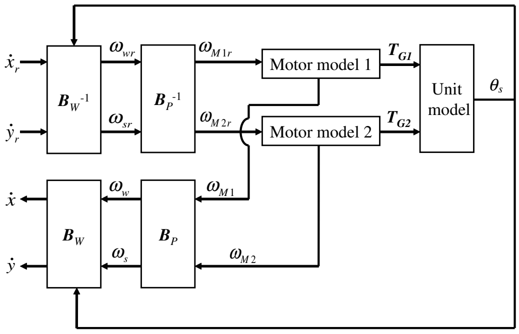


Figure 7: Control system of DDSS.

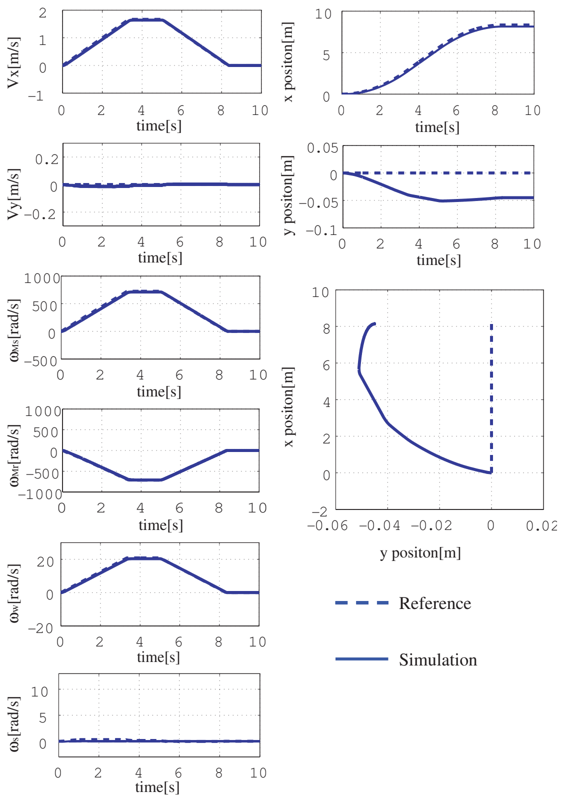


Figure 8: Simulation results (Vx = 6 km/h, Vy = 0 km/h).

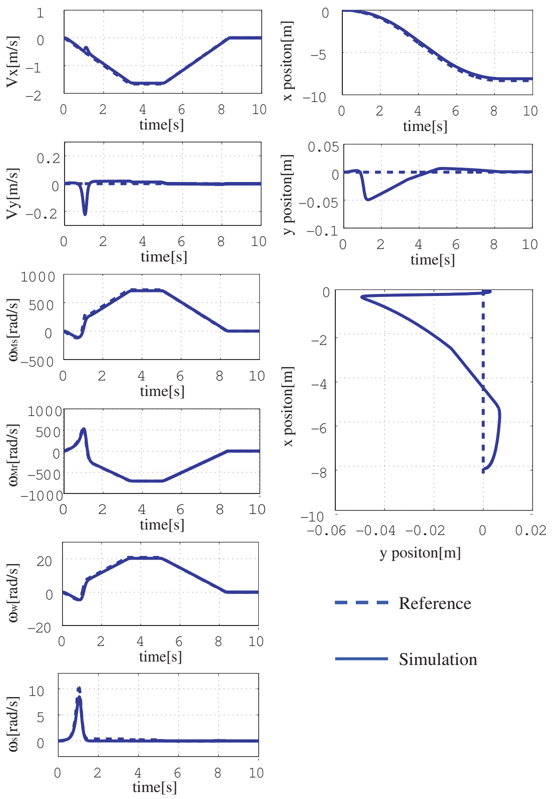


Figure 9: Simulation results (Vx = – 6 km/h,Vy = 0 km/h).

Figure 7 shows the control system of a wheel used in this simulation. The reference angular velocities ω_{M1r} and ω_{M2r} are given by (3) and (10). The velocity $(\dot{x}, \dot{y})$ of the DDSS is given from the angular velocities ω_{M1} and ω_{M2} and the steering angle θ_s.

Four translational motions toward +X, –X, +Y and –Y direction with maximum velocity of 6 [km/h] (= 1.67 [m/s]) and maximum acceleration of 0.5 [m/s^2] were examined. The initial value of θ_s was set to be zero in any case.

4.2 Simulation Results and Discussion

Figures 8 through 11 show the simulation results of motions toward +X, –X, +Y and –Y direction, respectively, from the initial state $\theta_s = 0$. Here, V_x, V_y, ω_{Ms}, ω_{Mr}, ω_w and ω_s indicate the velocity in X-direction, the velocity in Y-direction, the angular velocity of sun gear motor, the angular velocity of outer ring gear motor, the angular velocity of wheel and the angular velocity of steering axis, respectively.

As seen from these graphs, error from the reference trajectory is within 6 [cm] in any case. We can also see the feature of caster-drive wheel. In Fig. 9, the angular velocity ω_s of steering axis becomes large in the beginning of motion because the moving direction is opposite to the initial direction $\theta_s = 0$.

5 CONCLUSIONS

We proposed Differential-Drive Steering System (DDSS) for caster-drive wheel of holonomic omnidirectional mobile robot. The DDSS can provide high operation ratio of motors rather than conventional caster-drive wheel. Numerical analysis, examination by prototype model and simulation results showed effectiveness of the DDSS.

Future works include

- construction of an omnidirectional mobile robot and experiments,
- posture control on rough terrain,
- application to an omnidirectional wheelchair.

ACKNOWLEDGMENTS

This work was supported by Scientific Grant-in-Aid (No.19560271) and G-COE (Global Center of Excel-

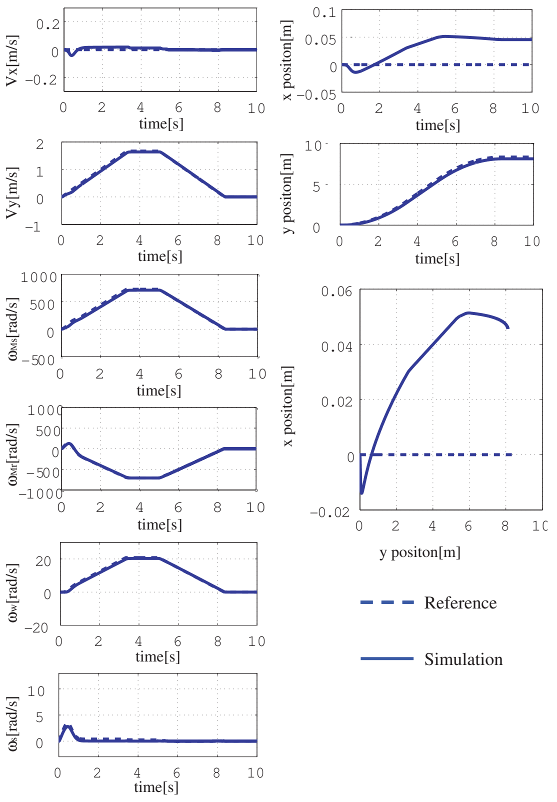


Figure 10: Simulation results (Vx = 0 km/h, Vy = 6 km/h).

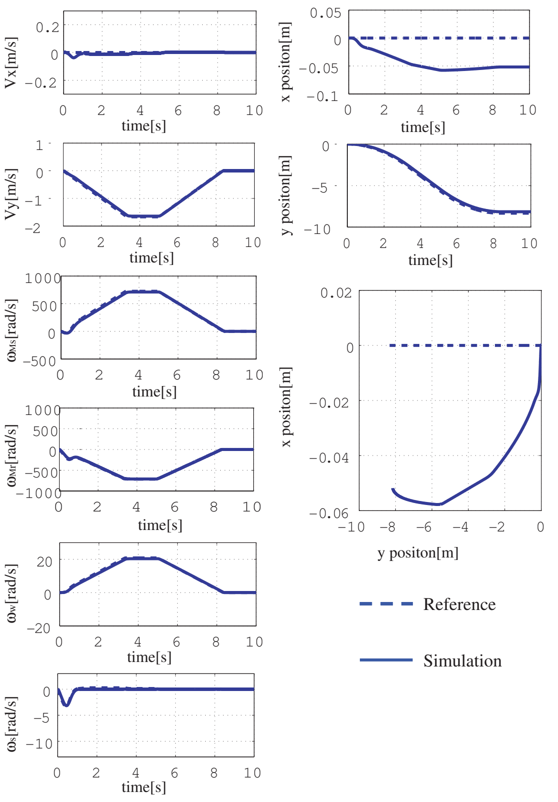


Figure 11: Simulation results (Vx = 0 km/h, Vy = – 6 km/h).

lence) Program “Frontier of Intelligent Human Sensing” by the Japanese Ministry of Education, Culture, Sports, Science and Technology.

REFERENCES

Arai, T., Nakano, E., Hashino, H., and Yamaba, K., 1981. The Control and Application of “Omni-Directional Vehicle (ODV)”. *Proc. 8th IFAC World Congress*, 1855–1860.

Damoto, R and Hirose, S., 2002. Development of Holonomic Omnidirectinal Vehicle “Vuton-II” with Omni-Discs. *Journal of Robotics and Mechatronics*, Vol.14, No.2, 186–192.

Kitagawa, H., Miyoshi, T., and Terashima, K., 2008. Skill-Assist Control of Omnidirectional Wheelchair Using Human-Friendly Interface. *Proc. IEEE Int. Conf. on Robotics and Biomimetics*, TuPA7.13.

Pin, F. G. and Killough, S. M., 1994. A new family of omnidirectional and holonomic wheeled platforms for mobile robots. *IEEE Trans. on Robotics and Automation*, vol.10, 480–489.

Wada, M. and Mori, S., 1996. Holonomic and omnidirectional vehicle with conventional tires. *Proc. IEEE Int. Conf. on Robotics and Automation*, 3671–3676.

Wada, M., Takagi, A., and Mori, S., 2000. A Mobile Platform with a Dual-wheel Caster-drive Mechanism for Holonomic and Omnidirectional Mobile Robots. *Journal of Robotics Society of Japan*, Vol.18, No.8, 1166–1172.

West, M. and Asada, H., 1992. Design of a holonomic omnidirectional vehicle. *Proc. IEEE Int. Conf. on Robotics and Automation*, 97–103.

EVALUATION OF SUN POSITION USING THE PHOTOVOLTAIC GENERATION

An Application for Attitude Estimation in Box-Shape Satellites

Ronilson Rocha, Alexandre José Ferreira, Caio Pequeno Gouvêia and Talita Bueno Barbosa
School of Mines - Department of Engineering of Control and Automation
Federal University of Ouro Preto, Campus Morro do Cruzeiro, 35400-000, Ouro Preto, MG, Brazil
rocha@em.ufop.br, {aleengaut,caiopeqgou, talita_bueno}@yahoo.com.br

Keywords: Artificial Satellites, Attitude Determination, Photovoltaic System.

Abstract: It is necessary at least two or more independent known vectors for attitude determination of an artificial satellite. One of these vectors can be the Earth Magnetic Field, which is used as reference in navigation for centuries. Other reference can be the sun, which is easily distinguished of other astronomical objects in the proximities of the Earth. While the Earth Magnetic Field can be measured by a small solid-state three-axis magneto-resistive transducer, the own photovoltaic system of the artificial satellite can provide a evaluation of the azimuthal position of the sun. This paper presents a way to estimate the sun position from the own satellite power system. This information is used for attitude determination, allowing the integration of two important subsystems of an artificial satellite.

1 INTRODUCTION

An artificial satellite is a component of a spatial system, an ample set of elements in the space (other satellites, spacecrafts, spatial stations, etc.) and in the ground (tracking stations, antennas, control centers, etc.). For the realization of a spatial mission, an artificial satellite transports several onboard equipments, such as radars, antennas, telescopes, photographic cameras, equipments for scientific measures, etc. The power supply for onboard equipments has vital importance in an artificial satellite. Considering characteristics such as modularity, cost, maintenance and life, the photovoltaic generation is the more adequate energy resource for spatial applications. Since the light-generated power in photovoltaic arrays is highly dependent of the intensity and direction of the incident sunlight, it is considerably variable in a satellite and must be conditioned and regulated by power electronic converters to supply onboard equipments.

The onboard equipments of an artificial satellite must be pointed to a specified aim to interact with others elements of the spatial system. Thus, the determination of the spatial position and orientation of an artificial satellite, also known as attitude, is fundamental for its perfect operation. The attitude related to an inertial reference is mathematically represented by an operator that rotates a vector inside of a coordinate system, which can be estimated using algorithms that require the observation of at least two independent and known vectors (Shuster and Oh, 1981). Several known vectors can be used as reference for attitude determination. One of these vectors is the Geomagnetic Field, a magnetic dipole aligned along the Earth's rotational axis which points toward to magnetic north. Since the sun is easily recognized by any object near of the Earth due to its relatively small apparent radius with a high brightness in relation to other astronomical bodies, its azimuthal position related to artificial satellite can be considered as a possible reference vector for attitude determination.

This paper proposes the attitude estimation of an artificial satellite using the own photovoltaic generation to evaluate the azimuthal position of the sun. A mathematical algorithm estimates the satellite attitude using this vector, which is evaluated from the power converter operation, and the Geomagnetic Field, which is measured using a three-axis solid-state magnetometer. Thus, two important subsystems of an artificial satellite can be integrated, providing various benefits always welcome in spatial applications, such as economy, redundancy, autonomy, etc.

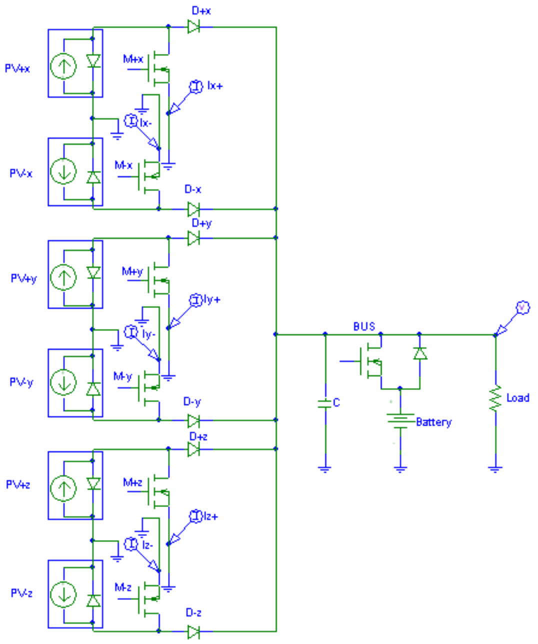


Figure 1: Sequential Shunt Regulator.

2 POWER REGULATION

An artificial satellite generally uses a regulated voltage bus and some form of direct energy transfer for power conditioning of the photovoltaic energy (Shum and Ashley, 1996). Due to a low and relatively constant heat dissipation, the Sequential Shunt Regulator (SSR) is widely used in satellite applications (Garrigós et al., 2006). This power electronic converter is shown in Fig. 1, where is considered a segmented photovoltaic array in six modules. Each photovoltaic module can be admitted as a current source, which contributions are summed by an OR connection of very fast diodes. The photovoltaic current I can either flow to the bus when the MOSFET switch is off, supplying loads and batteries, or be deviated, short-circuiting the photovoltaic module when the MOSFET switch is on. Since the current in the bus is pulsed, a large capacitor filter is required to smooth current pulses and to reduce the voltage ripple.

Since the variations in the capacitor charge reflect the energy exchanges between photovoltaic arrays and electrical loads, the bus voltage must be controlled to assure the power balance, restricting the voltage variations to operational limits of the onboard equipments. The voltage controller modulates the duty cycle of a PWM signal, which simultaneously drives all six MOSFET's switches of the SSR. The battery operates in stand-by, assuming the load when the bus voltage decreases and the series diode becomes directly polarized, which occurs during eclipses, faults and eventual current peeks. A MOSFET switch can connect the battery to the power bus for its recharge when the satellite is illuminated again.

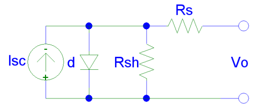


Figure 2: Equivalent electrical circuit of a photovoltaic cell.

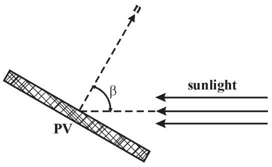


Figure 3: Definition of the sunlight incidence angle β.

3 SUN POSITION ACQUISITION

Usually, the azimuthal position of the sun is evaluated using a sensor based in photovoltaic cells, cameras or CCD sensors (Winetraub et al., 2005; Chen et al., 2006). However, an interesting possibility for artificial satellites is the use of its own photovoltaic system for evaluation of the azimuthal position of the sun (Sityar, 1992; Santoni and Bolotti, 2000; Viscito and Cerise, 2007), since the light-generated power is highly dependent of the intensity and direction of the incident sunlight. A photovoltaic module consists in series-parallel arrays of several photovoltaic cells, which equivalent electrical circuit is shown in Fig. 2. The series resistance R_s depends on the p-n junction depth, impurities and the contact resistance, while the shunt resistance R_{sh} is inversely related with leakage current to the ground. The light-generated electrical current I_{sc} depends on the efficiency of photovoltaic conversion and the incident solar radiation over the photovoltaic cell. This current can be approximated by the mathematical cosine model (Patel, 1999)

$$I_{sc} = I_{sc}|_{\beta=0^o} \cos\beta. \tag{1}$$

where the incidence angle β is defined in the Fig. 3. Since the electrical output in the real photovoltaic cells deviates significantly from the proposed cosine model for $\beta > 50^o$, this approach can lead to inaccuracies in the determination of the sun positioning (Winetraub et al., 2005; Sityar, 1992; Patel, 1999). For a better accuracy, it is recommended the use of

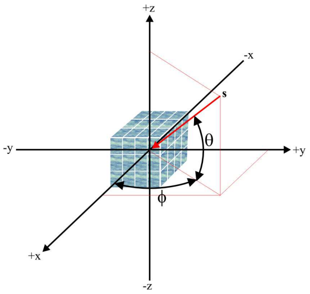


Figure 4: Incidence of the sunlight in a box shaped satellite.

other power-angle curves of the photovoltaic cell, such as the Kelly cosine (Patel, 1999). Neglecting the diode and shunt-leakage currents, which are very small in real cells, the current I_{sc} can be evaluated short-circuiting the output of a photovoltaic array (Sityar, 1992).

An illuminated box shape satellite with all six sides covered by photovoltaic modules is shown in Fig. 4. Considering a coordinate system constituted by the normal axes to the surfaces of this box satellite, a geometric inspection reveals that the components of a vector that points towards to sun are the own summations of light-generated currents of opposite side photovoltaic modules. Thus, the unitary vector $\hat{\mathbf{s}}$ that points towards to sun is given by

$$\hat{\mathbf{s}} = \frac{I_{sc_{+x}} - I_{sc_{-x}}}{I_{sc_x}\big|_{\phi=0^o,\theta=0^o}}\mathbf{i} + \frac{I_{sc_{+y}} - I_{sc_{-y}}}{I_{sc_y}\big|_{\phi=90^o,\theta=0^o}}\mathbf{j} + \frac{I_{sc_{+z}} - I_{sc_{-z}}}{I_{sc_z}\big|_{\theta=90^o}}\mathbf{k}, \tag{2}$$

where $I_{sc_{+x}}$, $I_{sc_{-x}}$, $I_{sc_{+y}}$, $I_{sc_{-y}}$, $I_{sc_{+z}}$, and $I_{sc_{-z}}$ are the light-generated currents by the photovoltaic panels respectively located in the axes $+x$, $-x$, $+y$, $-y$, $+z$, and $-z$.

The scheme for acquisition of azimuthal position of the sun from the SSR operation is shown in Fig. 5. When the outputs of opposite photovoltaic module are short-circuited by their respective shunt MOSFET's of the power converter, a Hall current transducer measures the differential light-generated current in the respective coordinate axis. This signal is sampled in S/H circuit while the shunt MOSFET's are on. When the shunt MOSFET's are off, the S/H circuit holds the differential current measurement, assuring that this component of vector $\hat{\mathbf{s}}$ is always availed for acquisition.

4 SIMULATION RESULTS

The acquisition of the vector sun position is verified from a computational simulation of the SSR using the SimPowerSystems of the MATLAB/SIMULINK. It is considered that a box shape satellite is rotating in the space at $\dot{\phi} = 1.26$rad/s and $\dot{\theta} = 6.28$rad/s (see Fig. 4). The surface of this satellite is covered by six photovoltaic arrays that generates $6 \times 10W_p$ at an irradiance of $1000W/m^2$. The sunlight irradiance in the space is considered $1367W/m^2$. The duty cycle D of the SSR is modulated by a PWM of 5kHz, and a capacitor of $1500\mu F$ is connected to power bus aiming to reduce the voltage ripples. Since the bus capacitance behaves as a big integrator, a simple proportional controller, which gain is adjusted to 300, is enough for a null error in the regulation of the bus voltage. The voltage reference is adjusted to 15V to supply a resistive load of 90Ω and a 12V battery. A hysteresis controller monitors the battery charge and commands the MOSFET switch, connecting the battery to the power bus when its voltage level is the minimum. Aiming to preserve the useful life of the battery, it is disconnected from bus voltage when the maximum charge is reached.

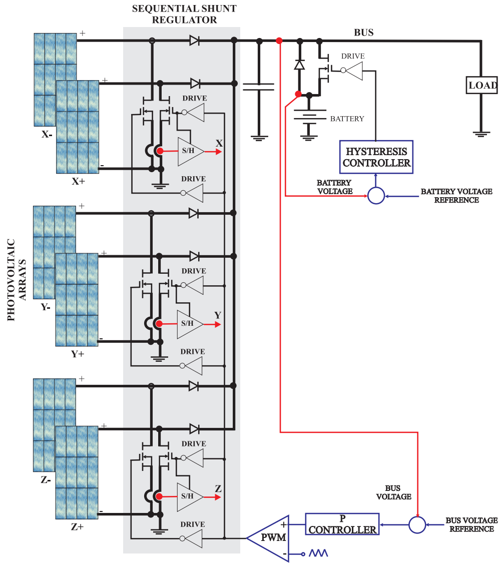


Figure 5: Scheme for acquisition of azimuthal position of the sun from the SSR and bus voltage regulation.

The real and acquired azimuthal sun position is shown in the Fig. 6. The error in the acquired vector is small, basically caused by the sample operation. The attitude motion (angular frequencies of the satellite) can be evaluated from the Fast Fourier Transform (FFT) of the acquired data. Although the photovoltaic power supply is highly variable due to satellite rotation, the proposed P controller is enough to assure a null voltage error, providing an excellent regulation of the bus voltage for the considered load, as shown in the Fig. 7.

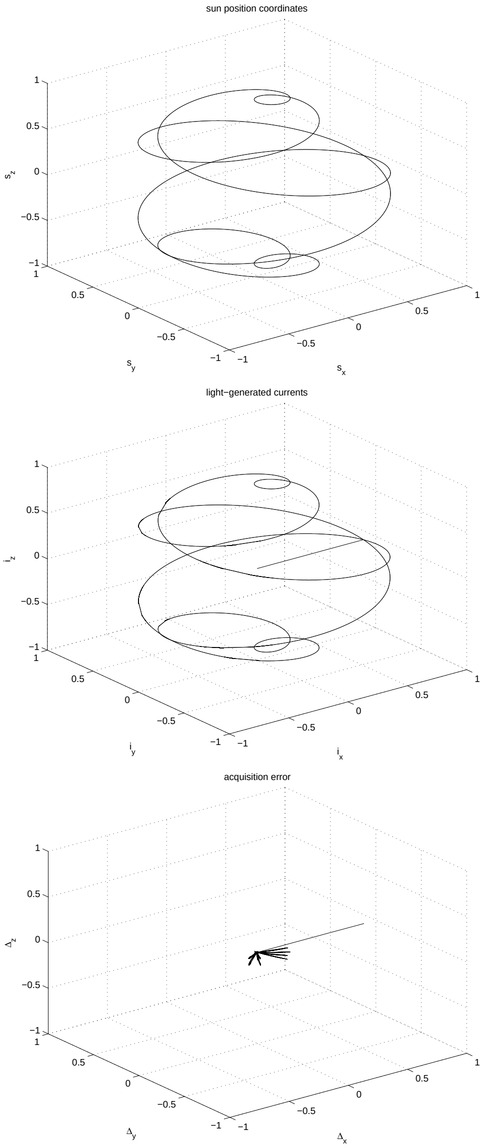


Figure 6: Azimuthal position of the sun.

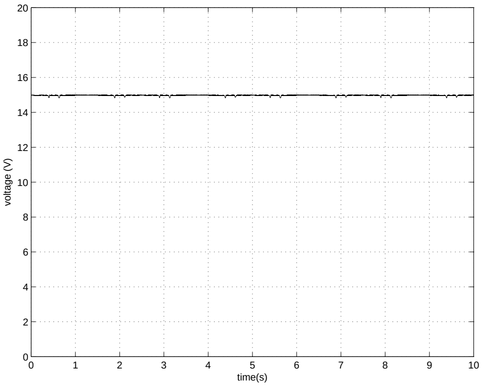


Figure 7: Regulated bus voltage in the Sequential Shunt Regulator.

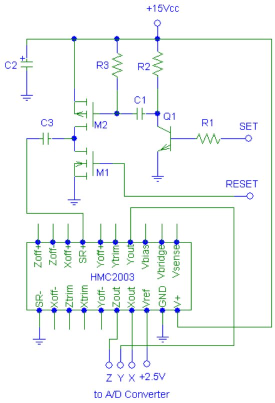


Figure 8: Magnetometer circuit.

5 MAGNETOMETER CIRCUIT

The measurement circuit for Geomagnetic Field is shown in Fig. 8. The heart of this circuit is a small solid-state three-axis magneto-resistive transducer Honeywell HMC2003,which output voltage signals are proportional to the magnitudes of the three ordinal components of the applied magnetic field in a range of the 0 to +5V, where +2.5V represents the reference value for a null intensity of the magnetic field. In order to maximize the transducer resolution, a strong SET/RESET pulse must be occasionally applied to transducer to eliminate the effect of the past magnetic history and to avoid the degradation of the output signal. The output voltage signals X, Y and Z can be connected directly to an analog-to-digital (A/D) converter.

6 ATTITUDE ESTIMATION

A common way to specify the attitude of a body is the use of the Euler's angles $\psi\theta\phi$, which represent three consecutive rotations in a convenient sequence around the axis of an inertial system. The combination of these rotations results in the attitude matrix $\mathbf{A}$, which represents the orientation of an object in relation to inertial coordinate system. Considering a stipulated reference vector $\mathbf{v}_i$, its rotation to obtain an observed vector $\mathbf{w}_i$ by one of the n sensors of the satellite is described as

$$\mathbf{w}_i = \mathbf{A}\mathbf{v}_i, \tag{3}$$

where an estimative of the attitude matrix $\mathbf{A}$ can be obtained from the minimization of the cost function:

$$L(\mathbf{A}) = \frac{1}{2}\sum_{i=1}^{n} a_i(\mathbf{w}_i - \mathbf{A}\mathbf{v}_i)^2 \tag{4}$$

with the non negative weights a_i submitted to restriction $\sum_{i=1}^{n} a_i = 1$ (Shuster and Oh, 1981). This optimization problem can be conveniently simplified expressing it in terms of the quaternion $\bar{\mathbf{q}}$, an alternative attitude representation defined as:

$$\bar{\mathbf{q}} = \begin{bmatrix} \mathbf{Q} \\ \mathbf{q} \end{bmatrix} = \begin{bmatrix} \sin(\theta/2)\mathbf{n} \\ \cos(\theta/2) \end{bmatrix} \qquad (5)$$

and related with an attitude matrix $\mathbf{A}$ by:

$$\mathbf{A}(\bar{\mathbf{q}}) = (\mathbf{q}^2 - \mathbf{Q}\mathbf{Q}^T)\mathbf{I} + 2\mathbf{Q}\mathbf{Q}^T + 2\mathbf{q}\tilde{\mathbf{Q}} \qquad (6)$$

In terms of quaternions, the solution of this optimization problem is given by an algorithm known as Q-method, which consists of a simple generalized problem of eigenvalues and eigenvectors described by (Keat, 1977; Shuster and Oh, 1981)

$$\mathbf{K}\bar{\mathbf{q}}_{opt} = \lambda_{max}\bar{\mathbf{q}}_{opt}, \qquad (7)$$

where the optimal quaternion $\bar{\mathbf{q}}_{opt}$ that minimizes the cost function $L(\mathbf{A})$ is the eigenvector associated to maximum eigenvalue λ_{max} of the matrix $\mathbf{K}$, given by

$$\mathbf{K} = \begin{bmatrix} \mathbf{S} - \sigma\mathbf{I} & \mathbf{Z} \\ \mathbf{Z} & \sigma \end{bmatrix}, \qquad (8)$$

where $\sigma = \sum_{i=1}^{n} a_i \mathbf{w}_i \mathbf{v}_i$, $\mathbf{S} = \sum_{i=1}^{n} a_i (\mathbf{w}_i \mathbf{v}_i^T + \mathbf{v}_i \mathbf{w}_i)^T$, and $\mathbf{Z} = \sum_{i=1}^{n} a_i \mathbf{w}_i \times \mathbf{v}_i$.

7 INTEGRATION

Considering spatial applications, both the power regulation and the attitude estimation must be integrated in an unique, compact and low consumption onboard platform, which should read analog signals, compute the present satellite attitude, and perform the power control and energy management. This platform must still execute other complementary functions of the satellite such as telemetry, command, control, communication and error analysis. In this context, an interesting high performance and low cost option is a DSP-based platform, which combines a high processing speed processor, great amount of memory and several peripheral devices for real time digital processing signal, such as A/D converters, I/O ports, PWM modules, parallel and serial communication interfaces, and special modules to read encoders, counters, timers, etc. The programming uses high level language, presenting several tools to develop complex algorithms such as FFT (Fast Fourier Transform), filters and other indispensable functions for the satellite operation, such as attitude estimation and control, power regulation and management, auto-diagnose, communications, fail analysis, and data storage.

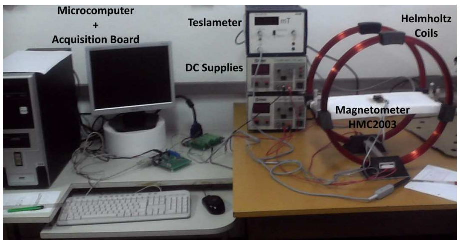


Figure 9: Experimental implementation.

8 EXPERIMENTAL RESULTS

The Q-method algorithm is experimentally implemented using C language in a starter kit module based in the Texas Instruments DSP TMS320F2808. A photography of this practical implementation is presented in the Fig. 9. While reference vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ are considered fixed and its values are directly inserted in the code, the observation vectors $\mathbf{w}_1$ and $\mathbf{w}_2$ are acquired using sample rate superior to 10 Hz. The vector $\mathbf{w}_1$ is the magnetic field produced by Helmholtz coils, which is measured using the magnetometer circuit presented in Fig 8, while the vector $\mathbf{w}_2$ is emulated by potentiometers. The experimental results of implementation considering two static known situations are shown in the table I, where is observed that this DSP platform can provide satisfactory attitude estimations for this satellite application.

9 FINAL DISCUSSION

This paper presents a proposal to acquire the azimuthal position of the sun using the own power photovoltaic supply of an artificial satellite. Considering a box shape satellite, where all sides are covered by photovoltaic modules, the components of the azimuthal position of the sun correspond to summations of the light-generated currents by opposite photovoltaic modules, which can be evaluated from the operation of the SSR power converter. In the eventual absence of a photovoltaic module, photovoltaic cells or photodiodes can substitute it in the satellite configuration. This information about the sun position and the measurement of other known vector, such as the Geomagnetic Field, can be used to estimate the attitude, allowing the integration of two of the more important subsystems for the operation of an artificial satellite. The integration of these subsystems can be implemented in a DSP platform, which would realize data acquisition, power regulation, battery management, attitude determination and others important satellites functions. The simulation results

Table 1: Experimental results.

	$\mathbf{v}_1$	$\mathbf{w}_1$	$\mathbf{v}_2$	$\mathbf{w}_2$
$\mathbf{w}_1$ and $\mathbf{w}_2$ are aligned with $\mathbf{v}_1$ and $\mathbf{v}_2$.				
x	0.0000	0.0000	0.0000	0.0000
y	1.0000	1.0000	0.0000	0.0000
z	0.0000	0.0000	1.0000	1.0000
Theoretical attitude matrix	$\begin{bmatrix} 1.0000 & 0.0000 & 0.0000 \\ 0.0000 & 1.0000 & 0.0000 \\ 0.0000 & 0.0000 & 1.0000 \end{bmatrix}$			
Experimental attitude matrix (DSP)	$\begin{bmatrix} 0.9994 & 0.0352 & -0.0006 \\ -0.0352 & 0.9992 & -0.0179 \\ 0.0000 & 0.0179 & 0.9999 \end{bmatrix}$			
$\mathbf{w}_1$ is inclined 45^o in relation to $\mathbf{v}_1$, while $\mathbf{w}_2$ and $\mathbf{v}_2$ are aligned.				
	$\mathbf{v}_1$	$\mathbf{w}_1$	$\mathbf{v}_2$	$\mathbf{w}_2$
x	1.0000	1.0000	0.0000	0.0000
y	0.0000	1.0000	0.0000	0.0000
z	0.0000	0.0000	1.0000	1.0000
Theoretical attitude matrix	$\begin{bmatrix} 0.7071 & -0.7071 & 0.0000 \\ 0.7071 & 0.7071 & 0.0000 \\ 0.0000 & 0.0000 & 1.0000 \end{bmatrix}$			
Experimental attitude matrix (DSP)	$\begin{bmatrix} 0.7017 & -0.7124 & 0.0028 \\ 0.7124 & 0.7013 & 0.0029 \\ -0.0040 & 0.0000 & 1.0000 \end{bmatrix}$			
Reference and observation vectors				
	$\mathbf{v}_1$	$\mathbf{w}_1$	$\mathbf{v}_2$	$\mathbf{w}_2$
x	1.0000	0.70523	0.0000	-0.00793
y	0.0000	0.70884	1.0000	0.00881
z	0.0000	-0.01353	0.0000	0.99999
Theoretical attitude matrix (MATLAB)	$\begin{bmatrix} 0.70522 & -0.00338 & 0.70897 \\ 0.70894 & 0.01337 & -0.70513 \\ -0.00709 & 0.99990 & 0.01183 \end{bmatrix}$			
Experimental attitude matrix (DSP)	$\begin{bmatrix} 0.70523 & -0.00337 & 0.70896 \\ 0.70894 & 0.01339 & -0.70513 \\ -0.00711 & 0.99904 & 0.01183 \end{bmatrix}$			

shows that the azimuthal position of the sun can be evaluated from the SSR operation with sufficient accuracy for attitude determination. The results of the experimental implementation in a DSP platform of the q-Method, an algorithm that involves a theoretically great computational effort, are satisfactory for this satellite application. An experimental evaluation of this proposal will realized using the little prototype of a box-shape satellite (Fig. 10), where the subsystems related to power regulation and attitude estimation will be integrated by a DSP platform. Other satellite configuration will be also considered in future studies related to this subject.

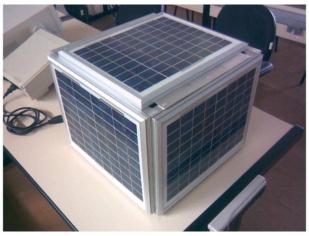


Figure 10: Box-shape structure.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the financial support of the Federal University of Ouro Preto, National Counsel of Technological and Scientific Development (CNPq), State of Minas Gerais Research Foundation (FAPEMIG) and Gorceix Foundation.

REFERENCES

Chen, F., Feng, J., and Hong, Z. (2006). Digital sun sensor based on the optical vernier measuring principle. *Measurements Science and Technology*, 17:24942498.

Garrigós, A., Carrasco, J., Blanes, J., and Sanchis-Kilders, E. (2006). A new sequential switching shunt regulator - digital shunt regulator (S3R-DSR) for solar array regulators. In *2006 IEEE International Symposium on Industrial Electronics - ISIE 06*.

Keat, J. (1977). Analysis of least-squares attitude determination routine DOAOP. Technical report, National Aeronautics and Space Administration (NASA).

Patel, M. (1999). *Wind and solar power systems*. CRC Press.

Santoni, F. and Bolotti, F. (2000). Attitude determination of small spinning spacecraft using threeaxis magnetometer and solar panels data. In *2000 IEEE Aerospace Conference*.

Shum, K. and Ashley, C. (1996). A new full shunt switching unit for solar array using coupled-inductor boost converter. In *31st Intersociety Energy Conversion Engineering Conference - IECEC 96*.

Shuster, M. and Oh, S. (1981). Three-axis attitude determination from vector observations. *Journal of Guidance and Control*, 4(1):70–77.

Sityar, I. (1992). Sun sensor implementation using solar power arrays. Master's thesis, Naval Postgraduate School, Monterey, USA.

Viscito, L. and Cerise, M. (2007). Rate and attitude determination using solar array currents. In *2007 Colorado Space Grant Consortium's: Undergraduate Space Research Symposium*.

Winetraub, Y., Bitan, S., Nativ, Y., and Heller, A. (2005). Attitude determination advanced sun sensors for picosatellites. In *17th European Union Contest for Young Scientists*.

EXHIBITING PLANAR STRUCTURES FROM EGOMOTION

Samia Bouchafa, Antoine Patri and Bertrand Zavidovique
Institut d'Electronique Fondamentale, University Paris XI, 91405 Orsay Cedex, France
{samia.bouchafa, antoine.patri, bertrand.zavidovique}@u-psud.fr

Keywords: Image motion analysis, Pattern recognition, Image scene analysis, Ego motion.

Abstract: This paper deals with plane extraction from a single moving camera through a new optical-flow cumulative process. We show how this c-velocity defined by analogy to the v-disparity in stereovision, could serve exhibiting any plane whatever their orientation. We focus on 3D-planar structures like obstacles, road or buildings. A translational camera motion being assumed, the c-velocity space is then a velocity cumulative frame in which planar surfaces are transformed into lines, straight or parabolic. We show in the paper how this representation makes plane extraction robust and efficient despite the poor quality of classical optical flow.

1 INTRODUCTION

Our work deals with obstacle detection from moving cameras. In this application, most of real-time implemented approaches are based on stereovision. Yet stereo analysis shows two main drawbacks. First, it tends to group objects which are close to one-an-other. Second, height thresholds limit the detection implying to miss small obstacles close to the ground for example. Motion information is only exploited afterwards for detected objects. Motion analysis, on the other hand, allows the detection of any moving object. Therefore, we propose to exploit the ego-motion of the camera to distinguish between various moving objects. To that aim, we have established a correspondence with a very efficient stereovision technique based on the **v**-disparity concept (Labayrade, Aubert and Tarel, 2002). Our conjecture is that such result is general. Thereby we show how to extend the technique to detect planes along an image sequence shot from a moving vehicle. The apparent velocity from the scale change occurring to image data takes place of the disparity leading to the so-called **c**-*velocity* frame. In this paper we propose a complete plane detection process. Peculiar emphasis is placed on the parabolas detection in the **c**-*velocity* space.

The paper is organized as follows: in the next section we take a look at ego-motion based object detection. Then we recall some pertaining relations between 2D and 3D motion. The fourth part is devoted to the computation of constant velocity curves in the image plane – analogue in our ***c****-velocity* frame of the lines of the *v-disparity,* and we explain the cumulative process. The fifth section details how parabolas – 3D planes – are extracted in the ***c****-velocity* space using a Hough transform enriched by a K-mean technique. After the section devoted to results we conclude with discussions and future work.

2 PREVIOUS WORK

Recent years have seen a profusion of work on 3D motion, ego-motion or structure from motion estimation using a moving camera. It was followed by numerous classifications of existing methods based on various criteria. A classification commonly accepted groups existing techniques into three main categories: discrete, continuous and direct approaches.

- Discrete approaches (Hartley, 1995) are based on matching and tracking primitives that are extracted from image sequences (point, contour lines, corners, etc.). They are usually very effective. However, they suffer from a lack of truly reliable and stable features, e.g. time and viewpoint invariant. Moreover, in applications where the camera is mounted on a moving vehicle, homogeneous zones or linear marking on the ground hamper the extraction of reliable primitives.

- Continuous approaches exploit optical flow (MacLean, Jepson and Frecker, 1994). The relationship between the computed optical flow and real theoretical 3D motion allows - through optimization techniques - to estimate the motion parameters and depth at each point. Results are dependent on the quality of the computed optical flow.

- In direct approaches (Stein, Mano and Shashua, 2000), motion is determined directly from the brightness invariance constraint without having to calculate explicitly an optical flow. Motion parameters are then deduced by conventional optimization approaches.

- A large group of approaches (Irani, Rousso and Peleg, 1997) - which can be indifferently discrete, continuous or direct - exploits the parallax generated by motion (motion parallax, affine motion parallax, plane+parallax). These methods are based on the fact that depth discontinuities make it doable to separate camera rotation from translation. For instance, in "Plane+parallax" approaches, knowing the 2D motion of an image region where variations in depth are not significant permits to eliminate the camera rotation. Using the obtained residual motion parallax, translation can be exhibited easily.

3 PRELIMINARIES

Consider a coordinate system O XYZ at the optical centre of a pinhole camera, such that the axis OZ coincides with the optical axis. We assume a translational rigid straight move of the camera in the Z direction. That does not restrict the generality of computations. Moreover, the origin of the image coordinates system is placed on the top left of the image. If (x_0, y_0) are the coordinates of the principal point, then the ego-motion (u, v) becomes:

$$u = \frac{T_z}{Z}\left(y_0 - y\right) \text{ and } v = \frac{T_z}{Z}\left(x - x_0\right)$$

The previous equations describe a 2D motion field that should not be confused with optical flow which describes the motion of observed brightness patterns. We will assume here that optical flow is a rough approximation of this 2D motion field. In order to tackle the imprecision of optical flow velocity vectors, we propose to define a Hough-like projection space which – thanks to its cumulative nature –allows performing robust plane detection.

4 NEW CONCEPT: C-VELOCITY

In stereovision, along a line of a stereo pair of rectified images, the disparity is constant and varies linearly over a horizontal plane in function of the depth. Then, in considering the mode of the 2-D histogram of disparity value vs. line index, i.e. the so called *v-disparity* frame, the features of the straight line of modes indicate the road plane for instance (Labayrade, Aubert and Tarel, 2002). The computation was then generalized to the other image coordinate and vertical planes using the *u-disparity* by several teams including ours on our autonomous car.

In the same way we have transposed this concept to motion. Our computations build on the fact that any move of a camera results into an apparent shift of pixels between images: that is disparity for a stereo pair and velocity for an image sequence. The *v-disparity* space draws its justification, after image rectification that preserves horizontal – iso-disparity – lines, from inverse-proportional relations between first image horizontal-line positions vs. depth, second depth vs. disparity. We show here under how to exhibit the same type of relation in the ego-motion case between $\|\mathbf{w}\|$ (≅ disparity) and the iso-velocity function index *c* (≅ line index *v*).

$$\|\mathbf{w}\| = \sqrt{u^2 + v^2} = \left|\frac{T_z}{Z}\right| \sqrt{(y_0 - y)^2 + (x - x_0)^2}$$

$$\|\mathbf{w}\| = K \times f(x, y) \Rightarrow \frac{\|\mathbf{w}\|}{K} = f(x, y) = c$$

The translation T_Z being that of the camera, identical for all static points, if depth *Z* is constant the iso-velocity curves are circles. *c* varies linearly with the velocity vector. Beyond that “punctual” general case, *Z* can be eliminated in considering linear relations with (*X*,*Y*) i.e. plane surfaces well fitting the driving application for instance.

4.1 The Case of a Moving Plane

Suppose now the camera is observing a planar surface of equation (Trucco and Poggio, 1989): $\mathbf{n}^{\mathrm{T}}\mathbf{P} = d$, with $\mathbf{n} = (n_x, n_y, n_z)$ the unit normal to the plane and *d* the distance "plane to origin". Let us assume that the camera has a translational motion $\mathbf{T} = \left(0, 0, T_Z\right)$. We study four pertaining cases of moving planes and establish the corresponding motion field. a) Horizontal: road. b) Lateral: buildings. c)Frontal$_1$: fleeing or approaching obstacle, with $\mathbf{T} = (0, 0, T_Z^o)$. d) Frontal$_2$: crossing obstacle, with $\mathbf{T} = (T_X^o, 0, 0)$.

	Normal vector	*Associated 3D motion*	*Dist. to the origin*
a)	$\mathbf{n} = (0,1,0)$	$\mathbf{T} = \left(0, 0, T_Z\right)$	dist. d_r
b)	$\mathbf{n} = (1,0,0)$	$\mathbf{T} = \left(0, 0, T_Z\right)$	dist. d_b
c)	$\mathbf{n} = (0,0,1)$	$\mathbf{T} = \left(0, 0, T_Z + T_Z^o\right)$	dist. d_o
d)	$\mathbf{n} = (0,0,1)$	$\mathbf{T} = \left(T_X^o, 0, T_Z\right)$	dist. d_o

The corresponding motion fields, after [2] for instance, become those listed in the table below for each case. Let $\|\mathbf{w_o}\|$, $\|\mathbf{w_r}\|$ and $\|\mathbf{w_b}\|$ be respectively the module of the apparent velocity of an obstacle point, a road point and a building point. We choose to group all extrinsic and intrinsic parameters in a factor *K* and make it the unknown:

a)	$u = \dfrac{T_Z}{f \times d_r}(y - y_0)(x_0 - x)$ $v = \dfrac{T_Z}{f \times d_r}(x_0 - x)^2$
b)	$u = \dfrac{T_Z}{f \times d_b}(y - y_0)^2$ $v = \dfrac{T_Z}{f \times d_b}(y - y_0)(x_0 - x)$
c)	$u = \dfrac{T_Z + T_Z^o}{d_o}(y - y_0)$ $v = \dfrac{T_Z + T_Z^o}{d_o}(x_0 - x)$
d)	$u = \dfrac{T_Z}{d_o}(y - y_0) - \dfrac{T_X^o f}{d_o}$ $v = \dfrac{T_Z}{d_o}(x_0 - x)$

a)	$\|\mathbf{w}_r\| = K\sqrt{(x_0 - x)^4 + (y - y_0)^2 (x_0 - x)^2}$
b)	$\|\mathbf{w}_r\| = K\sqrt{(y - y_0)^4 + (y - y_0)^2 (x_0 - x)^2}$
c)	$\|\mathbf{w}_r\| = K\sqrt{(y - y_0)^2 + (x_0 - x)^2}$
d)	$\|\mathbf{w}_o\| = K$ if T_X^o □ $\mathrm{T_Z}$ $\|\mathbf{w}_o\| = K\sqrt{(x_0 - x)^2 + (y - y_0)^2}$ otherwise

Each type of $\|\mathbf{w}\|$ leads to the corresponding expression of **c** and then to the related iso-velocity curve. For instance in the case of a building plane:

$$c = \frac{\|\mathbf{w}\|}{K} = \sqrt{(y - y_0)^4 + (y - y_0)^2 + (x_0 - x)^2}$$

The final formula above proves that *c* is constant along iso-velocity curves and proportional to $\|\mathbf{w}\|$**,** same as the disparity is proportional to the line value *v*. Thus, as explained in introduction, the **c**-*velocity* space will be a cumulative space that is constructed in assigning to each pixel (x, y) the corresponding *c* value through the chosen model, and in incrementing the resulting $\left(c, \|\mathbf{w}\|\right)$ cell were **w** is the velocity found in (x, y). The latter **w** is computed thanks to a classical optical flow method. A study of the function $c(x, y)$ that corresponds to each plane model – in particular for the road and the building model – led us to the following conclusions: first, each previous curve intersects the *x* axis (road model) or *y* axis (building model) in the image plane in: $y_0 \pm \sqrt{c}$ or $x_0 \pm \sqrt{c}$ respectively. Second, for a standard image size, the range of variation of *c* is very large. For instance, for an image size of 320×240: $c_{\max} = 32000$ (road model) and 24000 (building model). As a consequence and for implementation reasons, we propose to deal for these two models with the relations between ||**w**|| and $\sqrt{c}$ (see Figure 1).

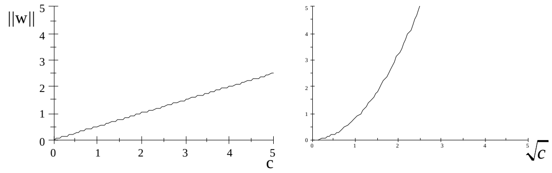
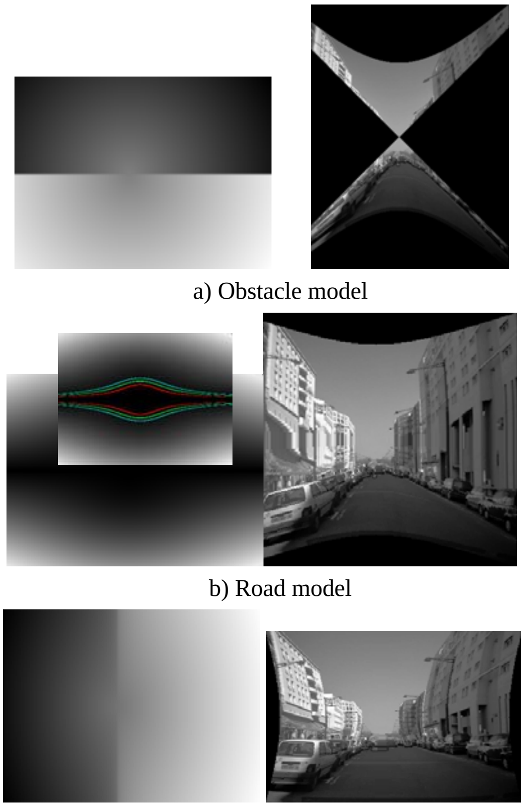


Figure 1: The c-velocity space that depends on the chosen relation between c and w: Linear for the obstacle model and parabolic for the road and the building model.

4.2 Cumulative Curves

For each point $\mathbf{p} = (x, y)$ in the image, there is an associated **c** value depending on the chosen plane model (see left column of Figure 2). We can calculate it once off-line because it only depends on (x, y). Also, it is possible for implementation facilities and by analogy to image rectification (that makes all epipolar lines parallel) to compute the transformation that makes all the c-curves parallel to the image line, that is the intensity function $I(c, y)$ for road and obstacle model and $I(x, c)$ for building model (see right column of Figure 2).



a) Obstacle model

b) Road model

c) Building model

Figure 2: Left: for each model, the corresponding c-values for each point of the image. Right: images constructed using the geometric transformation that makes all c-curves parallel**.**

5 1D HOUGH TRANSFORM AND K-MEAN CLUSTERING FOR PARABOLAS EXTRACTION

Planes are represented in the **c**-*velocity* space by parabolas that could be extracted using a Hough transform. The distance *p* between each parabola and its focus or its linea directrix is then cumulated in a one dimensional Hough transform (see Figure 3). The classes of the histogram split by K-mean clustering. Of course, any other clustering approach could be applied.

$$\|w\| = K\left(\sqrt{c}\right)^2 \Rightarrow p = -\frac{1}{4K} = -\frac{\left(\sqrt{c}\right)^2}{4\|w\|}$$

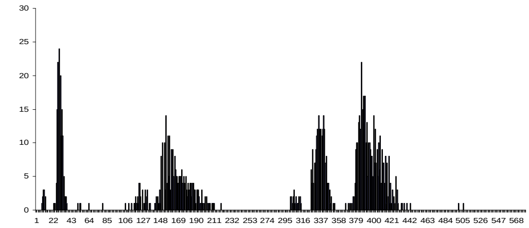
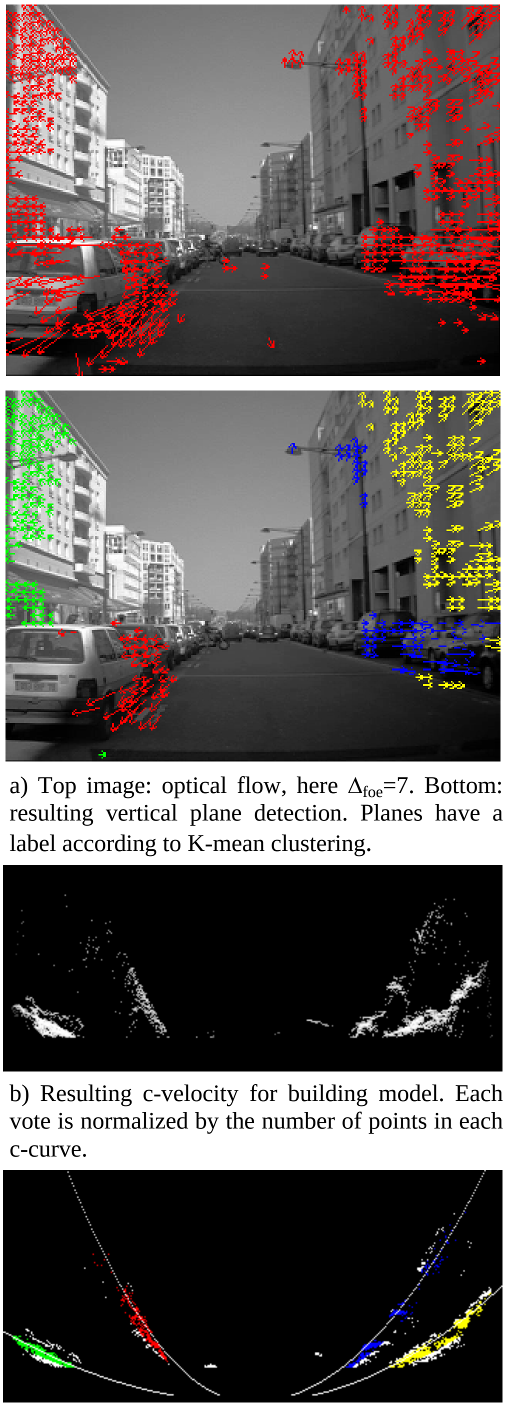


Figure 3: Example of a 1D Hough transform on the c-velocity space for detecting parabolas. For each (c,w) cell, a *p* value is cumulated.

6 EXPERIMENTAL RESULTS

In Figure 4, we have considered an image sequence in which one can see 6 moving planes: 2 planes corresponding to buildings, 2 planes corresponding to cars parked on the sides, a frontal moving obstacle (a motorcycle crossing the road) and the road plane. We have used the Lukas & Kanade method for optical flow estimation (Lucas and Kanade, 1981). In this sequence, velocity vectors are in majority on vertical planes. In the building **c**-velocity space, we get as expected 4 parabolas (see Figure 4.b). We have studied in the effects of 3 kinds of perturbations that have a consequence on the thickness of the parabolas. First, inter-model perturbation, second the imprecision on optical flow and third the possible pitch, yaw or roll of the camera. We use 2 kinds of confidence factors. First one is related to the translational motion hypothesis; it is the difference Δ_{foe} between the coordinate of image centre and the position of the Focus of expansion (for its estimation see section 6.1, results on Figure 6). Second one is related to possible contamination by planes of other models; it is the variance σ of each K-mean class. Points far from the mean belong probably to another plane model (Figure 4.c).



a) Top image: optical flow, here Δ_{foe}=7. Bottom: resulting vertical plane detection. Planes have a label according to K-mean clustering.

b) Resulting c-velocity for building model. Each vote is normalized by the number of points in each c-curve.

c) Results of parabolas extraction using a 1D Hough transform followed by a K-mean clustering (4 classes). In white the points that are discarded: they probably belong to another plane model. σ_{mean} = 10.

Figure 4: Example of results obtained from a database of the French project “Love” (Logiciel d’Observation des VulnérablEs).

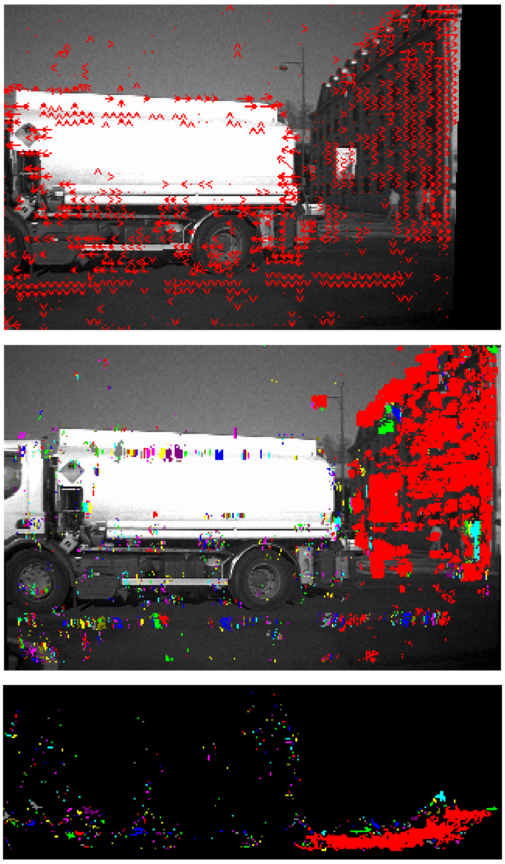


Figure 5: Results of a building detection (top left image in red). The crossing obstacle here is – as expected – not detected in the building **c**-*velocity* space.

6.1 FOE Estimation

Several methods exist (Sazbon, Rotstein and Rivlin, 2004). For sake of further real on board implementation, we favor here a method coherent with the present computations. All pixels are asked to vote for a global intersection point of apparent velocity vectors within a regular Hough space.

Indeed, in the case of a translational motion, each velocity vector with angle θ is directed toward the FOE. Let us assume that (x_0, y_0) are the FOE coordinates in the image. Then we have:

$$\theta = \tan^{-1}\left(\frac{v}{u}\right) = \tan^{-1}\left(\frac{x - x_0}{y_0 - y}\right)$$

The above relation means that we can extract the FOE by estimating the intersection of all velocity vector lines. In practice, we have carried out a voting space where each velocity vector votes for all the points belonging to its support line. The FOE corresponds then to the point with maximum votes.

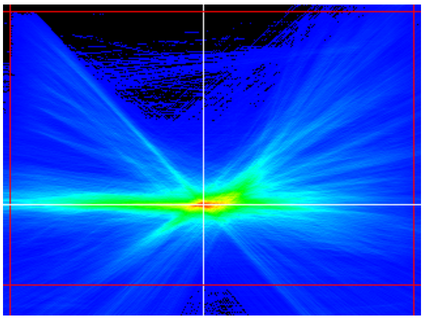
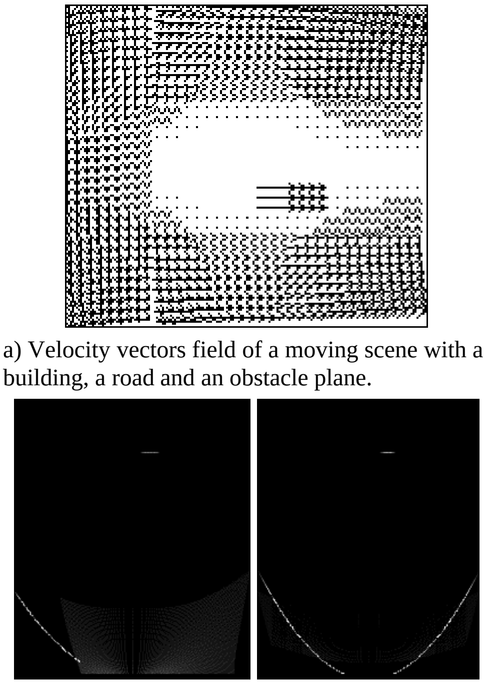


Figure 6: Voting space for FOE determination.

6.2 Results on Synthetic Images

In the following toy example, we generate a synthetic velocity vectors field of a moving 3D scene with 3 planes: a vertical one (on the left of the image), an horizontal one (on the bottom of the image) and a frontal plane with its own motion parameters (a crossing obstacle), see Figure 7.a.



a) Velocity vectors field of a moving scene with a building, a road and an obstacle plane.

b) Associated c-velocity spaces (left: building, right: road). The parabolas indicate the expected moving planes. The constant segment corresponds to the obstacle; it appears in all the c-velocity spaces because of its constant velocity.

Figure 7: Results on synthetic images.

The results confirm that this simulated ego-motion (Figure 7,b) transforms a road plane and a building plane into a parabola in the corresponding c-velocity space. Likewise the obstacle in the middle of the road is a segment with its own constant **w.**

7 CONCLUSIONS

First results are very encouraging and confirm that the cumulative process is efficient in retrieving major entities of a moving scene environment. Our future work deals with implementing an iterative approach that deals with all the c-velocity spaces. Each detected plane from a given space could be discarded from the other spaces in order to reduce inter-model perturbation. On the other hand, we propose to progressively generalize the approach to more complex structures than planes and to more complex motion models, including rotations for instance.

REFERENCES

Hartley, R.I, 1995. In defense of the 8-point algorithm. In Proc. *IEEE Int. Conf. on Computer Vision*, Cambridge, MA, pp. 1064-1070.

Irani, M , Rousso, B, Peleg, S, 1997. Recovery of Ego-Motion using region alignment. In *IEEE Trans on PAMI*, vol. 19, n°3.

Labayrade, R, Aubert, D, Tarel, JP, 2002. Real Time Obstacle Detection on Non Flat Road Geometry through 'V-Disparity' Representation. In *IEEE Intelligent Vehicles Symposium 2002*, Versailles, June 2002.

Lucas, B.D, Kanade, T, 1981. An Iterative Image Registration Technique with an Application to Stereo Vision (DARPA). In *Proceedings of the 1981 DARPA Image Understanding Workshop*, pp. 121-130.

MacLean, W.J, Jepson, A.D , Frecker, R.C, 1994. Recovery of egomotion and segmentation of independant object motion using the EM algorithme. In *Proc. of the 5th British Machine Vision Conference*, pp. 13-16, York, U. K.

Sazbon, D, Rotstein, H, Rivlin, E, 2004. Finding the focus of expansion and estimating range using optical flow images and a matched filter. In *MVA*, vol. 15, n°4, pp. 229-236.

Stein, G. P , Mano, O, Shashua, A, 2000. A robust method for computing vehicle ego-motion. In *IEEE Intelligent Vehicles Symposium*, Dearborn, MI.

Trucco, E, Poggio, T, 1989. Motion field and optical flow: qualitative properties. In *IEEE Transactions on Pattern Analysis Machine Intelligence*, vol. 11, pp. 490-498.

USING PASSAGES TO SUPPORT OFF-ROAD ROBOT NAVIGATION

Christopher Armbrust, Helge Schäfer and Karsten Berns
Robotics Research Lab, Department of Computer Sciences
University of Kaiserslautern
P.O. Box 3049, 67653 Kaiserslautern, Germany
{*armbrust, b_schaef, berns*} *@cs.uni-kl.de*

Keywords: Virtual sensors, Virtual sensor probes, Passages, Off-road navigation, Autonomy.

Abstract: In this paper an approach for the detection of passages and their use in autonomous off-road robot navigation is presented. The authors argue that many two-layered architectures of robot navigation systems suffer from the gap between the typically coarse-grained high-level path-planning and the basically reactive low-level collision avoidance. In this context, passages shall be defined as paths leading through obstacles. The proposed approach is based on the idea that passages in the proximity to the robot should be evaluated with respect to their relevance for reaching the target area in order to avoid local detours by following suitable passages.

The detection and assessment of passages is based on virtual sensors, a standardized data representation offering a unified, straightforward, and flexible retrieval mechanism for accessing the data provided by different sensor systems. For the evaluation of passages the authors introduce the concept of virtual sensor probes which can move independently from the robot. That way the point of view on the environment information can be tailored to support the detection and evaluation strategy.

The proposed approach was deployed on the mobile off-road platform RAVON which serves as a testbed for the experiments carried out in the context of this work.

1 INTRODUCTION

Many robot navigation systems are designed following a two-layer concept: A deliberative path-planner, the “navigator”, builds global maps, which it uses for high-level path planning. The output of this planner are single path points, which are passed to a lower level component, the “pilot”. This component transforms the target coordinates into motion commands, taking into account sensor data to realize collision avoidance. The pilot typically operates in a nearly reactive way, storing only little state information and having a very much limited view on the robot’s environment. In such an architecture, the pilot exerts a draw towards the target, which the anti-collision system may counteract, resulting in the robot driving around obstacles towards the target.

While this principle works well in simple environments, it can easily fail in more complex terrain. As the collision-avoidance works locally, missing the “big picture”, it cannot keep the navigator from drawing the robot away from a path and into small openings between obstacles, which are so wide that the collision avoidance does not get active directly, but which could be easily recognized as indentations when using a larger scope (see Figure 1, (1)).

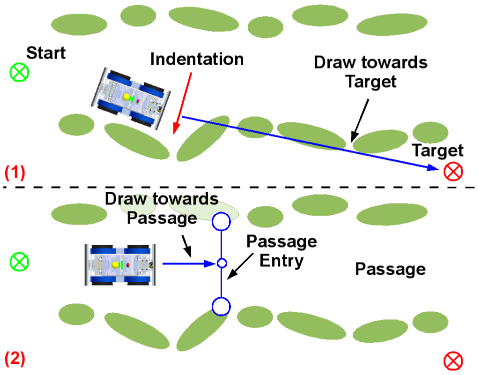


Figure 1: The problem of dealing with indentations.

The idea that is presented here is to search for paths leading through obstacles, so-called *passages*, and to evaluate them with respect to their value for the robot’s navigation. In the described situation, the path the robot is driving on would be detected as such a passage, while the indentation would be ignored (see Figure 1, (2)). Furthermore, estimating a passage's orientation would allow for directing the robot in a way that it can more easily enter the passage.

It shall be emphasized here that the described approach targets at environments in which there is no clear path that could be followed using a path tracking approach like the ones mentioned in section 2. Instead, the paths constituted by passages are consecutive spaces between obstacles that are wide enough to be used by a robot.

2 STATE OF THE ART

The concept of separating a robot navigation system into two layers is widespread and described in many places in the literature. The classic separation results in a low-level system for local navigation that has a limited view on the world and operates in a fast, yet shortsighted manner. Different approaches have been followed to establish the interaction of the two parts ((Wooden et al., 2007), (Ranganathan and Koenig, 2003)). An approach for detecting narrow passages in indoor environments is described in (Schröter, 2005). Its author used the polar data of a laser scanner to identify and assess passages and combined it with 3D rectangular objects reconstructed from the images of a stereo vision system to identify doors. The algorithm used for processing the laser data resembles the one presented here. However, it only processes the data of one polar sensor and not of several (virtual) sensors. Evidently, using a stereo system in such a way to detect passages will not work in unstructured off-road environments. Approaches for keeping the robot from leaving the road or path include detecting curbs using a light-stripe scanner (Thorpe et al., 2003), detecting lanes using edge extraction from images, or road detection using a combination of LADAR data and color information (Hong et al., 2002). The work described in (Lieb et al., 2005) uses the assumption that the vehicle is situated on the road to form templates of the road's appearance and from them and current images calculates an estimate of the road's curvature. (Alon et al., 2006) describes a system that uses two different path-finding algorithms in parallel and uses the output of the one with the highest confidence. However, all of these approaches are tailored to detecting a path in an environment that is completely different from the one of this work.

3 SYSTEM ARCHITECTURE

3.1 A Two-layered Navigation System

The work at hand is embedded into a two-layered robot navigation system consisting of a classical deliberative navigator (Braun and Berns, 2008) on top of a behavior-based system for local navigation, i.e. target-oriented motion and collision avoidance (Schäfer et al., 2008b). The navigator creates and updates topological maps of the environment, which are used as basis for the path planning process. It sends the points of a path one by one to the lower layer, from which it receives basic status information.

3.2 Virtual Sensors

Special behaviors of the local navigation system translate the goal poses into motion commands, which are altered by a behaviour-based anti-collision system. The behaviors use a special type of representation that provides a powerful yet slim interface to arbitrary types of sensors. Different aspects and ranges of the robot's environment are abstracted using polar and Cartesian sector maps, which represent *virtual sensors* (VS). Polar maps are defined by start and stop angles whereas Cartesian maps have an extent in positive and negative y-direction. Each sector holds the most relevant representative of the area it covers. In the architecture presented here, the VS are filled with data from a scrollable, robot-local, and orientation-fixed grid map that stores highly preprocessed sensor data of a pannable laser range finder (Schäfer et al., 2008a). Using this short-range map as input for the virtual sensors allows for monitoring areas which are currently not in the range of the robot's real sensors.

3.3 Virtual Sensor Probes

While the virtual sensors used for collision avoidance are defined with respect to the robot coordinate system (RCS) and are fixedly mounted on the robot, virtual sensors that are used to gather information about specific structures in the environment have to be fix with respect to the working coordinate system (WCS). Hence their pose in terms of the RCS changes whenever the robot moves. This special type of virtual sensor is called *virtual sensor probe* (VSP).

The preprocessing of the sensor data, the coordinate transformations (sensor to robot to working CS), and the filling of the VSPs with the correct data is done in the sensor processing subsystem. Therefore higher components do not have to deal with these technical details, but can simply provide a WCS pose

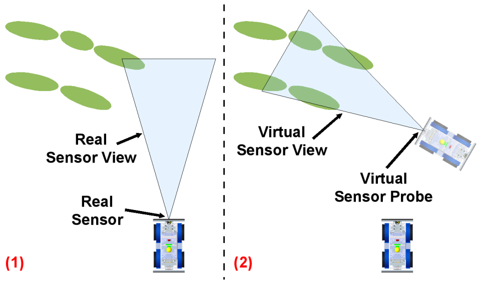


Figure 2: The views of a real sensor and a virtual sensor probe.

for a VSP and be assured that it is filled with the correct information.

Using VSPs, the robot can "look" at a place in the environment from a different point of view (see Figure 2). While this naturally does not yield new information that the robot cannot gather from its actual pose, it structures the existing sensor data in an abstract way, allowing for the use of more straightforward algorithms.

4 THE USE OF PASSAGES

4.1 Passage Detection

A passage is defined by its entry, which is delimited by an obstacle on each side. These two obstacles are called *Passage Entry Points* (PEP), the line connecting them *Passage Entry Line* (PEL). The algorithm for detecting passages traverses all sectors s_i of a polar sector map (see Figure 3, (1)) covering the area in front of the robot and compares the distances d_i of the obstacles o_i (with $0 \leq i < n$ and n being the number of sectors).

If o_{j+1} is farther away from the origin than o_j by at least the threshold t_0, then o_j is considered as potential first PEP. Be o_k with $k \geq j+1$ the first obstacle that is closer to the origin than o_{j+1} by at least the threshold t_1. Then o_k is the second PEP and all obstacles between o_j and o_k are considered to lie within the passage. The new passage is added to a list and the search goes on with the remaining sectors. In the following, pseudocode for the detection algorithm is shown.

For every passage, the *passage entry midpoint* (PEMP), which is defined as the point lying in the middle between o_j and o_k, is calculated. It is a passage's characteristic point and, as described below, is important for navigating into a passage.

As the distance values in the sectors refer to the virtual sensor's coordinate system, the algorithm tends to not detect passages that lie in the sideway parts of the polar sector map. To overcome this deficiency, two Cartesian sector maps are also used as data sources (see Figure 3, (2)). Due to the generic interface of the sector maps, the algorithm described above needs only simple modifications to process the data of Cartesian sector maps.

Algorithm 1. Detecting Passages.

```
searching_for = cFIRST_PEP; // the status of the search process
index_first_pep = 0; // the first PEP's index
index_point_in_passage = 0; // the index of a point within the passage
for (i = 0; i < n - 1; i++) do
    if (searching_for = cFIRST_PEP) then
        if (d[i] - d[index_first_pep] < t_0) then
            index_first_pep = i;
        else
            searching_for = cSECOND_PEP; index_point_in_passage = i;
    else // currently searching for second PEP
        if (d[index_point_in_passage] - d[i] >= t_1) then
            passages.add(new Passage(o[index_first_pep], o[i]);
            index_first_pep = i;
            searching_for = cFIRST_PEP;
        else
            // nothing to be done here
```

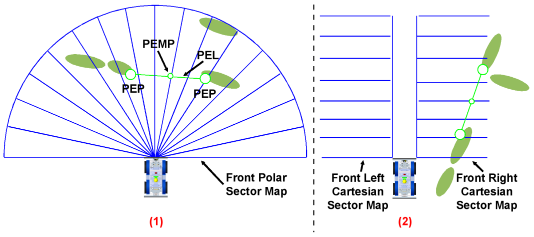


Figure 3: The sector maps used for passage detection.

It shall be remarked here that by using sector maps, the passage detection and evaluation algorithms do not have to access the complex grid map, but can operate on a much simpler data structure.

A temporal aspect was integrated by creating and updating a list of *persistent* passages: In every sensor processing cycle, the PEMP of each passage that is detected is compared to the PEMP of each persistent passage. Two passages are regarded as similar if the distance between their PEMPs is below a certain threshold. Taking into account the passages' PEPs was also considered, but experiments showed that the distance between two PEMPs is a suitable criterion for similarity. If a similar persistent passage is found, it is updated with the newly found passage, i.e. the persistent passage's PEPs and PEMP are set to the ones of the new passage. If not, the newly found passage is added to the list of persistent passage. Persistent passages that have not been seen for a certain amount of time are removed.

4.2 Passage Evaluation

The passage evaluation operates on the persistent passages. Its first step checks whether a passage fulfills the *basic requirements*, i.e. whether it is wide enough, has a minimum age, lies in a suitable direction, has a suitable orientation, and is long enough. A passage which fulfills these requirements is called *suitable passage*.

The first four aspects can be checked easily. A passage's width is calculated as the distance between its two PEPs. If a passage's entry is too narrow for the robot, it is obviously irrelevant for navigation. The age is calculated as time that has passed between the first and the latest detection. Passages which have not been seen for a minimum amount of time are ignored.

As passages shall be used to assist the navigation system in driving the robot to its target, a suitable passage shall lie in the direction of the target with respect to the robot. This is checked by comparing the orientation of two lines—one going from the robot to the target and one going from the robot to the PEMP. Furthermore, a suitable passage has to point approximately in the direction of the target. This is checked by first calculating the orientation of a line that starts at the PEMP and is perpendicular to the PEL. This orientation is then compared to the one of a line starting at the PEMP and going to the target. If the difference between the two orientations is below a certain threshold, the passage is considered to be well-oriented.

A VSP represented by a Cartesian sector map consisting of only one sector is used to “look” into the passage, measure the distance to the closest obstacle and thus estimate the passage's length (see Figure 4, (1)). Using a sector map as VSP instead of accessing the grid map directly facilitates the data access and spares the evaluation component more complex data processing.

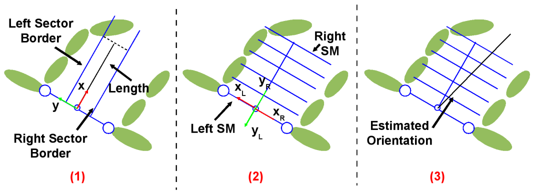


Figure 4: The sector maps for length and orientation estimation.

It is advisable to navigate the robot in a way that it reaches the PEMP with approximately the passage's orientation as this facilitates entering the passage. The orientation estimation described above does not take into account information about the presence of obstacles in the passage. Thus a more precise estimate is calculated for the passage that best satisfies the above criteria, the *relevant passage*. Two Cartesian sector maps are used to monitor the sides of the passage (see Figure 4, (2)). Each of their sectors stores information about the closest obstacle in the area it covers, i.e. it contains a local estimate of how far away the passage's border is. For each sector, the angle between the following two lines is calculated: (a) a line that starts at the PEMP, is perpendicular to the PEL and points into the passage, and (b) a line going from the PEMP to the sector's obstacle. The passage's orientation is calculated from the arithmetic mean of these angles, so its axis is pushed away from obstacles (see Figure 4, (3)). Of course this is only a rough estimate, but it can be calculated using the existing mechanisms, while other methods need more complex, specialized algorithms.

4.3 Integration

The component that detects and evaluates passages and navigates the robot to a PEMP has been implemented as behavior and integrated into the behavior-based system mentioned above. It is referred to as *passage behavior*. Whenever it detects a relevant passage, it gets active and sends the coordinates of the PEMP to the behaviors that drive the robot to given target coordinates. These behaviors are also used by the navigator, but when a passage is detected, the behavior gets active and overwrites the target pose provided by the navigator.

In some cases, an inhibition of the passage behavior is sensible. If the robot follows a wide path that leads to the target and has only few obstacles, the passage behavior would constantly detect a relevant passage and would try to guide the robot to its PEMP. If the number of visible obstacles is too low, the orientation calculations would produce poor estimates. Using these passages would result in a swinging robot motion although it would be possible to drive straight ahead. To cope with this problem, another behavior uses a wide Cartesian sector map to monitor the area in the direction of the target. If there is no obstacle, there is no need for using passages, so the monitoring behavior inhibits the passage behavior and thus returns control to the navigator.

The system's operation shall be illustrated taking up the example from section 1. At first, only a small part of the situation is visible to the robot (see Figure 5, (1)). A large passage is detected between the outer obstacles. The area in the direction of the target (green square) is free, so the passage is ignored. As soon as the free space to the right of the robot is recognized as indentation, the navigation system guides the robot towards the newly detected suitable passage instead of driving into the indentation (see Figures 5, (2) and (3)).

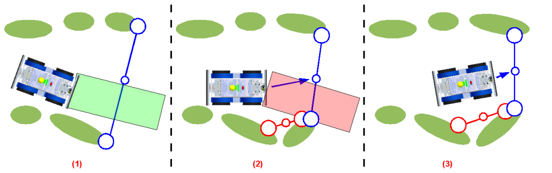


Figure 5: The operation of the passage behavior in the situation depicted in Figure 1.

5 EXPERIMENT AND RESULTS

For experiments the passage detection mechanism presented in this paper was integrated into the navigation software of the mobile off-road platform RAVON[1], a 4WD vehicle with the dimensions and weight of a city car, which is equipped with a variety of sensor systems (e.g. 2D laser scanners, a 3D laser scanner, stereo camera systems). Detailed information about the robot can be found in (Armbrust et al., 2009).

Several test runs have been conducted in the Palatinate forest, of which one shall be presented here as example[2]. RAVON started on a trail and was given a target location several hundred meters away. Its task was to drive there fully autonomously. Figure 7 shows a part of the run as pose trace. Red crosses symbolize obstacles that were classified by the sensor processing system as definitely not traversable. Vegetation that was probably traversable when driving cautiously is marked with green crosses. It shall be pointed out here that the robot drove the complete route depicted in the figure without any user action.

Figure 6: RAVON.

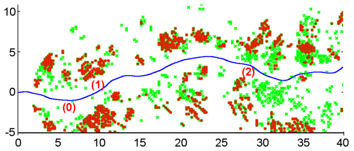


Figure 7: The pose trace with the checkpoints. The unit of length is meter.

Different passages were detected during the drive. For lack of space, only three significant checkpoints shall serve as examples here. For each of them, a visualization of the scrollable grid map filled with the 3D scanner's data is depicted (see Figure 8). The robot is situated in the map's center (blue square). The colored circles depict various types of obstacles (red: too high to be driven over; green: other types of obstacles with ground contact; pink: places where the robot could hit the ground). The direction to the target is visualized by a red line starting at the robot's center. Passages are visualized with circles for the PEPs and PEMPs, and lines for the PEL and for visualizing the first and second orientation estimates. For reasons of clarity, only two types of passages are displayed: suitable, but not relevant ones (green), and relevant ones (blue).

Shortly after the start, at checkpoint 0, the path made a slight turn to the left. The relevant passage was oriented to the left, following the path (see Figure 8, (0)). Although it did not lie exactly in the direction of the target and was oriented a bit away from it, it was still suitable. While this was a good first example of the passage behavior's operation, the combined action of the navigator and the anti-collision system would most likely have dragged the robot around the obstacles to its right.

At checkpoint 1 (see Figure 8, (1)), the deviation of the path's direction from the direction to the target was so large that following the navigator's drag would have led the robot into the underwood. The usage of the passage along with its orientation resulted in the robot driving a curve around the obstacles to its right.

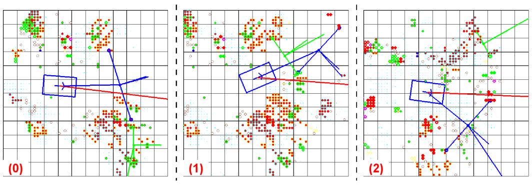


Figure 8: The situation at the three checkpoints.

At the last checkpoint (see Figure 8, (2)), the robot's motion would have probably been the following without the use of the passage: When driving towards the target, the robot would have gotten so close to the obstacles ahead of it that the collision avoidance behaviors would have had to turn it to the left or the right. Due to noisy sensor data and the lack of a larger scope, it is possible that the robot would have been turned to the left, thus driving straight into the underwood. At this point, backing off maneuvers would have to be conducted. By contrast, the detection of a suitable passage made the robot turn to the right. It avoided the obstacles before getting too close to them and stayed on the trail. Due to the obstacles

[1] Ravon: Robust Autonomous Vehicle for Off-road Navigation

[2] video showing large part of test available at: `http://rrlab.cs.uni-kl.de/robot-gallery/ravon/`

on the left border of the trail, no suitable passage was detected there—which is the desired behavior of the detection component.

6 CONCLUSIONS AND FUTURE WORKS

In this paper a concept for passage detection tailored to navigation in terrain with many obstacles and without a clear path was presented. Furthermore, a loosely coupled interaction of the passage detection facility with a topological navigator on the basis of passage orientation estimation was proposed. A suitable passage is negotiated by continuously passing the characteristic passage entry midpoint to a behavior-based point approacher. That way non-holonomic platforms—as presented in the experiments by example of the off-road robot RAVON—can enter even narrow passages without unneeded maneuvering.

The presented system is capable of leading the robot through complex terrain on the basis of a single target location avoiding local detours where possible. As only local terrain information is available on this layer, difficulties like dead ends still remain a problem. As a next step, the passage behavior shall provide detected passages to the navigator, which shall store them in terms of navigation-relevant spots. When a dead end is reached, the navigator shall tag passages leading to the current location as dead end entry points to avoid global detours in the future.

Furthermore, the system shall be extended so that other types of navigation-relevant places like very narrow ways (which require especially careful movements) or crossroads (which offer several options for the robot to proceed to the target area) are also detected. Exchanging information about these places with the higher navigation layer will support backtracking if following a path turns out to be unhelpful in getting to the target.

ACKNOWLEDGEMENTS

Team RAVON thanks the following companies for their technical and financial support: IK elektronik, Mayser, Hankook, MiniTec, SICK, DSM Computer, Hübner Giessen, John Deere, Optima, ITT Cannon, MOBOTIX, and Unitek.

REFERENCES

Alon, Y., Ferencz, A., and Shashua, A. (2006). Off-road path following using region classification and geometric projection constraints. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 689–696, New York, NY, USA. IEEE Computer Society (Washington, DC, USA).

Armbrust, C., Braun, T., Föhst, T., Proetzsch, M., Renner, A., Schäfer, B., and Berns, K. (2009). Ravon — the robust autonomous vehicle for off-road navigation. In *Proceedings of the IARP International Workshop on Robotics for Risky Interventions and Environmental Surveillance 2009 (RISE 2009)*, Brussels, Belgium. IARP.

Braun, T. and Berns, K. (2008). Topological edge cost estimation through spatio-temporal integration of low-level behaviour assessments. In *Proceedings of the 10th International Conference on Intelligent Autonomous Systems (IAS-10)*, Baden Baden, Germany.

Hong, T.-H., Rasmussen, C., Chang, T., and Shneier, M. (2002). Road detection and tracking for autonomous mobile robots. In *Proceedings of SPIE Aerosense Conference*, Orlando, FL, USA.

Lieb, D., Lookingbill, A., and Thrun, S. (2005). Adaptive road following using self-supervised learning and reverse optical flow. In *Robotics: Science and Systems*, pages 273–280, Cambridge, Massachusetts, USA.

Ranganathan, A. and Koenig, S. (2003). A reactive robot architecture with planning on demand. In *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1462–1468, Las Vegas, Nevada, USA.

Schäfer, H., Hach, A., Proetzsch, M., and Berns, K. (2008a). 3d obstacle detection and avoidance in vegetated off-road terrain. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 923–928, Pasadena, USA.

Schäfer, H., Proetzsch, M., and Berns, K. (2008b). Action/perception-oriented robot software design: An application in off-road terrain. In *IEEE 10th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, Hanoi, Vietnam.

Schröter, D. (2005). *Region & Gateway Mapping: Acquiring Structured and Object-Oriented Representations of Indoor Environments*. PhD thesis, Institut für Informatik der Technischen Universität München, München, Germany.

Thorpe, C., Carlson, J., Duggins, D., Gowdy, J., MacLachlan, R., Mertz, C., Suppe, A., and Wang, B. (2003). Safe robot driving in cluttered environments. In *11th International Symposium of Robotics Research*, Siena, Italy.

Wooden, D., Powers, M., MacKenzie, D., Balch, T., and Egerstedt, M. (2007). Control-driven mapping and planning. In *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3056–3061, San Diego, CA, USA.

PROMOTION IN RESCUE ROBOT
According to the Experience Gained by Participating in Bam Earthquake Rescue Operation

Pooya Heiraty, Aboozar Aghajani, Hojatollah Shirzadi Laskoukelayeh, Majid Zeraat Pisheh Fard
and Sayyed Mohammad Hosseini Monsef
Department of R&D, Sprooz International Co., Ltd., Flat No. 2, No.9, Eftekhar St.
Valiasr St., Next to Motahhari Junction, Tehran, Iran
Heiraty@sprooz.com, Aghajani@sprooz.com, Shirzadi@sprooz.com, M.Zeraatpisheh@sprooz.com, M_hm50@yahoo.com

Keywords: Four-Wheeled Robot, Locomotion, Control, Communication, Sensor, Navigation, Victim, Differential Drive, Map Generating, Bam Earthquake, Rescue Robot, Operator.

Abstract: Nowadays rescue robots are used in some rescue operations. Increasing the speed and accuracy of victim detection with sensors and equipment which are installed on the robot and yet increasing human safety factor of rescuers are among the advantages of using rescue robots.

By the experience of rescue operation in Bam earthquake and participating in some robocup competitions, a new four-wheeled robot has been designed which has highly operational capability. In initial part of this article, robot locomotion and controlling in different situations and the method of connection operator with robot are surveyed; then, sensors which are used in for navigation and victim detection are explained. At the end, the method of generating the map of robot’s movement route, which is very important for identifying the trapped victim's location in a rapid rescue operation, was studied.

1 INTRODUCTION

In December 2003, a severe earthquake destroyed one southern Persian city called Bam. This earthquake was the most devastating earthquake in the Middle East. Writers of this article who have been designing and building several robots, joined the rescue teams with their rescue robots and searching devices for detecting victims.

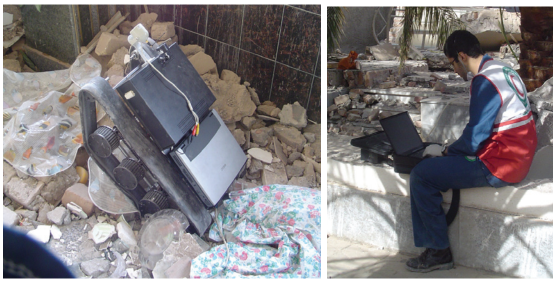


Figure 1: Robot navigation in Bam earthquake collapses.

By the experience of Bam earthquake, another rescue robot was designed and built whose characters would be describe in this article. This robot has high movement capability which could traverse the obstacles easily. The electronical and software parts of the robot have the ability of distance navigation, victim detection, and generating the map of victim places.

Individual Height adjustment of each wheels, carrying baby robot with ability of separating from the main robot and going through the small hole which the main robot could not goes, rapid ability for changing from four-wheeled to the track situation and mapping ability in both systems are among interesting idea which are used on this robot.

2 ROBOT LOCOMOTION AND OTHER MECHANISMS

The designed rescue robot is based on the four-wheeled robot mechanism. The design and production of it is so that the robot has a high capability of movement so that it can traverse the obstacles and unevenness easily. This robot is made using modern technology.

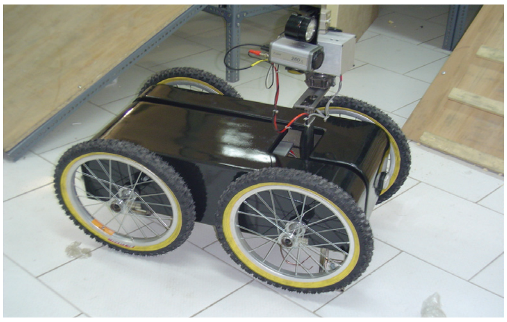


Figure 2: The robot.

The designed robot has four moving wheels and its dimensions are 87*50*80 cm. The weight of robot is about 32 kilograms. Four DC 12 V engines are used for the moving system of the robot each of them acts independently. The velocity of the robot is about 1 m/s and the rotation speed is 30 degrees per second.

The central computer which has the duty of processing, receiving and transmitting the data is an industrial PC (PC-104). Four packs of batteries of Ni-cadmium type are used to supply robot energy.

This robot uses the height adjustment system and the height of each wheel can be adjusted separately. By using this system the robot can easily climb the obstacles. Also in those cases where the patrol area is crowded, and there is a possibility for the bottom of the robot to collide with the obstacles, the height of robot could be increased to remove this problem. The changing range of the height in this robot is about 10 centimeters and the height is adjustable at the speed of two centimeters per second.

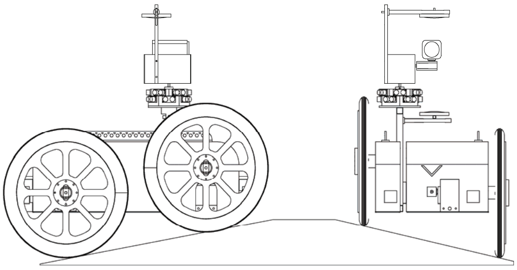


Figure 3: The height adjustment system.

In order to increase the moving ability, an innovative suspension system is used in this robot. This suspension system is devised in the form of a joint at the middle of the robot. This joint is capable of being locked at any angle and can help robot to traverse the obstacles and unevenness. The freedom degree of the joint is between -20 to +20. The time needed for the joint to be locked is about three seconds.

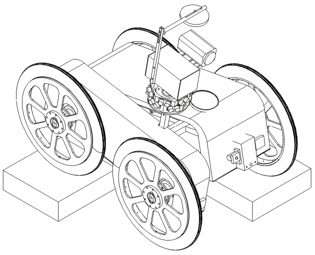


Figure 4: Joint suspension system of the robot.

To control the robot, three moving cameras are used. One of which is at the front the other at the back and the last one is located on the top of robot. The camera which is located on the robot is equipped with two Omni-directional mirrors above and under the camera. These mirrors give the robot the opportunity to have a visual angle of 360 degrees vertically.

Under the pedestal of the camera, there are 12 Sonar sensors which have the duty to help the robot in measuring the distance and also to obtain the map of environment.

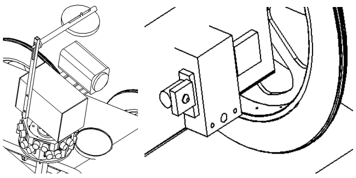


Figure 5: Pan-tilts and omni-directional mirrors.

The speed of the cameras at the back and front which are moved by RC Servos is about 50 degrees per second and the speed of the camera on the above is about 10 degrees per second.

For increasing the moving ability of the robot in special environments, and the ability of changing the moving system, four-wheeled tracked are used. It could be very rapid to change this situation.

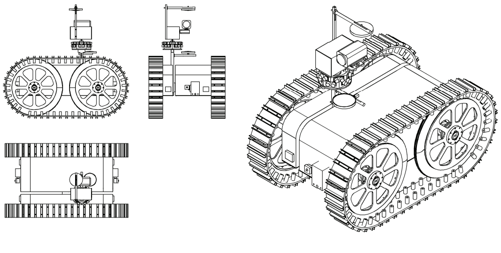


Figure 6: Track system.

In order to increase the searching ability of robot, a baby robot is used which is located inside the robot and it will exit whenever it is necessary. This baby robot is linked to the main robot through a cable.

The moving system of this robot is of the four-wheeled type, its dimensions are 15*10*8 cm and it weighs 1 KG. Two DC (12 V) engines are used in its moving system. The moving speed of the robot is about 0.2 m/s.

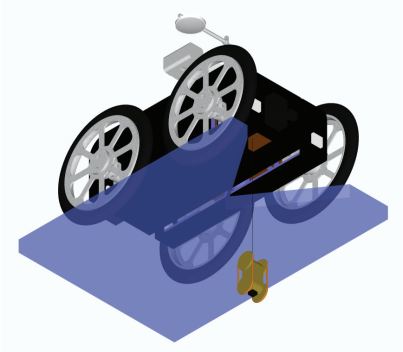


Figure 7: Baby robot.

3 CONTROL METHOD AND HUMAN-ROBOT INTERFACE

Controlling the robot is a partial autonomy. The major part of the controlling effort is performed by the operator and some movement decisions are made by the robot itself with the prior permit of the operator (which will be activated in the robot guiding software). Some of these decisions are: to automatic prevent collisions with the surrounding environment and to automatic return the robot to the starting point. But the software is designed so that the human decisions are considered prior to the robot decisions. For instance, in a narrow place the operator may deem it reasonable for the robot to collide with the walls so that it can pass the narrow entrance; therefore the orders of the operator are considered prior to the automatic decisions of the robot.

Regarding the mechanical specifications and the moving situation of the robot, three situations are defined for its movement: slow speed, medium speed and fast speed. The operator determines which of the above mentioned speeds shall be selected. But when the robot moves on a slope, the robot's movement will be set automatically according to the angle of the slope (which will be measured by ADXL330 accelerometer) and the software setting in which speed of the robot, while moving on the slope, is determined.

Moreover the PID controller is used to correct the mechanical errors and to adjust the exact speed of the robot in different situations.

4 COMMUNICATION

Regarding the fact that the robot is a PC base, W-Lan was used to establish connection between the robot and the computer (operator). Moreover, since in some regions it is impossible to establish wireless connections because of the high rate of noise, the capability of making connections through wire was added to the above system so that establishing the connection is practical even for long distances (several kilometres). A wire gathering system is devised inside the robot which could spread the long wire simultaneous with the robot's movement when connections shall be established through wire. In this way the wire will not hinder the robot while moving and it does not interfere with the robot's movement.

Table 1: W-Lan Specification.

Frequency	Channel/Band	Power (mW)
5.0 GHz - 802.11a	4	100

Through displacing a few simple jumpers which are devised on the robot and also through activation of the multiple choices exist in the provided software, we can easily determine the method for establishing connection with the robot.

To transmit the video pictures and voice, the 3W video transmitter with 2GHz frequency are used.

5 SENSOR FOR NAVIGATION AND LOCALIZATION

The baseline of navigation in a robot is to use graphical and video pictures sent by various cameras and Sonar system installed on the robot. Processing data, sent by cameras, is performed by the operator and the data sent by sonar system will be processed by a central microcontroller.

In addition to sonar system, several photo sensors are also installed on the robot's critical movement points which prevent robot colliding with the surrounding environment. According to the robot maximum speed, the identifying distance of obstacle by sensor is adjustable in a way that the robot will not collide the obstacles.

In order to locate the robot, a combination of data sent by various sensors are used which reduce the errors (that are inevitable). Increasing the rate of certainty in identification, grouping and also eliminating of the ambiguities & conflicts are among the benefits of this work. To perform the data combining system, it requires selecting and using the group of techniques harmoniously to achieve the best answer.

In order to apply the intelligent combination of data from the viewpoint of data processing, the high level method is used.

These sensors include:

- Four encoders which are mounted onto the motor shafts.
- Digital compass plus µ-metal to eliminate the noise of engines which could completely disable the digital compass sensor.
- the accelerometer sensor which obtains the robot's vertical angle in two vertical directions robot (using ADXL330 IC which most importantly is used to obtain the vertical angle (zx, zy) of the robot and to measure the height of the distance traversed by the robot).

To use the data provided by these sensors for illustrating the map, the following definitions shall be considered.

The break points: whenever the command for the robot's movement sent by operator through PC is interrupted or the Stop command is transmitted to the robot, that place will be considered as break point.

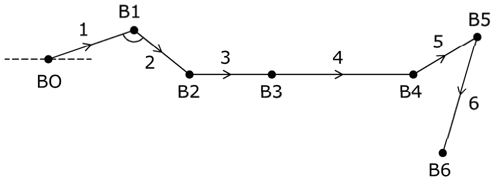


Figure 8: Break points for drawing map of robot's movement.

Note 1: the angle and the measure of the traversed distance from the previous break point and also the vertical distance of the robot from the horizontal line of the starting point will be desirable at the break point.

Note2: B3 is also defined as a break point here because the robot's movement in the straight direction is interrupted although it continued to move without changing its direction.

Note3: having the data about the robot's location (angle and the distance) at the break points, in an ideal condition we could illustrate the route map of robot in an accurate and simple way. This is the advantages of the break point definition.

The movement route: the robot's route of movement is obtained through connecting the vectors of break points.

Note4: the specifications stated in note 1, will be stored in the side memory of the robot in a special arrange. Whenever the investigating operation is finished, returning to the starting point could be automatically delegated to the robot by using the reversed combination of the stored data and real time data provided by the sensors simultaneously. (Obviously facing changes in the environment is unavoidable thus just in the ideal situation saved break points are used for returning to the start point and in real situation the real time data of sensors which could rectify the previous errors are used to obtain the new break points.)

Note5: Advantages of storing the data about break points in the side memory:

- The automatic returning of robot even in cases when the operator and the robot are disconnected.
- Access to the data in different steps of movement in order to perform the combining operation of the data provided by sensors (sensor fusion) to obtain a relatively accurate mapping.
- Access to the complete information of the route stored inside the memory so that in case the robot and PC are disconnected, the information could be transferred to PC (after returning of the robot).

6 SENSORS FOR VICTIM IDENTIFICATION

The most important method to identify the victim, is using pictures transmitted by the cameras which are installed on the robot (three cameras are installed on the robot but by using an analog switcher, the operator selects one of the pictures provided by one of these cameras to be transmitted.

And besides we can use:

- CO2 gas sensors for detecting the victim's breathing.
- Installation of highly sensitive stereo microphones on the robot helps the operator to find the victim.
- LM75 IC for measuring the temperature of the environment.
- The non-contact thermometer sensor model IL301, with D:S (30:1) for measuring the body temperature of the victim from a far distance.
- Also a motion detector is installed on the robot which is sensitive to any trivial movement in the environment.

7 ROBOT MECHANIC AND LOCALIZATION

Firstly, we describe the mathematical equations of localization differential mechanism and then by using those equations, we could generate four-wheeled robot and tracked robot equations with acceptable accuracy.

7.1 Differential Drive

Figure 9 shows a typical differential drive mobile robot. In this design incremental encoders are mounted onto the two drive motors to count the wheel revolutions. The robot can perform dead reckoning by using simple geometric equations to compute the momentary position of the vehicle relative to a known starting position.

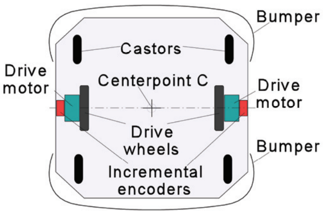


Figure 9: A typical differential drive mobile robot (top view).

20 Part I Sensors for Mobile Positioning For completeness, we rewrite the well-known equations for odometry below. Suppose that at sampling interval I the left and right wheel encoders show a pulse increment of N_L and N_R, respectively. Suppose further that:

$$C_m = \pi D_n / n C_e \quad (1)$$

where:
C_m = conversion factor that translates encoder pulses into linear wheel displacement
D_n = nominal wheel diameter (in mm)
C_e = encoder resolution (in pulses per revolution)
n = gear ratio of the reduction gear between the motor (where the encoder is attached) and the drive wheel.

We can compute the incremental travel distance for the left and right wheel, $\Delta U_{L,i}$ and $\Delta U_{R,i}$ according to:

$$\Delta U_{L/R,i} = C_m N_{L/R,i} \quad (2)$$

And the incremental linear displacement of the robot's center point C, denoted ΔU_i, according to:

$$\Delta U_i = (\Delta U_R + \Delta U_L)/2 \quad (3)$$

Next, the robot's incremental change of orientation was computed:

$$\Delta\theta_i = (\Delta U_R - \Delta U_L)/b \quad (4)$$

Where b is the wheelbase of the vehicle and ideally measured as the distance between the two contact points between the wheels and the floor.

The robot's new relative orientation θ_i can be computed from:

$$\theta_i = \Delta\theta_{i-1} + \Delta\theta_i \quad (5)$$

And the relative position of the center point is:

$$x_i = x_{i-1} + \Delta U_i \cos \theta_i \quad (6)$$
$$y_i = y_{i-1} + \Delta U_i \sin \theta_i \quad (7)$$

where:
x_i, y_i = relative position of the robot's counterpoint c at instant i.

7.2 Tracked Vehicles

Yet another drive configuration for mobile robots uses tracks instead of wheels. This very special implementation of a differential drive is known as skid steering and is routinely implemented in track form on bulldozers and army vehicles. Such skid-steer configurations intentionally rely on track or wheel slippage for normal operation (Figure 10), and as a consequence provide rather poor dead-reckoning information. For this reason, skid steering is generally employed only in tele-operated as opposed to autonomous robotic applications, where the ability to surmount significant floor discontinuities is more desirable than accurate odometry information. An example is seen in the track drives popular with remote-controlled robots intended for explosive ordnance disposal.

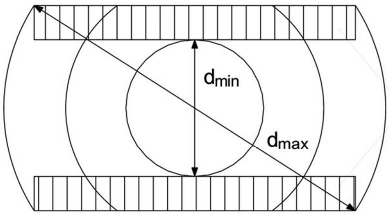


Figure 10: The effective point of contact for a skid-steer vehicle is roughly constrained on either side by a rectangular zone of ambiguity corresponding to the track footprint. As is implied by the concentric circles, considerable slippage must occur in order for the vehicle to turn.

7.3 Four-wheeled Vehicles

Obtaining the robot route map is one of the issues about which designer really concern. For this, there are simple mathematic equations. The robot route map equations parameters are described below. These equations are valid while wheels revolutions are equal, it means that the robot should go forward, backward or rotate at its place and could not use these equations in complicate movement. In rotation, robot changing angle is calculated and it has no movement.

Table 2: Parameters of movement route for four-wheeled robots.

Dimensional distance traveled by wheel for each encoder pulses	Ct
Effective width of robot divided by two	O
Incremental travel distance for the left and right wheel in the straight movement	ΔUR1 = ΔUL1
Incremental travel distance for the left and right wheel in u turn movement	ΔUR2 = ΔUL2
Number of pulses received by left and right encoder	NR,i = NL,i
Wheelbase of the vehicle, measured as the distance between the two contact points (wheels & floor)	b

The mathematic equations of robot route movement are described below:

$$\text{Sin}\ \alpha = O / b \quad (8)$$

$$C_m = \pi D_n / nC_e \quad (9)$$

$$\Delta U_{L1} = C_m\, N_{L,i} \quad (10)$$

$$\Delta U_{R1} = Cm\, N_{R,i} \quad (11)$$

$$C_t = C_m \times \text{Sin}\ \alpha \quad (12)$$

$$\Delta U_{L2} = C_t\, N_{L,i} \quad (13)$$

$$\Delta U_{R2} = C_t\, N_{R,i} \quad (14)$$

$$\Delta U_i = (\Delta U_{R1} + \Delta U_{L1}\ ) / 2 \quad (15)$$

$$\Delta \Theta_i = \Delta U_{R2} / b \quad (16)$$

$$\Theta_i = \Theta_{i\text{-}1} + \Delta \Theta_i \quad (17)$$

$$X_i = X_{i\text{-}1} + \Delta U_i \cos \Theta_i \quad (18)$$

$$Y_i = Y_{i\text{-}1} + \Delta U_i \sin \Theta_i \quad (19)$$

If compass sensor or gyroscope use more than encoders, $\Delta\Theta_i$ parameter is directly obtained from sensors and other movement route equations shall use without change.

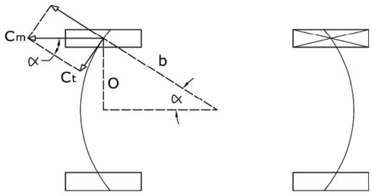


Figure 11: A typical four-wheeled drive mobile robot (top view).

8 MAP GENERATION

The map generation process, for illustrating the route of robot's movement is studied on this part. The recorded data, observed by cameras and also the piece of information were received by various sensors, illustrated on the map of robot movement (in form of a general report for the robot performance). This part consists of two parts called automatic mapping and manual mapping which the automatic mapping is chosen as the default. The work procedure is as follows:

Data about shaft encoders exist on the robot, and the data received through the digital compass sensor are transmitted continuously from robot. The computer firstly combines the received data and applies a series of error correcting algorithms to reduce the errors of shaft encoders (sensor fusion) and then the coordinates of the robot in each second will be obtained. By using these coordinates, two dimensional routes of robot's movement will be illustrated. Meanwhile, the data provided by gas and voice sensors, transmitted by the robot, will be saved in the computer and they will be illustrated on the map at the place which they were received. In addition, some tools are designed for the operator by which the operator can record its observations and insert information in the map.

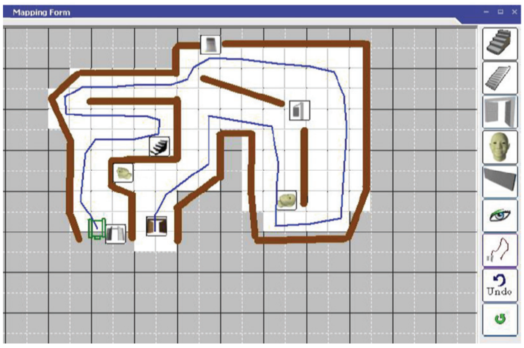


Figure 12: The scheme of the software used to illustrate the route map.

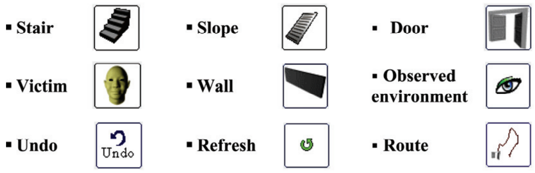


Figure 13: Symbols which are used in map generating.

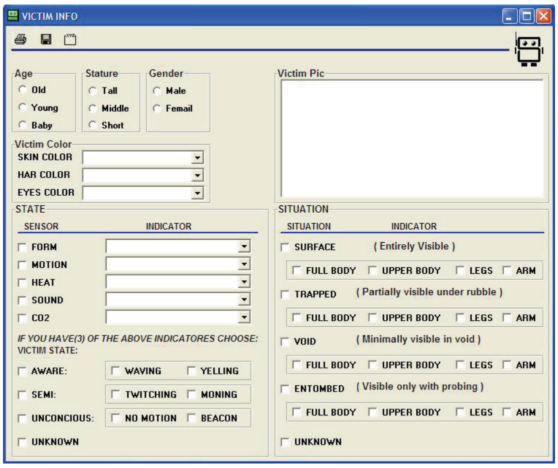
Whenever the operator realizes that at the place where the robot locates or at its surroundings, there is a stair, wall, door, slope, or a victim, it can locate a symbol of what it observes on the route map.

By using *observed environment* button and while navigating the robot, the operator will mark the places visited by the robot to prevent the repetitive visiting and then, saves time. For returning to the previous situation the operator just need to press *undo* button.

While working, whenever the robot and the computer are disconnected or whenever major errors in the coordinates data or the data provided by sensors are observed which may caused as a result of robot damage, the operator can select the manual mapping choice to illustrate the route. In this way, he can enjoy the benefits of software in controlling the robot and will face fewer errors.

In case that operator uses each of the above objects in a wrong place, the software has the ability to delete the object by a right click or to replace it by a left click. The places in white colour show that they are observed by the operator.

While observing an injured person, and placing its symbol on the route map, another form will appear for the operator in which the operator should enter the related data to be recorded and also to be used for the next reportages.



VICTIM INFO

Age
○ Old
○ Young
○ Baby

Stature
○ Tall
○ Middle
○ Short

Gender
○ Male
○ Femail

Victim Pic

Victim Color
SKIN COLOR
HAR COLOR
EYES COLOR

STATE

SENSOR INDICATOR
□ FORM
□ MOTION
□ HEAT
□ SOUND
□ CO2

IF YOU HAVE(3) OF THE ABOVE INDICATORES CHOOSE:
VICTIM STATE:

□ AWARE: □ WAVING □ YELLING
□ SEMI: □ TWITCHING □ MONING
□ UNCONCIOUS: □ NO MOTION □ BEACON
□ UNKNOWN

SITUATION

SITUATION INDICATOR
□ SURFACE (Entirely Visible)
□ FULL BODY □ UPPER BODY □ LEGS □ ARM
□ TRAPPED (Partially visible under rubble)
□ FULL BODY □ UPPER BODY □ LEGS □ ARM
□ VOID (Minimally visible in void)
□ FULL BODY □ UPPER BODY □ LEGS □ ARM
□ ENTOMBED (Visible only with probing)
□ FULL BODY □ UPPER BODY □ LEGS □ ARM
□ UNKNOWN

Figure 14: Form for recording the victim data.

9 CONCLUSIONS

Height adjustment and suspension systems enrich the robot to go through the entirely destructed buildings; however, it is not possible for it to cross some obstacles such as steps. Robot mapping system acts perfectly inside the buildings, but in the open areas or the rubble it needs to improve. In the improvement plan, these limitations must be removed. In its final version, laser scanner and radio positioning will be installed. The received information plus the other sensors data result in a better mapping achievement. It order to accelerate its movement on the steps and through the obstacles, some changes will be applied on the wheels and the motion system.

ACKNOWLEDGEMENTS

Prof. Caro Lucas.
Dr. Soleiman Farshid Fam.
Dr. Sayyed Hassan Mir Hosseini.

REFERENCES

Heiraty, P., Aghajani, A., Chitsazan, M., 2008. *The applied knowledge of designing and building robot.* Azarakhsh publishing company. Tehran, 1st edition.

Borenstein, J., Everett, H. H., Feng, L., 1996. *Where am I? Sensors and Methods for Mobile Robot Positioning.* The University of Michigan. Michigan, 1st edition.

Ko, Albert., Y. K. Lau, Henry., 2009. *Robot Assisted Emergency Search and Rescue System With a Wireless Sensor Network.* International Journal of Advanced Science and Technology field. Vol. 3, February.

Sandin, P. E., 2003. *Robot Mechanisms and Mechanical Devices Illustrated.* McGraw-Hill Company. New York, 1st edition.

Jacoff, A., Messina, E., Evans, J., 2000. *A Standard Test Course for Urban Search and Rescue Robots, Intelligent Systems Division.* National Institute of Standards and Technology.

Jacoff, A., Messina, E., Evans, J., 2000. *Experiences in Deploying Test Arenas for Autonomous Mobile Robots*, *Intelligent Systems Division*. National Institute of Standards and Technology.

RoboCup and AAAI rescue robot competition rules, 2008.

FATIGUE RECOGNITION USING EMG SIGNALS AND STOCHASTIC SWITCHED ARX MODEL

Hiroyuki Okuda, Fumio Kometani, Shinkichi Inagaki and Tatsuya Suzuki
Nagoya University, Furo-cho, Chikusa-ku, Nagoya, Japan
h_okuda@nuem.nagoya-u.ac.jp

Keywords: Human fatigue, Hybrid system, Electro Myo-Gram, Recognition.

Abstract: The man-machine cooperative system is attracting great attention in many fields, such as industry, welfare and so on. The assisting system must be designed so as to accommodate the operator's skill, which might be strongly affected by the fatigue. This paper presents a new fatigue recognizer based on the Electro Myo-Gram (EMG) signals and the Stochastic Switched ARX (SS-ARX) model which is one of the extended model of the standard Hidden Markov Model (HMM). Since the SS-ARX model can represent complex dynamical relationship which involves switching and stochastic variance, it is expected to show higher performance as the fatigue recognizer than using simple statistical characteristics of the EMG signal and/or standard HMM. The usefulness of the proposed strategy is demonstrated by applying to a peg-in-hole task.

1 INTRODUCTION

The man-machine cooperative system is attracting great attention in many fields, such as manufacturing, medicine, welfare and so on. The main purpose of assisting system is to reduce physical burden of the operator. Since a human skill is strongly affected by fatigue of the operator, the assisting system must be designed so as to accommodate with the change of skill characteristics caused by fatigue. To meet this requirement, fatigue must be detected and evaluated based on some quantitative manner. One of the basic ideas to evaluate the degree of fatigue is to measure physiological signals, such as the density of lactic acid in blood. This approach, however, requires the operator to stop the task, to take special examination and to be injured for sampling.

Recently, Electro Myo-Gram (EMG) signal is recognized as a promising one to measure the degree of physical fatigue without any special examination. EMG signal can be easily detected by only putting the probe on surface of the corresponding muscle. The relationship between the fatigue and the change of features such as Muscle Fiber Conduction Velocity (MFCV), magnitude, spectrum of EMG and so on are reported (Sadoyama and Miyano, 1981; Lippold et al., 1960; Arendt-Nielsen and Mills, 1988; D. K. Kumar and Bradley, 2003). Although these previous researches enable us to characterize the relationship between fatigue and the statistical characteristics of the EMG signal, their applications have been restricted in simple monotonous motion because those measures are developed under the Maximal Voluntary Contraction (MVC) condition. If the target task is more complex, fatigue recognition based on these features turns difficult cause of large variance of the measured signals in dynamic motion. To overcome this problem, a model-based approach, which can reflect the effect of the dynamic motion, must be exploited for the fatigue recognition.

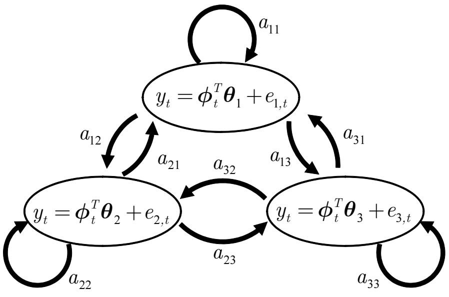

a_{11}
$y_t = \boldsymbol{\phi}_t^T \boldsymbol{\theta}_1 + e_{1,t}$
a_{12}
a_{21}
a_{32}
a_{13}
a_{31}
$y_t = \boldsymbol{\phi}_t^T \boldsymbol{\theta}_2 + e_{2,t}$
$y_t = \boldsymbol{\phi}_t^T \boldsymbol{\theta}_3 + e_{3,t}$
a_{23}
a_{22}
a_{33}

Figure 1: SS-ARX model (three states).

This paper presents a new fatigue recognizer based on the EMG signals and the Stochastic Switched ARX (SS-ARX) model. The SS-ARX model (Sekizawa et al., 2007) can be regarded as an extension of standard Hidden Markov Model (HMM) wherein each Auto Regressive eXogenous (ARX) model is embedded in each discrete state of the

HMM. In the proposed framework, we pay attention not only to the measured signal itself but also to the dynamic relationship between the EMG signals and motion, i.e. movement of the tool. Since the SS-ARX model can represent complex dynamics, which involves switching and stochastic variance, it is expected to show higher performance as the fatigue recognizer using standard HMM. This advantage is more emphasized when the target task becomes much more complex. Furthermore, we demonstrate the usefulness of the proposed strategy by applying to a peg-in-hole task. A comparison with standard HMM is also discussed.

2 STOCHASTIC SWITCHED ARX MODEL

SS-ARX model is defined as the system wherein one autoregressive exogenous (ARX) models is switched to the other one according to the state transition probability(Sekizawa et al., 2007). Figure 1 shows the SS-ARX model with three states.

This model can be regarded as the model wherein each ARX model is embedded in each discrete state of standard HMM. In the following, the definition and three important problems of the SS-ARX model are briefly reviewed (see detail in (Sekizawa et al., 2007)).

2.1 Parameters in SS-ARX Model

The parameters in SS-ARX model are specified as follows:

- S_i: Discrete state $(i{=}1,2,\cdots,N)$
- a_{ij}: State transition probability $(i{=}1,2,\cdots,N; j{=}1,2,\cdots,N)$
- π_i: Initial state probability $(i{=}1,2,\cdots,N)$
- θ_i: Parameters in ARX assigned to S_i $(i = 1,2,\cdots,N)$
- σ_i: Variance of equation error $e_{i,t}$ in ARX model assigned to $S_i(i{=}1,2,\cdots,N)$

N denotes the number of discrete states. We denote the set of parameters in the SS-ARX model by $\lambda{=}(\pi_i,a_{ij},\theta_i,\sigma_i)$.

2.2 Three Fundamental Problems

To address several fundamental problems listed below, the measured signal and its occurrence probability are defined for SS-ARX model as follows: First of all, a measured signal $o_{l,t}$ at time t is defined as combination of the output $y_{l,t}$ and the regressor $\psi_{l,t}$, that is, $o_{l,t}{=}(y_{l,t},\psi_{l,t})$. Where l is index of observed sequences, i.e. the index of trials. Then, its occurrence probability $b_i(o_{l,t})$ is defined by assumption of the Gaussian distribution of the equation error, and is given by

$$b_i(o_{l,t}) = \frac{1}{\sqrt{2\pi}\sigma_i}\exp\left\{-\frac{(\theta_i^T\psi_{l,t}-y_{l,t})^2}{2\sigma_i^2}\right\}. \qquad (1)$$

Based on these definitions, the following three fundamental problems can be addressed for SS-ARX model.

1. Evaluation problem
   The probability $P(O_l|\lambda)$ that the measured signal sequence $O_l{=}(o_{l,0},o_{l,1},\cdots,o_{l,t},\cdots,o_{l,T})$ occurs from the model $\lambda{=}(\pi_i,a_{ij},\theta_i,\sigma_i)$, that probability is called as likelihood, is calculated. This problem can be solved by applying Forward algorithm (Rabiner, 1989).

2. Decoding problem
   The most likely underlying state sequence $s = (s_{l,0},s_{l,1},\cdots,s_{l,t},\cdots,s_{l,T})$, which yields the measured signal sequence $O_l = (o_{l,0},o_{l,1},\cdots,o_{l,t},\cdots,o_{l,T})$, is found for the model $\lambda{=}(\pi_i,a_{ij},\theta_i,\sigma_i)$. This state estimation can be realized by applying Viterbi algorithm (Rabiner, 1989).

3. Estimation problem
   The model parameter $\lambda = (\pi_i,a_{ij},\theta_i,\sigma_i)$, which gives the highest occurrence probability for the measured signal sequence $O_l{=}(o_{l,0},o_{l,1},\cdots,o_{l,t},\cdots,o_{l,T})$, is estimated.

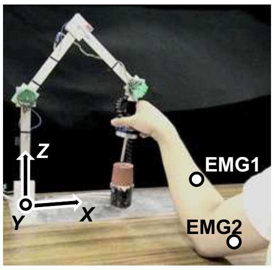


Figure 2: Data acquisition of peg-in-hole task.

The solution for problems 1 and 2 are same as ones for standard HMM. However, the parameter estimation algorithm for the SS-ARX model requires some extension to the one for standard HMM. The concrete parameter estimation algorithm for the SS-ARX model can also be derived based on the EM algorithm. The resulting parameter update law of $\boldsymbol{\theta}_i$ is given as follows:

$$\theta'_i = \left\{ \sum_{t=0}^{T} \sum_{l=1}^{L} k_l \psi_{l,t} \psi_{l,t}^T \alpha(l,i,t) \beta(l,i,t) \right\}^{-1} \times \left\{ \sum_{t=0}^{T} \sum_{l=1}^{L} k_l \psi_{l,t} y_{l,t} \alpha(l,i,t) \beta(l,i,t) \right\} \quad (2)$$

where k_l is defined by $1/P(O_l|\lambda)$, and $\alpha(l,i,t)$ and $\beta(l,i,t)$ are the forward probability and the backward probability of SS-ARX model, which resemble them of HMM respectively. Other update laws and its derivation are written in our previous study (Sekizawa et al., 2007).

Note that this model is applicable not only to the linear dynamics but also to a certain class of nonlinear dynamics, which may include switching mechanism. This benefit strongly motivates us to apply to the modeling and recognition of complex human skill.

3 EXPERIMENT SETUP AND DATA ACQUISITION

The fatigue recognizer is realized using SS-ARX model, and applied to the peg-in-hole task shown in Fig. 2. The peg-in-hole task is widely known as the typical skill which involves the switching in the dynamics caused by change of the contact configuration (Hirana et al., 2004; Ricker et al., 1996). In this work, the peg is supposed to move only on $X-Z$ plane. The mechanical arm in Fig. 2 provides no assisting force. As shown in Fig. 2, examinee holds the peg by grasping the end of the arm. There is no clearance between the rubber hole and peg. This implies that much force is required to accomplish the peg insertion. The examinees execute the task following the scenario depicted in Figure 3.

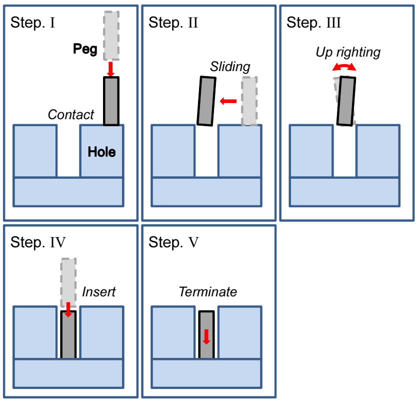


Figure 3: Typical motion of peg.

Table 1: Model parameters of examinee A (case of non-fatigue).

State transition probability				
a_{ij}	$i=1$	$i=2$	$i=3$	$i=4$
$j=i$	0.962	0.956	0.959	1
$j=i+1$	0.038	0.044	0.041	0

ARX-model parameters					
	θ_{i1}	θ_{i2}	θ_{i3}	θ_{i4}	σ_i
state1	0.404	0.134	0.042	0.549	0.005
state2	0.466	-0.166	0.031	0.472	0.006
state3	0.961	-0.088	0.006	-0.012	0.010
state4	0.189	-0.008	-0.014	0.014	0.004

Table 2: Model parameters of examinee A (case of fatigue).

State transition probability				
a_{ij}	$i=1$	$i=2$	$i=3$	$i=4$
$j=i$	0.978	0.923	0.950	1
$j=i+1$	0.022	0.077	0.050	0

ARX-model parameters					
	θ_{i1}	θ_{i2}	θ_{i3}	θ_{i4}	σ_i
state1	0.945	-0.091	0.006	0.052	0.007
state2	1.071	0.347	0.229	-0.200	0.013
state3	0.984	0.029	-0.056	-0.021	0.005
state4	0.180	0.002	0.040	0.003	0.003

Step. I The peg goes down vertically until it contacts with the surface of stage.

Step. II The peg slides to top of hole on the surface with keeping contact.

Step. III The operator uprights the peg for preparing the insertion.

Step. IV The peg is inserted firmly to the end of the hole.

Step. V Terminate.

Furthermore, the operators are well trained so as to be able to ignore the effect of experiences. The data for parameter estimation and recognition are acquired by the procedure shown in Fig. 4.

As a whole, twenty five data are acquired for verification of recognition. Examinees are expected to be more fatigued in the latter trials. Three examinees followed this procedure.

During the experiment, the position of the peg p_Z and two EMG signals at different locations shown in Fig. 2 (Extensor carpi ulnaris and Triceps brachii muscle) are measured every 1[msec]. The reason why these muscles are chosen is that these are well related with a force along with direction of peg insertion. The

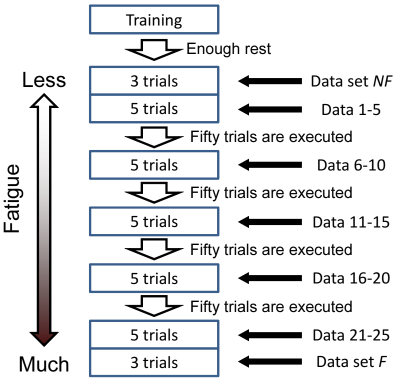


Figure 4: Data acquisition procedure.

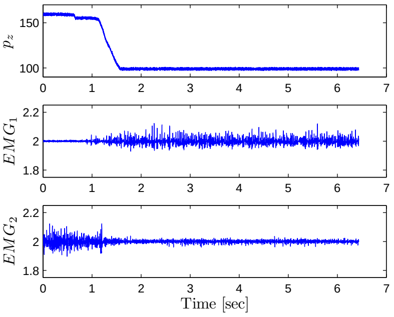
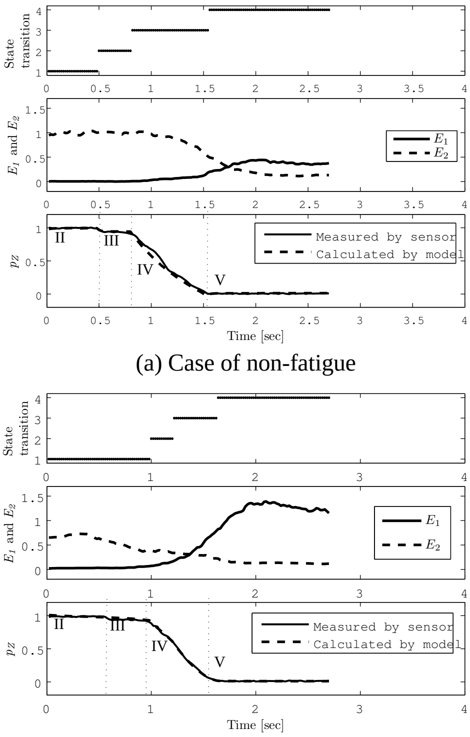


Figure 5: Example of signals EMG_1, EMG_2 and p_Z (examinee A, case of non-fatigue).



(a) Case of non-fatigue

(b) Case of fatigue

Figure 6: State transition, feature value of EMG, and p_Z(examinee A).

EMG signals are amplified with a gain of 1000 (Biometrics Ltd; SX230). Examples of measured EMG signals are shown in Fig. 5 together with p_Z.

In addition, the EMG signals are transformed to feature values by using the moving integral and normalized using the minimum and maximum values in trial 1 of Dataset NF, and also decimated by 20. In the following, E_1 and E_2 are used to denote the normalized feature values of the EMG_1 and EMG_2, respectively.

4 PARAMETER ESTIMATION RESULTS

In this section, the parameters of SS-ARX model are estimated based on learning data and the parameter update algorithm described in section 2. First of all, the signals and parameters appearing in the ARX model in the state k are defined as follows:

$$y_t = p_Z(t) \tag{3}$$

$$\psi_t = \{p_Z(t-1), E_1(t-1), E_2(t-1), 1\} \tag{4}$$

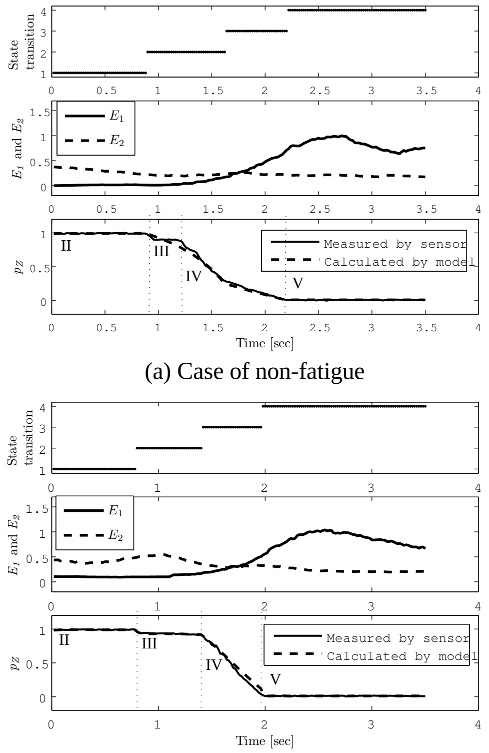
$$\theta_k^T = \{\theta_{k1}, \theta_{k2}, \theta_{k3}, \theta_{k4}\} \qquad (k = \{1, 2, \cdots, N\}) \tag{5}$$

$\boldsymbol{\theta}_k$ is the coefficient vector in the ARX model at state k. For reduction of the computational burden and complexity, the analysis is restricted in the motion along Z-axis which requires much more muscle force than other direction in the insertion task. Furthermore, the number of states is set to be $N = 4$ by try and error, and the left-to-right SS-ARX model is adopted.

The parameters of SS-ARX model of non-fatigue case, λ_{NF} is estimated using Data set NF. On the other hand, the parameters of SS-ARX model of fatigue case, λ_F is estimated using Data set F. 500 sets of initial parameters for the SS-ARX model were tested in the parameter estimation algorithm to find semi-optimal parameters. The parameter estimation results are shown in Tables 1 and 2.

Although we can see big difference in parameters between two models, this is partly because the physical meaning of the state in each model differs.

In Figs. 6 and 7, the estimated state transition, normalized feature values of EMG signals, and the comparison between the observed p_Z and calculated one using the estimated model are depicted from the top. The top figure represents the estimated state transition using Viterbi algorithm (Note that the state transition is not measured explicitly in our framework). The bottom figure indicates that the observed output



(a) Case of non-fatigue

(b) Case of fatigue

Figure 7: State transition, feature value of EMG, and p_Z(examinee B).

agree well with the calculated output. Thus, the accuracy of the SS-ARX model can be verified.

Also, the steps in the motion of the peg (II to V) are superimposed in the bottom figure. Intuitively, the state transition scenario must be associated with the switching occurred in the real task. Thus, we can see that the state definition of λ_F is different from one of λ_{NF}. In addition, we can see the big difference in the profiles of the E_1 and E_2 in the case of examinee A, however, the differences are not clear in the case of examinee B as shown in Fig. 7. In this case, it seems almost impossible to discriminate fatigue and non-fatigue cases only by looking at the profiles of E_1 and E_2 and the state transition in each case. However, Since the SS-ARX model explicitly includes the dynamic relationship between E_1, E_2 and p_Z, the fatigue recognition can be realized even in such a case as shown in the next section.

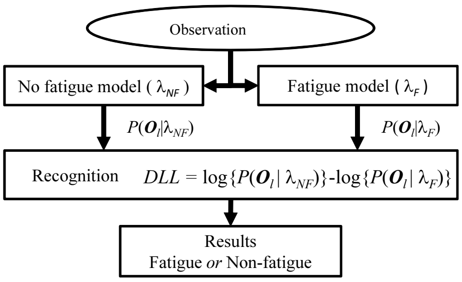


Figure 8: Proposed Recognition Scheme

5 FATIGUE RECOGNITION

In this section, fatigue is recognized using the two models estimated in the previous section. The log-likelihood values of the measured data over the two models are computed and compared to recognize the degree of fatigue of examinee. The illustrative diagram of the proposed scheme is shown in Fig. 8. The degree of fatigue of each examinee is evaluated by the difference of two log-likelihood values (denoted by *DLL*) given as follows:

$$\begin{aligned} DLL &= \log\left\{\frac{P(O_l|\lambda_{NF})}{P(O_l|\lambda_F)}\right\} \\ &= \log\{P(O_l|\lambda_{NF})\} - \log\{P(O_l|\lambda_F)\} \quad (6) \end{aligned}$$

where O_l is the measured sequence. $\log\{P(O_l|\lambda)\}$, which is log-likelihood of the measured sequence over the model, can be easily calculated by using Forward algorithm introduced in section 2.

We can see the clear tendency that the *DLL* goes down according to increase of the trial number. In addition, the trial when the *DLL* across zero is regarded as the turning point from 'non-fatigue trial' to 'fatigue trial'. Thus, the degree of fatigue of the examinee can be evaluated in quantitative manner.

Table 3: Correlation *r* between the *DLL* and trial number.

Exam.	SS-ARX	HMM
Exam.A	-0.80	-0.77
Exam.B	-0.83	-0.25
Exam.C	-0.77	-0.62
Exam.D	-0.62	-0.65
Exam.E	-0.93	-0.85

Finally, some discussions on the comparison with the standard HMM are given in the following. For the comparison, the number of states of the HMM were set to 8 (left-to-right structure), although the proposed SS-ARX model has 4 states. In the numerical experiments, the 4-state HMM did not work at all as the fatigue recognizer. The measured signals E_1, E_2 and p_Z were vector quantized by using 32 symbols. Here, a correlation of five examinees between the *DLL* and data number, which is regarded as a typical index to evaluate the relationship between the *DLL* and degree of fatigue, is calculated and shown in Table 3. This result implies that the growth of *DLL* calculated by

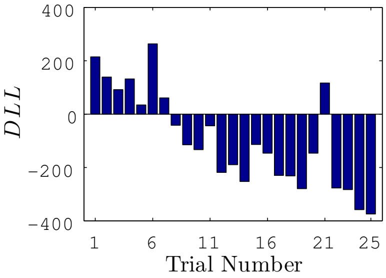


Figure 9: *DLL* of examinees A.

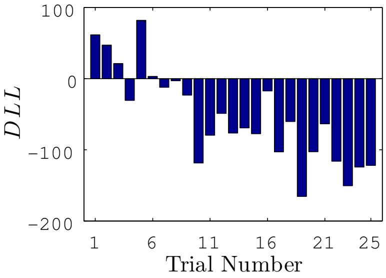


Figure 10: *DLL* of examinees B.

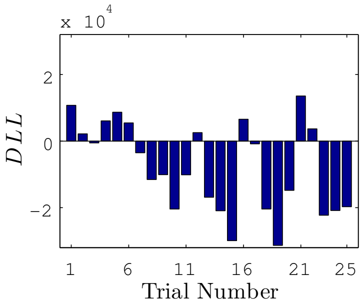


Figure 11: *DLL* of examinee B(in the case of HMM).

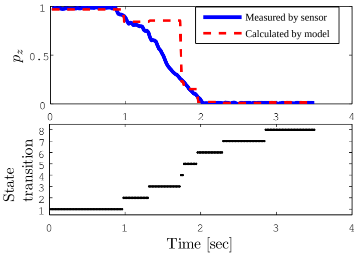


Figure 12: p_Z and state transition of trial 1 of examinee B(in the case of HMM).

SS-ARX has stronger correlation with the increase of trial number compared with that of standard HMM (except examinee D.) This comes from the fact that the HMM cannot capture the accurate dynamic characteristics underlying the measured signals compared with the SS-ARX model.

The recognition performances of the standard HMM and the SS-ARX model are compared using the profile of examinee B in the following. The recognition result of the HMM of examinee B is shown in Fig. 11. Also, the calculated p_Z and estimated state transition obtained by Viterbi algorithm are shown in Fig. 12.

In Fig. 11, obtained *DLL* does not related to trial number apparently. According to this result, it is almost impossible to discriminate between fatigue trials and non-fatigue trials. Therefore, the degree of fatigue does not seem to be recognized by standard HMM for examinee B.

6 CONCLUSIONS

This paper has presented a new fatigue recognizer based on the EMG signals and the stochastic switched ARX (SS-ARX) model. Since the SS-ARX model can represent complex dynamics which involves switching and stochastic variance, high performance as the fatigue recognizer was achieved. And the usefulness of the proposed strategy was demonstrated by applying to a peg-in-hole task. The design of adaptive assisting system which can accommodate with the change of skill characteristics caused by fatigue is our future work.

REFERENCES

Arendt-Nielsen, L. and Mills, K. (1988). Muscle fibre conduction velocity, mean power frequency, mean emg coltage and force during submaximal fatiguing contractions of human quadriceps.

D. K. Kumar, N. D. P. and Bradley, A. (2003). Wavelet analysis of surface electromyography to determine muscle fatigue.

Hirana, K., Suzuki, T., and Okuma, S. (2004). Formulation and motion planning of the peg-in-hole task with mixed logical dynamical system theory.

Lippold, O., Ledfean, J., and Vuco, J. (1960). The electromyography of fatigue.

Rabiner, L. (1989). A tutorial on hidden markov models and selected applications in speech recognition. In *Proc. of the IEEE, Vol. 77, No. 2, pp. 257-286.*

Ricker, S., Sarkar, N., and Rudie, K. (1996). A discrete-event systems approach to modeling dextrous manipulation.

Sadoyama, T. and Miyano, H. (1981). Frequency analysis of surface emg to evaluation of muscle fatigue.

Sekizawa, S., Inagaki, S., Suzuki, T., Hayakawa, S., Tsuchida, N., Tsuda, T., and Fujinami, H. (2007). Modeling and recognition of driving behavior based on stochastic switched arx model.

AN EXPERIMENTAL COMPARISON OF NONHOLONOMIC CONTROL METHODS

Kang-Zhi Liu
Department of Electrical and Electronics Engineering, Chiba University, Chiba, Japan
kzliu@faculty.chiba-u.jp

Keywords: Parking benchmark, Khennouf-Wit method, Astolfi's method, Sordalen-Egeland method, Ikeda-Nam-Mita method, Jiang's method, Liu-Sampei method.

Abstract: Although numerous nonholonomic control methods have been proposed, few is known about the advantages and disadvantages of each method. So in this paper an automatic parking system is used as a benchmark to test several typical nonholonomic control approaches experimentally. The emphasis is put on the applicability and control performance.

1 INTRODUCTION

Since the last decade nonholonomic control has been studied extensively and numerous methods have been proposed. However, no experimental comparison was reported up to date to the knowledge of the author. So in this paper the automatic parking problem will be used as a benchmark to test several typical nonholonomic control approaches experimentally.

The methods to be tested are (1) Khennouf-Wit method (H.Khennouf and Wit, 1996), (2) Astolfi's method (Astolfi, 2000), (3) Sordalen-Egeland method (Sordalen and Egeland, 1995), (4) Ikeda-Nam-Mita method (Ikeda et al., 2000), (5) Jiang's Method (Jiang, 2000) and (6) Liu-Sampei method (Sampei and et al., 1995; Liu et al., 2006).

The following specifications are used for comparison: (1) applicability to automatic parking subject to steering angle and parking space constraints, (2) safety, (3) convergence performance of each variable, (4) oscillatory behaviour during the parking control process.

2 MODEL AND EXPERIMENT SET-UP

The plant is a rear-drive 4-wheeled car illustrated in Fig. 1. Subject to the assumption that no side slip occurs, the kinematic model is described by

$$\dot{x} = u_1 \cos\theta,\ \dot{y} = u_1 \sin\theta,\ \dot{\theta} = u_1 \frac{1}{L} \tan\phi \tag{1}$$

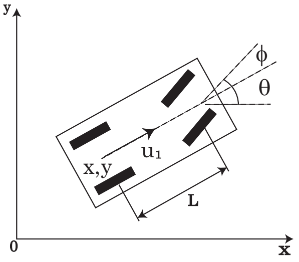


Figure 1: Model of 4-wheeled car.

in which L denotes the wheel base, (x,y) is the position of the center of rear wheel, θ is the orientation angle with respect to x axis. Further, ϕ and u_1 denote the steering angle and driving velocity respectively. Here

$$\eta = \tan\phi$$

and the driving velocity u_1 are regarded as the control input.

As real cars are subject to limitation of steering angle, the steering angle will saturate when the designed steering angle surpasses this limitation. That is, when the limit of steering angle is given by

$$\phi \in [-\phi_{\max}, \phi_{\max}],\ \phi_{\max} > 0 \tag{2}$$

the real input η becomes

$$\eta = \begin{cases} \eta^*, & |\phi| \le \phi_{\max} \\ sgn(\eta^*)|\eta_{\max}|, & |\phi| > \phi_{\max} \end{cases} \tag{3}$$

where $\eta^* = \tan\phi^*$ denotes the designed input and $\eta_{\max} = \tan\phi_{\max}$.

Applying the following variable transformation

$$z_0 = x,\ z_1 = y,\ z_2 = \tan\theta \tag{4}$$

as well as input transformation

$$\begin{cases} v_0 = u_1 \cos\theta \\ v_1 = \frac{1}{L}\eta^*(1+\tan^2\theta)u_1 \quad (\eta^* = \tan\phi) \end{cases} \tag{5}$$

to the system (1) leads to a 3rd order chained system:

$$\dot{z}_0 = v_0,\ \dot{z}_1 = z_2 v_0,\ \dot{z}_2 = v_1. \tag{6}$$

Most of the 6 methods are built with respect to this chained system.

The prototype motor car used in experiment is shown in Fig.2, in which the garage and road are indicated by the white lines.

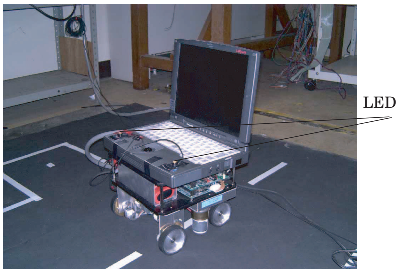


Figure 2: Experiment system.

Numerous parking experiment have been conducted and two sets of them will be shown (Table 1).

Table 1: Initial values.

	$x(0)$	$y(0)$	$\theta(0)$
Experiment 1	37[cm]	20[cm]	85[deg]
Experiment 2	41[cm]	16[cm]	33[deg]

In experiments, the designed driving velocity and steering angle are applied to their closed loop systems as reference input. Also in all figures of responses the solid, dotted lines show the measured data and the computed reference, respectively.

3 KHENNOUF-WIT METHOD

The input is given by

$$v = \begin{bmatrix} v_0 \\ v_1 \end{bmatrix} = 2f\frac{S(z)}{W(z)}\begin{bmatrix} -z_2 \\ z_0 \end{bmatrix} - k\begin{bmatrix} z_0 \\ z_2 \end{bmatrix} \tag{7}$$

in which $S(z)$ and $W(z)$ are

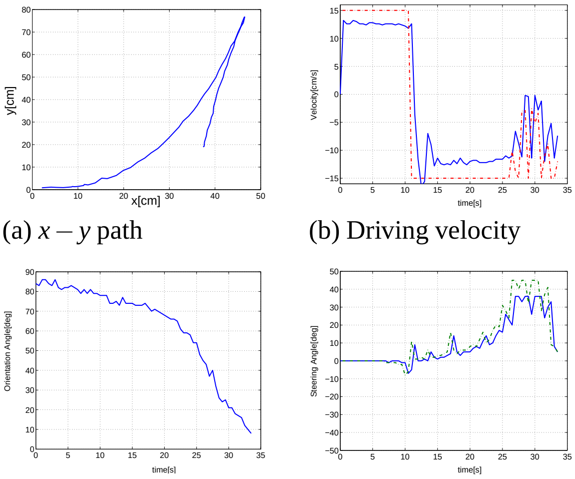
$$S(z) = z_1(t) - \frac{1}{2}z_0(t)z_2(t),\ W(z) = z_0^2 + z_2^2. \tag{8}$$

The closed loop system satisfies

$$W = W(z(0))\exp(-2kt),\ S = S(z(0))\exp(-ft) \tag{9}$$

when this input is applied to system (6). Therefore, W, S, z_0, z_1, z_2 converge to zeros.

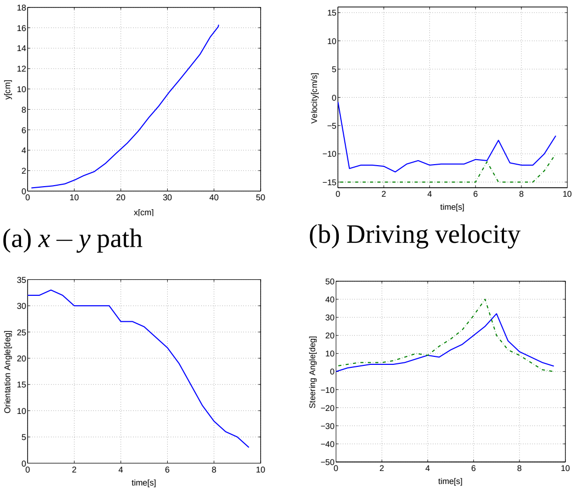
The results are illustrated in Fig. 3, 4. This method is good at controlling x, y, θ to zeros. However, in experiment 1 the car moved away from the garage to the position $(46, 77)$ before backing into the garage. This causes a safety problem. The phenomenon happens because the driving velocity is determined automatically and it can not be predicted where the car will make a turn. In this sense this method cannot be applied to parking control when the initial orientation angle is large.



(a) $x-y$ path (b) Driving velocity

(c) Orientation angle (d) Steering angle

Figure 3: Khennouf-Wit method: Experiment 1.



(a) $x-y$ path (b) Driving velocity

(c) Orientation angle (d) Steering angle

Figure 4: Khennouf-Wit method: Experiment 2.

4 ASTOLFI'S METHOD

This approach is proposed by Astolfi(Astolfi, 2000). First, the next coordinate transformation (σ-process)

$$y_1 = z_0,\ y_2 = z_2,\ y_3 = z_1/z_0 \tag{10}$$

is introduced to make the chained system (6) discontinuous. The transformed system is

$$\dot{y}_1 = v_0,\ \dot{y}_2 = v_1,\ \dot{y}_3 = \frac{y_2 - y_3}{y_1} v_0. \tag{11}$$

When

$$v_0 = -k y_1,\ \ k > 0 \tag{12}$$

is applied, y_1 is stabilized and

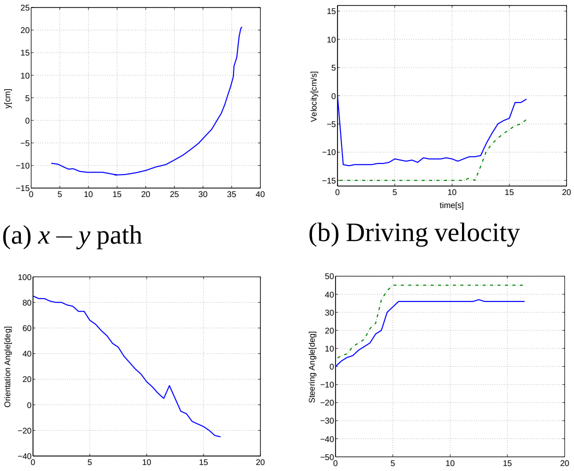
$$\begin{bmatrix} \dot{y}_2 \\ \dot{y}_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ -k & k \end{bmatrix} \begin{bmatrix} y_2 \\ y_3 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} v_1 \tag{13}$$

is controllable. Hence, a linear feedback

$$v_1 = -f_2 y_2 + f_3 y_3, \quad f_2 > k,\ f_3 > f_2 \tag{14}$$

can stabilize $y_2,\ y_3$. As a result, the original states z_1, z_2, z_3 are also stabilized.

The responses are shown in Figs. 5 and 6. As is seen from the measured data, the (x, y) path is pretty smooth, but the steering angle does not converge to zero. Further, the y coordinate moves to the opposite side when the initial orientation angle θ is over 80[deg] (Fig. 5), which may cause a safety problem.



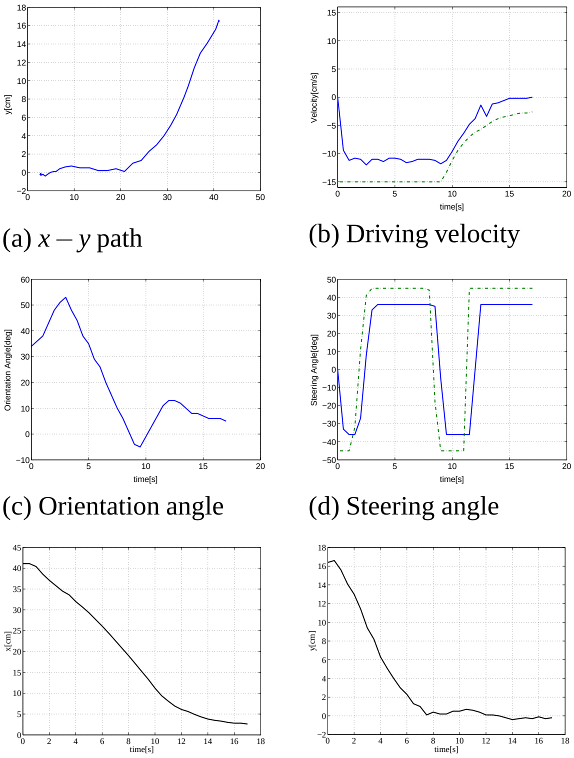
(a) $x-y$ path (b) Driving velocity

(c) Orientation angle (d) Steering angle

Figure 5: Astolfi's method: Experiment 1.

5 SORDALEN-EGELAND METHOD

Sordalen-Egeland method uses a periodic v_0 to drive the car and during this motion a time-varying $v_1(t)$ is applied to attenuate z_1, z_2 exponentially.



(a) $x-y$ path (b) Driving velocity

(c) Orientation angle (d) Steering angle

(c) x coordinate (d) y coordinate

Figure 6: Astolfi's method: Experiment 2.

The control law is as follows: let the period be T and $Z_2 = (z_1, z_2)^T$, set $v_0(t)$ in $iT \le t < (i+1)T$ as

$$v_0(t) = k(z(iT)) f(t) \tag{15}$$

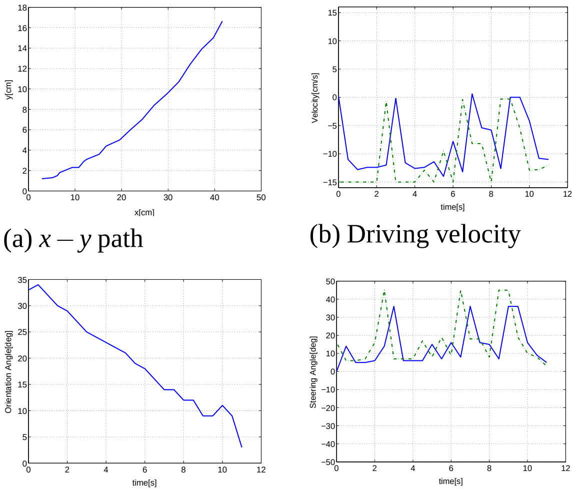
in which $f(t) = \frac{1}{2}(1 - \cos 2\pi t/T),\ G(Z_2) = c\|Z_2\|_2^{1/2}$, $k(z(iT)) = \mathrm{sat}(-[z_0(iT) + G(Z_2)\mathrm{sgn}(z_0(iT))]\beta,\ K)$.

Here $\beta > 0$,$K > 0$,$c > 0$ are design parameters and $\mathrm{sat}(x, K)$ is a saturation function of x with peak value K. On the other hand, in $iT \le t < (i+1)T$ $v_1(t)$ is determined as 0 when $z_0(iT) = 0$, and

$$v_1(t) = [\gamma_2, \gamma_3] Z_2,\ (z_0(iT) \neq 0) \tag{16}$$

where $\gamma_2 = -\lambda - f^3(t)\lambda$, $\gamma_3 = [-\lambda^2 f(t) - 2\lambda \dot{f}(t)] f(t)/k(z(iT))$ and $\lambda > 0$ is a parameter.

The results are shown in Fig. 7. The experiment failed when the initial orientation angle θ is around 80[deg] because the car moves out of the sensing range of PSD camera with an approximately 0[deg] steering angle. Moreover, the tuning of parameter is rather difficult since there are many parameters in the control law. The oscillation in response in intrinsic to this method. So this method is not suitable for parking control.



(a) $x-y$ path (b) Driving velocity

(c) Orientation angle (d) Steering angle

Figure 7: Sordalen-Egeland method: Experiment 2.

6 IKEDA-NAM-MITA METHOD

The control procedure of this method is divided into two steps. In step 1, the input is determined as

$$v_0 = -\lambda_2 \frac{z_1}{z_2},\ v_1 = -\lambda_1 z_2,\ \lambda_2 > \lambda_1. \tag{17}$$

Then the closed loop system becomes

$$\dot{z}_0 = -\lambda_2 \frac{z_1}{z_2},\ \dot{z}_1 = -\lambda_2 z_1,\ \dot{z}_2 = -\lambda_1 z_2 \tag{18}$$

and $(z_2, z_1) \to 0$. Note that z_1/z_2 is bounded if $\lambda_2 > \lambda_1$. In step 2, the input is switched to

$$v_0 = -\lambda_3 z_0,\ v_1 = -\lambda_1 z_2 \tag{19}$$

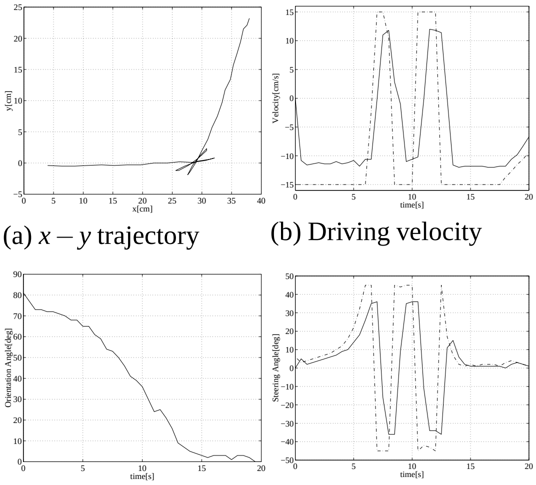
once z_2 is sufficiently close to zero. The corresponding closed loop system changes to

$$\dot{z}_0 = -\lambda_3 z_0,\ \dot{z}_1 = -\lambda_3 z_0 z_2,\ \dot{z}_2 = -\lambda_1 z_2 \tag{20}$$

and $(z_0, z_2) \to 0$. In this process z_1 will not deviate far away from 0 because the initial values of z_1, z_2 are sufficient small due to the control in step 1. In the experiment, the input is switched back to step 1 if z_2 deviates far away from zero due to disturbance.

In the experiments, the input of step 1 is used if $|\theta| \leq 0.1$[rad]. Otherwise the input of step 2 is used.

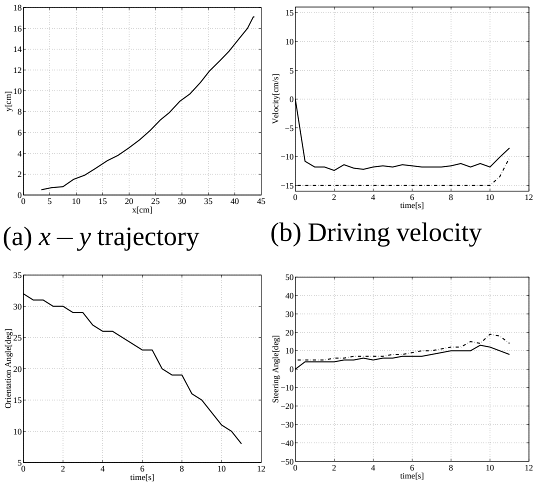
As can be seen from Fig.8 and 9, this method is able to control x, y, θ to zeros pretty good. But in the first experiment, the car moves back and forth four times around the x axis. However, compared with Khennouf-Wit Method the change of the direction of the driving velocity occurs only in positions that are far away from the origin, it is not so severe a drawback.



(a) $x-y$ trajectory (b) Driving velocity

(c) Orientation angle (d) Steering angle

Figure 8: Ikeda-Nam-Mita method: Experiment 1.



(a) $x-y$ trajectory (b) Driving velocity

(c) Orientation angle (d) Steering angle

Figure 9: Ikeda-Nam-Mita method: Experiment 2.

7 JIANG'S METHOD

In (Jiang, 2000) Jiang Proposed a robust exponential regulation method for a class of nonholonomic systems with uncertainty.

First, a rotation of (x, y) coordinate is introduced to avoid singularity in the transformation to canonical form

$$\begin{aligned} x_0 &= \theta,\ x_1 = x\sin\theta - y\cos\theta \\ x_2 &= x\cos\theta + y\sin\theta,\ u_0 = u_1 \frac{1}{L}\tan\phi. \end{aligned} \tag{21}$$

This transformation brings (1) into

$$\dot{x}_0 = u_0,\ \dot{x}_1 = x_2 u_0,\ \dot{x}_2 = u - x_1 u_0. \tag{22}$$

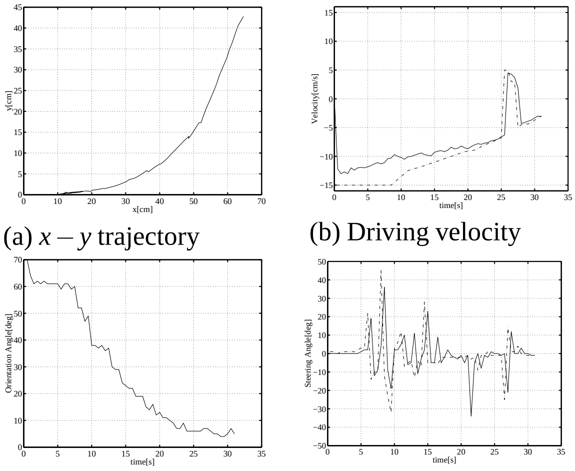
Further, a state scaling

$$z_1 = \frac{x_1}{x_0}, \quad z_2 = x_2 \tag{23}$$

is introduced. Then based on backstepping, the following control input are obtained:

$$\begin{aligned} u_0 &= -\lambda_0 x_0 &(24)\\ u_1 &= -[\lambda_2 + \lambda_0(\lambda_1+1) + \frac{\lambda_0}{4}(x_0^2 - 1 + \lambda_1 + \lambda_1^2)] \\ &\quad \times (z_2 - (\lambda_1+1)z_1), \quad \lambda_0,\ \lambda_2 > 0,\ \lambda_1 > 1. \end{aligned}$$

It is clear from the Figs. *Please place* `\label` ***after*** `\caption` and *Please place* `\label` ***after*** `\caption` that the car moves back and forth near the garage which may cause safety problem. Also, the steering angle is quite oscillatory and so is the orientation angle as its consequence.



(a) $x-y$ trajectory (b) Driving velocity

(c) Orientation angle (d) Steering angle

Figure 10: Experiment 1(x(0)=64.7[cm], y(0)=42.8[cm], θ(0)=70[deg]).

8 LIU-SAMPEI METHOD

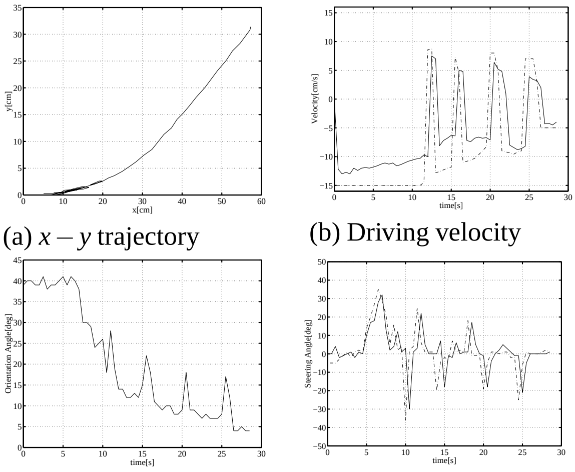
This method(Liu et al., 2006) evolved from Sampei's method(Sampei and et al., 1995). Its essence is to attenuate the orientation angle θ and y coordinate while drive the car back and forth on the allowed road, then finally park the car into the garage. A distinguishing feature of this method is that the driving velocity can be determined freely.

Let z_2^* be

$$z_2^* = -C_1 sgn(v_0) z_1,\ C_1 > 0 \tag{25}$$

and determine the control input v_1 as

$$v_1 = -C_1 z_2 |v_0| - z_1 v_0 - C_2(z_2 - z_2^*)|v_0|. \tag{26}$$



(a) $x-y$ trajectory (b) Driving velocity

(c) Orientation angle (d) Steering angle

Figure 11: Experiment 2(x(0)=57.3[cm], y(0)=31.4[cm], θ(0)=39[deg]).

Then the derivative of Lyapunov function $V = \frac{1}{2}z_1^2 + \frac{1}{2}(z_2 - z_2^*)^2$ satisfies $\dot{V} = -C_1 z_1^2 |v_0| - C_2(z_2 - z_2^*)^2 |v_0|$ which is negative semidefinite. Hence, the convergence of (z_1, z_2) is guaranteed.

The input v_0 is selected as follows: (1) Take v_0 arbitrarily if $z_1^2 + (z_2 - z_2^*)^2 > \gamma$. (2) $v_0 = -sgn(z_0)|U|$ when $z_1^2 + (z_2 - z_2^*)^2 < \gamma$ so as to stabilize z_0. Here, U is given by

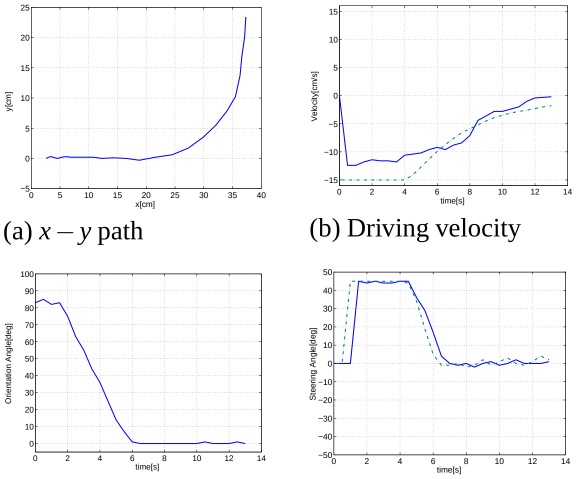
$$U = \begin{cases} u_{\max}\beta\cos(\theta) & if\ \sqrt{x^2+y^2} \geq \beta u_{\max} \\ \sqrt{x^2+y^2}\beta\cos(\theta) & if\ \sqrt{x^2+y^2} < \beta u_{\max} \end{cases} \tag{27}$$

$u_{\max}$ is the maximum of driving velocity and β is a deceleration factor.

The experiment data are illustrated in Fig. 12, 13. This method can stabilize x, y, θ from any initial state and provides the best performance for parking control.

9 CONCLUDING REMARKS

The applicability of 6 typical control methods for chained system has been tested experimentally by using an automatic parking benchmark. The results indicate that Astolfi's method(Astolfi, 2000) and Ikeda-Nam-Mita method(Ikeda et al., 2000) may be applied to parking control when the initial orientation angle is not too big. It is noted that in Astolfi's method the steering angle does not converge to zero. Liu(Liu et al., 2006)-Sampei(Sampei and et al., 1995) method is applicable to any situations. Meanwhile, Khennouf-Wit method and Jiang's method should be



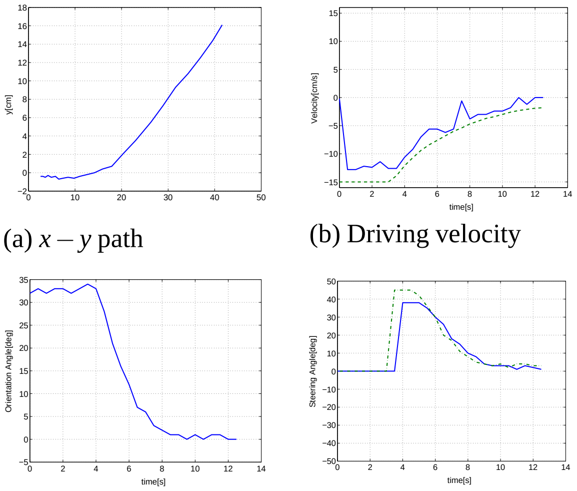
(a) $x-y$ path

(b) Driving velocity

(c) Orientation angle

(d) Steering angle

Figure 12: Liu-Sampei method: Experiment 1.



(a) $x-y$ path

(b) Driving velocity

(c) Orientation angle

(d) Steering angle

Figure 13: Liu-Sampei method: Experiment 2.

used with care due to safety concern. Sordalen-Egeland method is not suitable for parking control under steering angle and parking space limitation.

It is also worth noting that all methods, except Liu-Sampei and Ikeda-Nam-Mita methods, guarantees asymptotic stability, but their performances are not as good as those of Liu-Sampei and Ikeda-Nam-Mita methods. The author feels that the degrading of performance is caused by killing the freedom of control (the driving velocity) in order to prove the asymptotic stability. In contrast, both Liu-Sampei method and Ikeda-Nam-Mita method use switching of control input which provides the control flexibility and leads to better performance, although it is very difficult to show their asymptotic stability. The author strongly believe that control design based on asymptotic/exponential stability point of view is not suitable for this class of control problems, the emphasis should be put on improving the performance instead.

REFERENCES

Astolfi, A. (2000). Discountinuous control of nonholonomic systems. *Systems and Control Letters*, 27(1):37–45.

H.Khennouf and Wit, C. (1996). Quasi-continuous exponential stabilizers for nonholonomic system. *Proc. of 13th IFAC World Congress*, 2b(174):49–54.

Ikeda, T., Nam, T.-K., and Mita, T. (2000). Nonholonomic variable constraint control of free flying robotics and its convergence (in japanese). *Journ. of Robotics Society of Japan*, 18.

Jiang, Z.-P. (2000). Robust exponential regulation of nonholonomic systems with uncertainties. *Automatica*, pages 189–209.

Liu, K.-Z., Dao, M.-Q., and Inoue, T. (2006). An exponential ε-convergent control algorithm for chained systems and its application to automatic parking. *IEEE Trans. on Control Systems Technology*, 14(6):1113–1126.

Sampei, M. and et al. (1995). Arbitrary path tracking control of articulated vehicles using nonlinear control theory. *IEEE Trans. on Control Systems Technology*, 3(1):125–131.

Sordalen, O. and Egeland, O. (1995). Exponential stabilization of nonholonomic chained systems. *IEEE Trans. Automat. Control*, 40:35–49.

COMPARISON BETWEEN MEXICAN HAT AND HAAR WAVELET DESCRIPTORS FOR SHAPE REPRESENTATION

Adnan Abou Nabout and Bernd Tibken
Chair of Automatic Control, Faculty of Electrical, Information and Media Engineering
University of Wuppertal, Wuppertal, Germany
nabout@uni-wuppertal.de, tibken@uni-wuppertal.de

Keywords: Image Processing, Shape Representation, Wavelet Transformation, Wavelet Descriptors.

Abstract: The wavelet transformation is a well known method in several engineering fields. In image processing and pattern recognition the wavelet transformation is used for the recognition of object shapes by deriving so called wavelet descriptors. In this context the Mexican Hat as well as the Haar function were used as mother wavelets. To derive wavelet descriptors the methods use a periodical angle function derived from the contour polygon. The angle function describes an object shape by calculating the angle changes along the object contour beginning from a given starting point. Since object shapes are described by polygons, the angle function is step-shaped and therefore it includes discontinuity at the existing polygon corners. This causes big changes of the Haar wavelet descriptors if the positions of the polygon corners change due to small changes of the object shape. Such changes can be caused at least by digitalization or binarization errors. The Mexican Hat wavelet descriptors are more adapted and suffer however from small changes. In this paper we present the results of the comparison between both methods in there accurateness of describing object shapes.

1 INTRODUCTION

The automatic recognition of objects, their classification or representation is a very important task in the field of image processing and pattern recognition. In particular, the recognition of object shapes is a commonly needed process in many applications in this area (Grenander, Chow and Keeman, 1991), (Belongie, Malik and Puzicha, 2001), (Fergus, Perona and Zisserman, 2003). The recognition of weed species classes in agricultural applications is one of the interesting examples for the importance of object recognition using shape information, since the shapes of weed species change according to the growing stages of the weeds (Nabout, Nour Eldin, Gerhards, Su and Kühbauch, 1994).

The most known method for the above mentioned task uses the Fourier analysis (Zahn and Roskies, 1972). The method is used to obtain a number of coefficients, called Fourier descriptors (FD) to describe a given object shape. The recognition can be done through the comparison between the FD of the unknown object with those of the stored object samples using minimum distance or Fuzzy methods (Nabout, 1993). In (Nabout and Tibken, 2004, 2005, 2007 and 2008) we established an alternative method using Wavelet transformation. Similar to the FD we applied Wavelet descriptors WD using the Mexican Hat or Haar function as mother Wavelet.

In this paper we conclude the derivation of wavelet descriptors for both cases and compare the results of these different implementations in order to find out the appropriate Wavelet building set. To represent a given object shape we will show the right way how to apply a periodical angle function using the polygon data of a given object shape. This angle function must be free from any singularity which might arise due to object rotations. For that reason the paper shows the derivation of the angle function for a simple geometric object. To obtain a suitable number of WD we normalized the angle function over the interval $[0 - 2\pi]$ and derived a wavelet building set in the same interval. The results are shown on the basis of different characters to illustrate the different steps. We also present some results related to the recognition of weed species in plant fields. The paper is organized as follows:

Section 2 addresses the derivation of the angle function and describes the problem of singularity. Section 3 introduces the continuous Wavelet transformation. The derivation of the WD using Mexican Hat as well as Haar function is presented in section 4. In section 5 the results of applying the new method are demonstrated and discussed. In this context the minimum distance approach is described, which is used to compare two different WD sets. In section 6 the starting point problem is discussed.

2 SHAPE DESCRIPTION USING AN ANGLE FUNCTION

To derive an angle function we use the polygon information of a given object shape derived after contour extraction and approximation (Nabout, 1993). Fig. 1 shows the example of a triangle shaped object (a) and his derived angle (red) and periodical angle functions (green) (b).

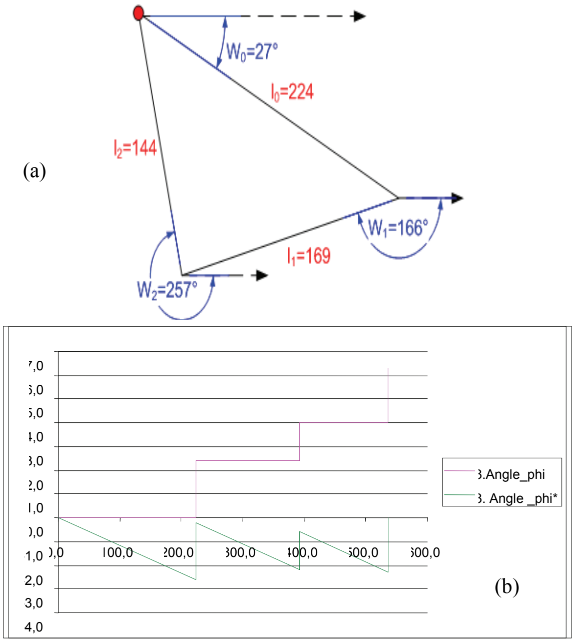


Figure 1: Polygon of triangle shaped object (a) and the angle functions of the given shape (b).

To obtain the angle function we calculate the angle differences between the absolute angles at every position on the given polygon and the absolute angle of the polygon at the starting point. It should be noted that the starting point depends on the object's position and orientation in the image. The calculation of the angle differences must take into account that the absolute angles can change according to the object rotation. In some cases small object rotations cause significant changes in absolute angles. We denote this problem as singularity problem (Nabout, Tibken, 2008).

To avoid any singularity, we calculate all absolute angles of the polygon edges with respect to the x-coordinate as given in Fig. 1. The absolute angles of a polygon edge $\overline{P_i P_{i+1}}$ are always positive and can be computed using the polygon data as follows:

```
$\mathrm{if}(x_{i+1} != x_i)$
    $\mathrm{if}(y_{i+1} != y_i)$
    {
        $\omega = \tan^{-1}[(y_{i+1} - y_i)/(x_{i+1} - x_i)]$
        $\mathrm{if}(x_{i+1} < x_i)\ \omega = \omega + \pi$
        $\mathrm{else\ if}(y_{i+1} < y_i)\ \omega = \omega + 2\pi$
    }
    $\mathrm{else\ if}(x_{i+1} > x_i)\ \omega = 0, \mathrm{else}\ \omega = \pi$
$else\ if(y_{i+1} > y_i)\ \omega = \pi/2, else\ \omega = 3\pi/2$
```

where (x_i, y_i) and (x_{i+1}, y_{i+1}) are the coordinates of the polygon corners P_i and P_{i+1}. To obtain the angle function we then calculate the angle differences as

```
$\alpha_i = \omega_i - \omega_0$
$\mathrm{if}(\alpha_i \geq \pi\ \&\ \alpha_i \leq 2\pi)\ \alpha_i = 2\pi - \alpha_i$
    $\mathrm{else\ if}(\alpha_i \geq -2\pi\ \&\ \alpha_i \leq -\pi)\ \alpha_i = -2\pi - \alpha_i$
$\mathrm{else}\ \alpha_i = -\alpha_i$
```

where ω_i is the absolute angle of the polygon edge $\overline{P_i P_{i+1}}$ and ω_0 the absolute angle of the first polygon edge $\overline{P_0 P_1}$. According to this definition, we obtain for the example in Fig. 1.a the angle differences $(0^0, 139^0, 230^0)$, which correspond to the radian values $(0, 2.4, 4.0)$ as shown in Fig. 1.b.

The angle differences are negative in clockwise direction. The angle function f(l) (red colored function in Fig. 1.b) can be derived by calculating the value of α for every position specified by the parameter l, where l is the contour length measured from the starting point up to the current contour position. The derived angle function is defined on the interval $[0, L]$, where L is the total length (circumference) of the given contour polygon and can be scaled on the $[0, 2\pi]$ -interval using the following parameter transformation:

$$\mathbf{l \rightarrow t, \quad t = 2\pi l/L} \tag{1}$$

with

$$\mathbf{f^*(t) = f(Lt/2\pi) - t} \tag{2}$$

we receive a periodical angle function (green colored function in Fig. 1.b) $f^*(t)$ with a period of 2π.

3 WAVELET TRANSFORMATION

Similar to the FT, the WT uses elementary functions, called wavelets, to describe a given signal. In contrast to the FT, which uses harmonic functions with different frequencies, the WT uses only one basis wavelet (mother wavelet) to derive the reconstruction signals (Daubechies 1992). Through dilatation, compression and shifting of the mother wavelet, we derive new variants of this signal which together constitute the so-called wavelet building set. Equation (3) describes the general derivation of wavelets $\Psi^{\mathrm{a,b}}(\mathrm{t})$ from the mother wavelet $\Psi(\mathrm{t})$ (Daubechies, 1992).

$$\Psi^{a,b}(t) = |a|^{-\frac{1}{2}}\,\Psi\left(\frac{\mathrm{t-b}}{\mathrm{a}}\right) \tag{3}$$

where a is the compression or dilatation parameter and b is the shifting parameter. Fig. 2 shows the mother wavelet based on the Haar function and some derived variants resulting from compression, dilatation and shifting using (3). Fig. 3 shows the equivalent Mexican Hat functions.

The function Ψ can be scaled over the interval $[0, 2\pi]$ similar to the periodic angle function.

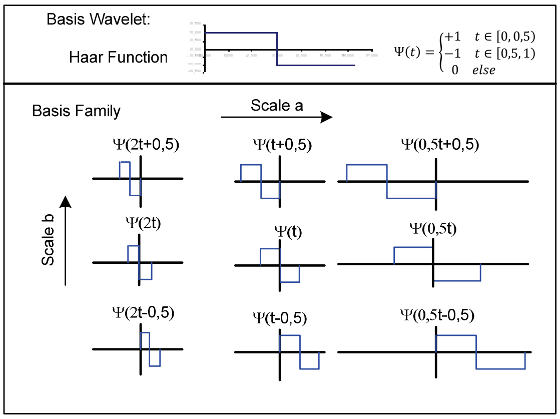


Figure 2: Wavelet building set based on Haar function.

Based on (3), the following equation shows the coefficient of the continuous Wavelet transformation $\mathrm{W}_{\Psi}\mathrm{f}^{*}(\mathrm{a,b})$ for the derived angle function $\mathrm{f}^{*}(\mathrm{t})$ given in (2).

$$\begin{aligned} &W_{\Psi}f^{*}(a,b) \\ &= |a|^{-1/2}\int_{-\infty}^{\infty} f^{*}(t)\,\Psi\left(\frac{t-b}{a}\right)dt \end{aligned} \tag{4}$$

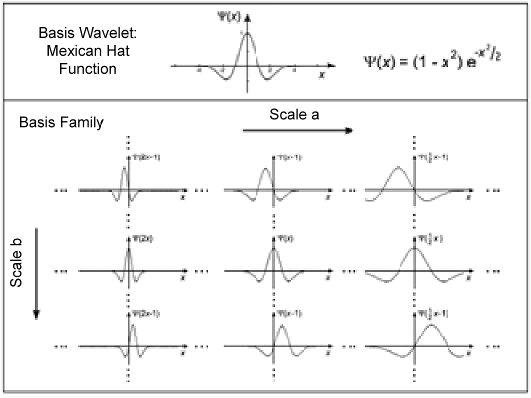


Figure 3: Wavelet building set based on Mexican Hat function.

Replacing the function Ψ in (4) by the scaled Haar function and setting the integration limits to $[0, 2\pi]$, we obtain the following expression:

$$\begin{aligned} W_{\mathrm{H}}f^{*}(a,b) = |a|^{-1/2}\Bigg[&\int_{b}^{b+a/2} f^{*}(t)\,dt \\ &- \int_{b+a/2}^{b+a} f^{*}(t)\,dt\Bigg] \end{aligned} \tag{5}$$

After executing the integrals in (5) we receive the following expression:

$$\begin{aligned} W_{\mathrm{H}}f^{*}(a,b) = |a|^{-1/2}\Bigg\{&b\left[f(l_j) - f(l_i)\right] \\ &- \sum_{m=i}^{j-1}\frac{2\pi}{L}l_m\alpha_m \\ &- (b+a)\left[f(l_k) - f(l_j)\right] \\ &+ \sum_{m=j}^{k-1}\frac{2\pi}{L}l_m\alpha_m + \frac{a^2}{4}\Bigg\} \end{aligned} \tag{6}$$

where i, j, k are the indices of the polygon edges according to the position of the current used Haar function within the interval $[0, 2\pi]$ and α_{m} are the calculated angel differences (Nabout, Tibken, 2007).

Simlar to (5) we receive (7) when we replace the function Ψ in (4) by the scaled Mexican Hat function.

$$\begin{aligned} W_M f^*(a,b) = a\sqrt{|a|}\left(e^{-z_n^2} - e^{-z_0^2}\right) \\ - \sqrt{2|a|}\, b\left(z_n e^{-z_n^2}\right. \\ \left. - z_0 e^{-z_0^2}\right) \\ + 2a\sqrt{|a|}\left(z_n^2 e^{-z_n^2}\right. \\ \left. - z_0^2 e^{-z_0^2}\right) \\ - \sqrt{2|a|}\sum_{m=1}^{n} z_m e^{-z_i^2}\alpha_m \\ - 2\sqrt{2|a|}\pi z_n e^{-z_n^2} \end{aligned} \qquad (7)$$

where

$$z_0 = \frac{1}{\sqrt{2}}\frac{b}{a},\ z_m = \frac{1}{\sqrt{2}}\left(\frac{2\pi l_m - bL}{aL}\right),\ z_n = \frac{1}{\sqrt{2}}\left(\frac{2\pi - b}{a}\right).$$

In (6) and (7) the terms which include the angle differences α_m are adequate to describe a given object shape.

We denote $W_H f^*(a,b)$ as Haar Wavelet descriptor (H-WD) and $W_M f(a,b)$ as Mexican Hat Wavelet descriptor (MH-WD).

4 DERIVATION OF WAVELET DESCRIPTORS

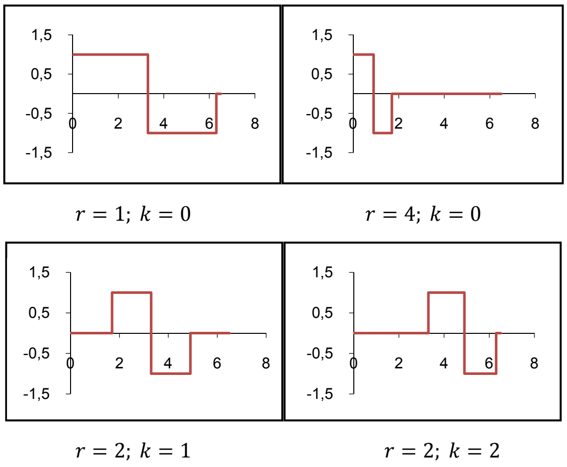
To obtain suitable WD for representing a given object shape we vary the values of the compression or dilatation parameter a and the shifting parameter *b* according to the following equations:

$$a = r\frac{2\pi}{m}; \qquad b = k\frac{2\pi}{m} \qquad (8)$$

with $m = \log_2(n)$ and n: number of WD

$r \in \{1, 2, \cdots, m\}$
$k \in \{0, 1, \cdots, m-1\}$

If we vary the parameter m as given in (8) we obtain a sufficient Wavelet building within the interval $[0, 2\pi]$. For $m \leq 6$ is $(a \geq 1)$ and (8) will deliver only components of the approximation signal. This signal describes the object shape roughly. Detailed signal information that describes small object shape changes can be derived for $m > 6$ or through additional use of the reciprocal values of a as given in (8). Generally only a few number of WD (e.g. 32) is needed in practical recognition applications to describe different object shapes. In this case the parameter m can be set to 4 if we use the reciprocal value of a to include components of the detail signal. For $m = 4$, Fig. 4 shows a part of the Haar wavelet building set for different parameter values.



$r = 1;\ k = 0$ $\qquad r = 4;\ k = 0$

$r = 2;\ k = 1$ $\qquad r = 2;\ k = 2$

Figure 4: Part of the Haar wavelet building set derived within the interval $[0, 2\pi]$.

Fig. 5 shows the corresponding Mexican Hat wavelet building set for the same parameters.

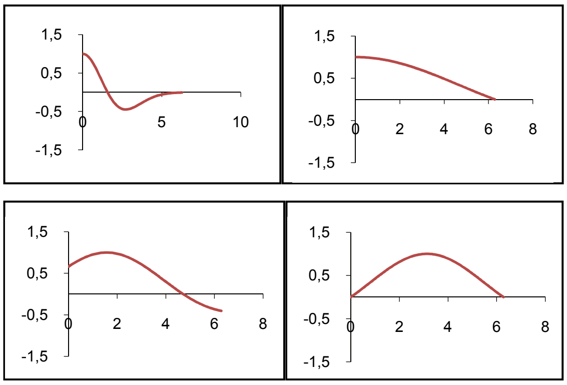


Figure 5: Part of the Maxican Hat wavelet building set derived within the interval $[0, 2\pi]$.

As shown in these figures small values of the parameter r produce compressed variants, big values on the other hand create dilated variants of the mother Wavelet. In both cases we receive an approximation signal of the Wavelet transformation, since $a > 1$. To receive components of the detail signal which describes small details of the contour shape we can use $1/a$ in combination with the same values of b. For such values we obtain WD which are qualified to describe small matches between the compared shapes.

5 RESULTS

Fig. 7 shows the 16 MH- as well as H-WD obtained from the approximation signal for the characters A and B as shown in Fig. 6. The used starting points of the derived angle functions are marked in Fig. 6 in green colour. The dilatation or compress parameter a and shifting parameter b are calculated as given in (11) for $r \in \{1,2,3,4\}$ and $k \in \{0,1,2,3\}$.

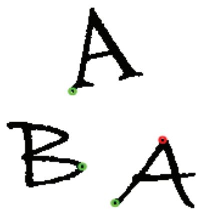


Figure 6: Example of an image with three characters.

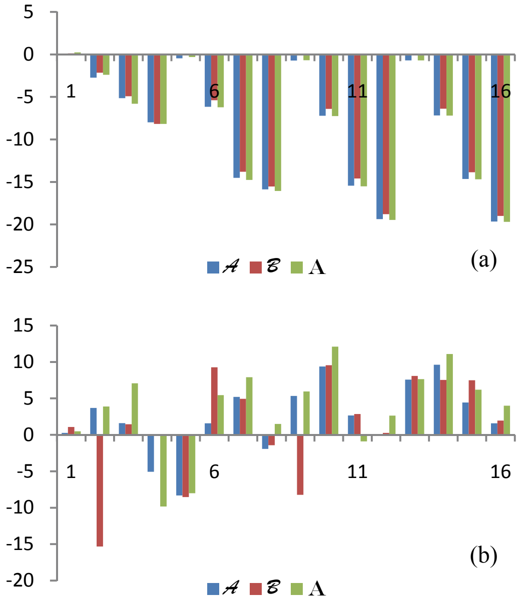


Figure 7: The first 16 MH (a) and H-WD (b) for the shapes of Fig. 6 obtained from the approximation signal.

As can be seen from Fig. 7 the differences between the MH-WD of the approximation signal are relatively small, the differences between the corresponding values of the H-WD are on the other hand large. This distinction is due to the discontinuity of the angle function, which causes big changes of the integration values in (6) when the Haar function jumps from negative to positive or positive to negative values. On the other hand the MH-WD indicate a periodical behaviour, so that only a few number of MH-WD of the approximation signal are needed to represent the given object shape. The H-WD on the other hand do not indicate such behaviour. This property carries forward even if we use different numbers of WD. Fig. 8 shows for instance the results of 25 WD of the approximation signal for $r \in \{1,2,3,4,5\}$ and $k \in \{0,1,2,3,4\}$.

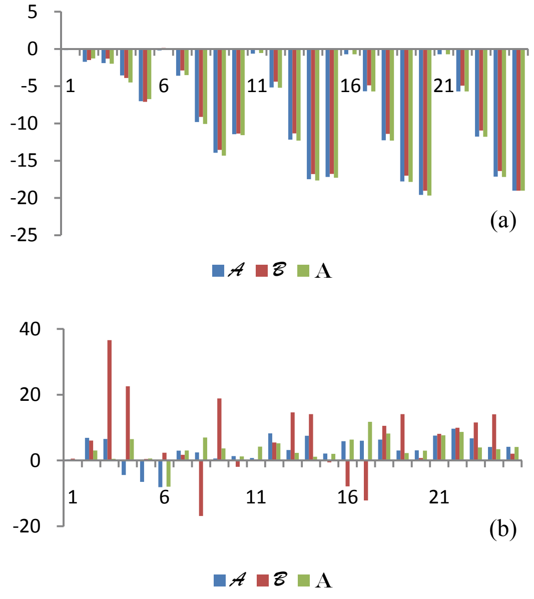


Figure 8: The first 25 MH and H-WD for the shapes of Fig. 6 derived from the approximation signal.

To study the detail signal we derived the 16 MH- and H-WD obtained from the detail signal by using the same parameter b and the reciprocal value of the parameter a given before (Fig. 9).

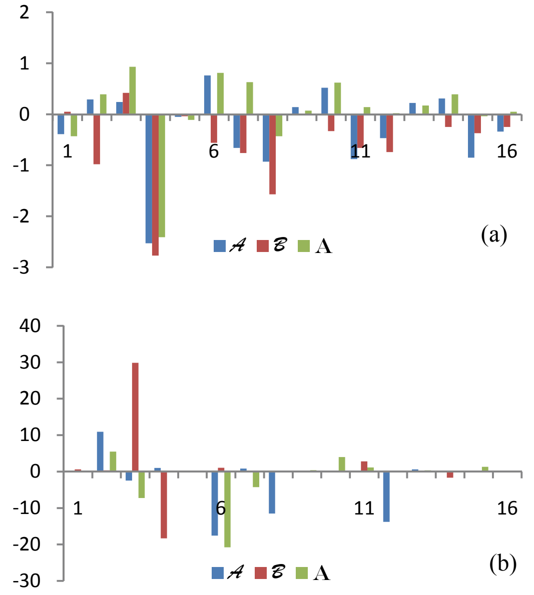


Figure 9: The first 16 MH and H-WD for the shapes of Fig. 6 derived from the detail signal.

The results in Fig. 9 indicate that the MH-WD of the detail signal do not show any periodical behaviour similar to the MH-WD of the approximation signal. On the other hand the values of the detail signal are relatively small. In contrast to the values of the MH-WD, the H-WD of the detail signal show relatively large differences.

To compare the WD between different object shapes we use the Euclidean distances d as given in the following equation:

$$d = \sqrt{\sum_{i=1}^{n} (WD_i - WD_i')^2} \tag{9}$$

where $\mathrm{WD_i}$ are the values of the Wavelet descriptors for the first object shape and $\mathrm{WD_i'}$ are the values for the second one.

The following table shows the distances between the object shapes given in our example derived from the approximation and detail signal separately.

Table 1: Euclidean Distances between the Object shapes of Fig. 6 obtained from MH-WD and H-WD.

Approximation signal (MH-WD)	**(d_A)**
„𝒜“ / „ℬ“	2,48
„𝒜“ / „A“	0,90
„A“ / „ℬ“	2,74

Detail signal (MH-WD)	**(d_D)**
„𝒜“ / „ℬ“	2,35
„𝒜“ / „A“	2,13
„A“ / „ℬ“	3,22

Approximation signal (H-WD)	**(d_A)**
„𝒜“ / „ℬ“	25,42
„𝒜“ / „A“	11,22
„A“ / „ℬ“	27,86

Detail signal (H-WD)	**(d_D)**
„𝒜“ / „ℬ“	47,05
„𝒜“ / „A“	20,81
„A“ / „ℬ“	47,51

As shown in table 1 the distances between the similar characters A and A are relatively small and for the different characters A and B as well as A und B on the other hand relatively large. Due to these results it is possible to recognize the different object shapes A und B using the minimum distance method. In our example the characters can be recognized using only the Euclidean distances of the approximation signal. In many other applications it is required to use also the detail signal to include more detail information about the local changes of the compared contour shapes.

The following example shows the results of applying the new method for the recognition of weed species (Fig. 10).

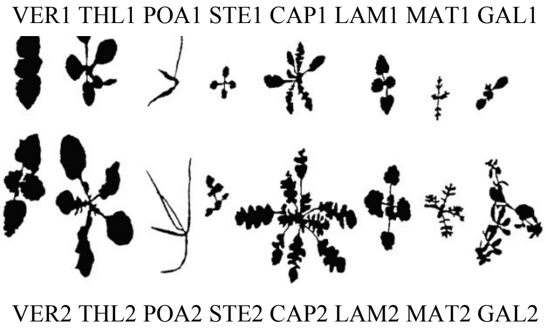

VER1 THL1 POA1 STE1 CAP1 LAM1 MAT1 GAL1
VER2 THL2 POA2 STE2 CAP2 LAM2 MAT2 GAL2

Figure 10: Eight weed species in different growth stages.

Table 2 shows the minimum distances between the WD of two different weeds using the WD components of the approximation and detail signal.

Table 2: Euclidean Distances between the weed LAM and MAT of Fig. 10 obtained from MH-WD and H-WD.

Approximation signal (MH-WD)	**(d_A)**
„LAM1“ / „LAM2“	0,55
„LAM1“ / „MAT1“	1,18
„LAM2“ / „MAT1“	1,07

Detail signal (MH-WD)	**(d_D)**
„LAM1“ / „LAM2“	1,77
„LAM1“ / „MAT1“	5,00
„LAM2“ / „MAT1“	5,52

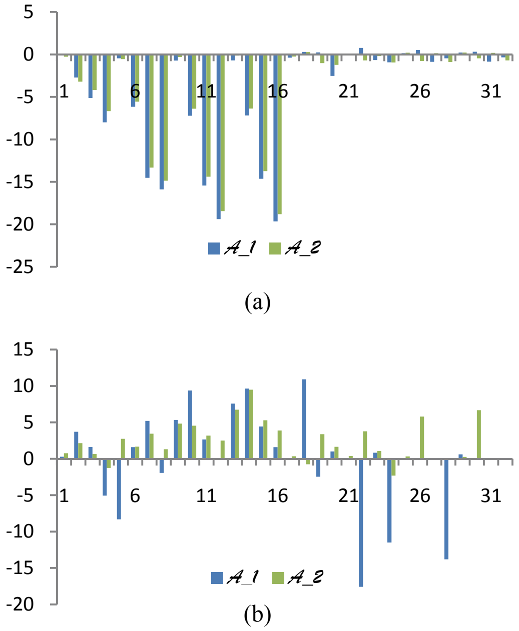
Approximation signal (H-WD)	**(d_A)**
„LAM1“ / „LAM2“	61,50
„LAM1“ / „MAT1“	94,32
„LAM2“ / „MAT1“	72,92

Detail signal (H-WD)	**(d_D)**
„LAM1“ / „LAM2“	73,50
„LAM1“ / „MAT1“	114,85
„LAM2“ / „MAT1“	128,08

As shown in table 2 the weed can be recognized correctly using the minimum distance method even when we use either the approximation or the detail signal alone. For some other weeds (VER and LAM) the recognition process has failed.

6 THE STARTING POINT PROBLEM

The results of table 1 are obtained according to the chosen starting points (green colored positions in Fig. 6). If the starting points change, the angle functions will also be changed and with them the corresponding WD. If we change the starting point of the character 𝒜 for instance from the green colored position to the red one we receive for this character the following MH-WD and H-WD values (Fig. 11). Here both approximation and detail signals are drawn in the same diagram.



(a)

(b)

Figure 11: The MH-WD (a) and H-WD (b) for the character 𝒜 for the two different staring points given in Fig. 6 derived from the approximation and detail signal.

As shown in Fig. 11 the change of the starting point leads to large changes of the WD. Since the position of the starting point in real applications depends on several parameters, which cannot be fixed, like position and rotation of the objects in the image, number of objects, extraction method etc., the recognition process using the minimum distance method will fail. Table 3 reports the Euclidean distances between the given characters A and B derived from the MH-WD for the new starting point of the character A (red position). Using the H-WD instead of the MH-WD indicates similar behavior.

Table 3: Euclidean Distances between the Objects of Fig. 6 derived from the MH-WD for the red starting point.

Approximation signal	(d_A)
„𝒜" / „ℬ"	4,43
„𝒜" / „A"	6,86
„A" / „ℬ"	2,74

Detail signal	(d_D)
„𝒜" / „ℬ"	2,73
„𝒜" / „A"	2,64
„A" / „ℬ"	3,22

As shown above the distances between the characters with the similar shapes A and A are higher than the distances between the different shapes A and B as well as A and B. This is due to the change of the angle functions within the interval $[0 - 2\pi]$ according to the change of the starting points. The following figure (Fig. 12) shows the angle functions of the character A for the two different starting points.

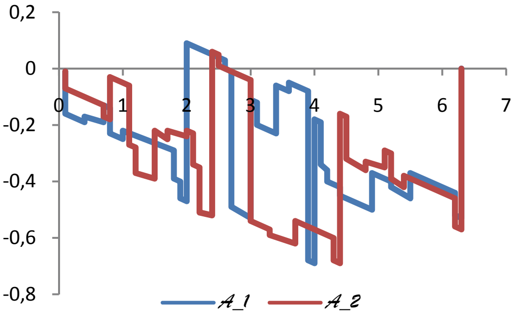


Figure 12: The angle functions of the characters 𝒜 for two different starting points.

As expected, the figure shows big differences of the angle functions related to the change of the starting point. Since the starting point on the contour depends on several parameters of the image, the above mentioned issue can cause confusion in recognition tasks, because it is not explicit clear whether big values of the Euclidean distance are related to shape differences or to different starting points. To solve this problem we use the following strategy:

Suppose we have a number of object samples Oj and an unknown object O which must be classified to one of the given object classes. The procedure can then be done as follows:

- Calculate the WD of all objects O_j for an arbitrary starting point and store them in a data base.
- Calculate the WD sets for all possible starting points of the unknown object O. This can be done easily if we use the polygon description of the object contour and change the starting point from one polygon corner to the next.
- Compare the WD sets of the unknown object separately with the stored WD of the object samples using the minimum distance method. We receive a number of Euclidean distances $d_{i,j};\ i = 1,2,\dots n;\ j = 1,2,\dots m$ according to the number of different starting points n used in step 2 and the number of object samples m given in step 1.
- Find the minimum value of d_{ij}. The stored object sample related to this minimum value j represents the recognized object.

7 CONCLUSIONS

The representation of object contours using wavelet descriptors is useful in object recognition tasks. In particular, the Mexican Hat as well as Haar function are qualified to be used as a mother wavelet to obtain a sufficient number of WD which can be used in recognition tasks. The WD can be calculated very easily using (6) for the H-WD and (7) for MH-WD. The number of WD needed to recognize a given object increases according to the complexity of the object shapes and must be set according to the given application. It is possible in some cases to use only the components of the approximation signal in order to recognize an unknown object using the minimum distance method, but generally the use of the detail signal will include detail information about small differences between the compared object shapes. The starting point on the contour has a big influence on the recognition process, since the values of the WD depend strongly on it. The paper describes one possible solution where not only one set of WD is computed and compared with the stored WD of the object samples, but several sets of WD according to the different starting points.

REFERENCES

Grenander, U., Chow, Y. and Keenan, D. M., 1991. HANDS: A Pattern Theoretic Study Of Biological Shapes. Springer Verlag.

Belongie, S., Malik, J. and Puzicha, J., 2001. Matching shapes, In ICCV, pp. I.454-461.

Fergus, R., Perona, P. and Zisserman, A., 2003. Object class recognition by unsupervised scale-invariant learning. In CVPR, pp. 264-271.

Nabout, A., Nour Eldin, H.A., Gerhards, R., Su, B., Kühbauch, W., 1994. Plant Species Identification using Fuzzy Set Theory, pp. 48-53, Proc. of the IEEE Southwest Symposium on Image Analysis and Interpretation, Dallas, Texas, USA.

Zahn, C., Roskies, R. Z., 1972. Fourier descriptors for plant closed curve, IEEE Trans. On C-21.

Nabout, A., 1993. Modular Concept and Method for Knowledge Based Recognition of Complex Objects in CAQ Applications, VDI Publisher, Series 20, No. 92.

Nabout, A., Tibken, B., 2004. Object Recognition Using Polygons and Wavelet Descriptors, 1st International Conference on Information & Communication Technologies, Proceedings of ICTTA'04, April 19 - 23, Damascus, Syria.

Nabout, A., Tibken, B., 2005. Wavelet Descriptors for Object Recognition Using Mexican Hat Function, 16th IFAC World Congress, Prague, Czech Republic.

Nabout, A., Tibken, B., 2007. Object Shape Recognition using Mexican Hat Wavelet Descriptors, 2007 IEEE International Conference on Control and Automation, Guangzhou, CHINA - May 30 to June 1, pp. 1313-1318.

Nabout, A., Tibken, B., 2008. Object Shape Description Using Haar-Wavelet Functions, the 3rd IEEE International Conference on Information & Communication Technologies: from Theory to Application, ICTTA'08, 7-11 April 2008, Umayyad Palace, Damascus, Syria.

Daubechies, I., 1992. Ten Lectures on Wavelets, Society for Industrial & Applied Mathematics.

A NOVEL HAPTIC INTERFACE FOR EXTENDED RANGE TELEPRESENCE

Control and Evaluation

Antonia Pérez Arias and Uwe D. Hanebeck
Intelligent Sensor-Actuator-Systems Laboratory (ISAS)
Institute for Anthropomatics
Universität Karlsruhe (TH), Germany
aperez@ira.uka.de, uwe.hanebeck@ieee.org

Keywords: Extended range telepresence, Motion compression, Haptic interface, Force control.

Abstract: A novel haptic interface for extended range telepresence is presented that allows the user simultaneous wide area motion and haptic interaction in remote environments. To achieve an extended workspace, the haptic interface consists of a haptic manipulator for precise haptic rendering and a large portal carrier system that enlarges the workspace by prepositioning the end-effector. As the prepositioning unit is grounded and driven by three linear drives, our approach has the advantages of high force capability and an accurate positioning of the haptic interface. The use of this haptic interface with Motion Compression permits to explore large remote environments even from small user environments. As a result, not only has the user visual, acoustic, and haptic feedback, but can also control the teleoperator or avatar by natural walking, which considerably increases the sense of immersion. A prototype system for haptic extended range telepresence was designed, implemented, and tested.

1 INTRODUCTION

Telepresence aims at creating the impression of being present in an environment, which is inaccessible to a human user. Such an environment can be real or virtual, and will be referred to in the following as *target environment*. The feeling of presence is achieved by visual and acoustic sensory information recorded from the target environment and presented to the user on an immersive display.

The more of the user's senses are involved, the better the immersion in the target environment. In order to use the sense of motion as well, which is especially important for human navigation and way finding, the user's motion can be tracked and transferred to the *teleoperator*, a mobile robot or an avatar, in the target environment. As a result, in extended range telepresence the user can additionally use the proprioception, the sense of motion, to navigate the teleoperator by natural walking, instead of using devices like joysticks, pedals or steering wheels.

However, without further processing the motion information, the motion of the user is restricted to the size of the *user environment*, which is limited, for example by the range of the tracking system or the available space. *Motion Compression* (Nitzsche et al., 2004) is an algorithmic approach that provides a nonlinear transformation, mapping the path in the target environment to a feasible path in the user environment by minimizing proprioceptive and visual inconsistencies.

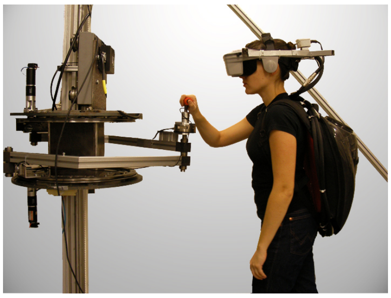


Figure 1: User and haptic interface for interaction with extended target environments.

Extended range telepresence can be applied in many fields, especially in those that require the human navigation skills to solve the task, for example tele-exploration of remote environments, visualization of complex structures, training of emergency evacuations, etc. An extended range telepresence system that uses Motion Compression to teleoperate a mobile robot is presented in (Rößler et al., 2005). Since haptic information is indispensable, amongst others, for haptic exploration and manipulation of objects in the target environment, a novel haptic interface for the extended range telepresence system was built. A picture of the system is shown in Fig. 1.

Force reflecting telepresence systems usually assume an immobile user and a restricted workspace. For example, industrial robots have often been used as haptic interfaces due to their accuracy and relative high force capability (Hoogen and Schmidt, 2001) but their limited workspace makes them unfeasible for extended range telepresence. In the last years, several haptic interfaces that allow a dexterous feedback and fairly high forces have been designed to enlarge their workspaces, e.g. a string-based haptic interface (Bouguila et al., 2000), or a grounded hyper-redundant haptic interface in (Ueberle et al., 2003). Portable haptic interfaces like exoskeletons (Bergamasco et al., 1994) solve the problem of wide area motion, since the interface is carried along by the user. However, working with exoskeletons can be fatiguing for the user due to the high weight of the system. In addition, they can display lower forces than grounded displays (Richard and Cutkosky, 1997). The only group of systems that really allow haptic interaction during wide area motion are mobile haptic interfaces (Nitzsche et al., 2003), (Formaglio et al., 2005), (Peer et al., 2007). These are usually small haptic devices mounted on a mobile platform. Drawbacks of such interfaces are a difficult control and a dependency of the force display quality on the localization of the mobile platform.

In this paper, we present a novel haptic device that allows haptic interaction in extended range telepresence and combines the advantages of grounded and mobile haptic interfaces. It consists of a grounded linear prepositioning unit that moves along with the user and a manipulator arm attached to the prepositioning unit that is used to display forces at any position in the user environment. This haptic interface allows in conjunction with Motion Compression unrestricted wide area motion and the possibility of effectively guiding the user in the target environment by means of haptic information. The control of the haptic interface is based on the decoupling of force control and prepositioning of the haptic device, which takes both the optimal manipulator's configuration and the user's position into consideration.

The remainder of this paper proceeds as follows. The following section presents the extended range telepresence system. Motion Compression is reviewed, since it determines the requirements of the haptic interface, and the mechanical setup of the haptic interface is presented. In section 3, a detailed description of the control design is given. Experimental results are shown in section 4. Finally, a summary and an outlook can be found in section 5.

2 EXTENDED RANGE TELEPRESENCE SYSTEM

2.1 Motion Compression

Motion Compression provides a nonlinear mapping between the user's path in the user environment and the path in the target environment. It consists of three modules: *Path prediction* tries to predict the desired path of the user in the target environment by means of tracking data and, if possible, information of the target environment. The resulting path is called *target path*. *Path transformation* transforms the target path so that it fits in the user environment. The resulting *user path* conserves the length and turning angles of the target path while there is a minimum difference in curvature. Fig. 2 shows these paths in both environments. Finally, the *user guidance* module guides the user on the user path, while he has the impression of walking on the original target path.

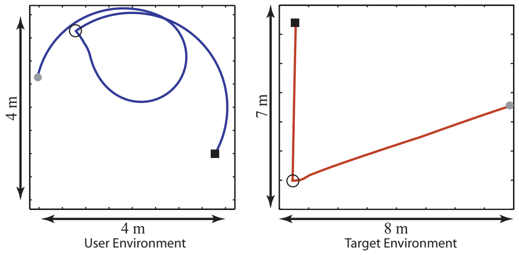


Figure 2: The corresponding paths in both environments. Left: user path in the user environment. Right: target path in the target environment.

The result of Motion Compression is a transformation between the user's position in the user environment and the teleoperator's position in the target environment at any time and position. This transformation can also be used to map the user's hand position, or to transform force vectors recorded by the teleoperator back into the user environment.

The use of Motion Compression for extended range telepresence puts a number of demands on the design of a haptic interface. The haptic interface must be able to reach all configurations in a user environment of $4 \times 4\,\mathrm{m}^2$, in which the user may move with a natural speed of up to $2\,\mathrm{m/s}$. Especially the rotational motion around the vertical axes must be unlimited.

2.2 A Novel Haptic Interface: Setup

Fine haptic rendering and wide area motion require very different characteristics regarding mechanics as well as control. Therefore a novel haptic interface was designed that consists of two subsystems: a linear prepositioning unit that accompanies the user along the user path so that he does not perceive the haptic display, and a manipulator arm attached to the prepositioning unit that is used to display defined forces at any position in the user environment. In this way, the workspace of the haptic interface covers the whole user environment. Fig.3 shows a CAD drawing of the complete setup.

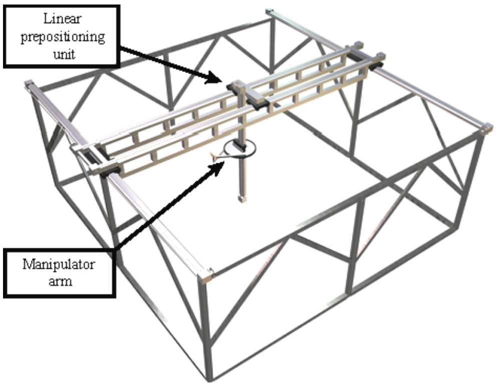


Figure 3: CAD drawing of the complete setup with linear positioning unit and manipulator arm.

The motion subsystem is realized as grounded portal carrier system of approximately $5 \times 5 \times 2\,\mathrm{m}^3$ with three translational degrees of freedom, which are realized by three independent linear drives. These linear drives are built using a commercially available carriage-on-rail system. The carriages are driven by a toothed belt. The x- and y-axis consist of two parallel rails each for stability reasons, while the z-axis is only a single rail. As a result, the system is driven by five synchronous AC-motors with helical-bevel servo gear units of 120 Nm maximal torque, that allow a maximum speed of $2\,\mathrm{m/\,s}$ and an acceleration of $2\,\mathrm{m/\,s}^2$. As the configuration space equals cartesian space, forward kinematics can be expressed by means of an identity matrix. Thus position control is extremely easy to handle and very robust (Rößler et al., 2006).

This construction has the advantages of a a high force capability and an accurate positioning of the manipulator, which is determined directly through encoder's information with relative accuracy 0.1 mm. By using a position control with high gains, the user does not perceive the motion subsystem, and the transparency depends only on the force-controlled subsystem (Nitzsche et al., 2003).

Because the acceleration of the human hand is typically much higher than the acceleration of the portal carrier, a fast manipulator was used. It covers the human arm workspace and has planar movement. It is implemented as a planar SCARA arm, which is attached to the z-axis of the portal carrier. The redundant planar degrees of freedom permit the separation of positioning and force display. Two active rotational joints driven by two 150 W DC-motors are integrated into the base, so that all moving joints are passive. Circular drives allow infinite motion around the z-axis. The manipulator arm was designed to display a force of 50 N at the end-effector. More details can be found in (Rößler et al., 2006).

3 CONTROL DESIGN OF THE HAPTIC INTERFACE

3.1 Kinematic Model

The control of this haptic interface is based on the decoupling of force control at the end-effector and position control of the haptic device. The position of the end-effector with respect to the basis coordinate frame $\underline{x}_E$ is described by the global position of the linear prepositioning unit, $\underline{x}_L$, and $\underline{x}_S$, the relative position of the manipulator arm with respect to the linear prepositioning unit as $\underline{x}_E = \underline{x}_L + \underline{x}_S$.

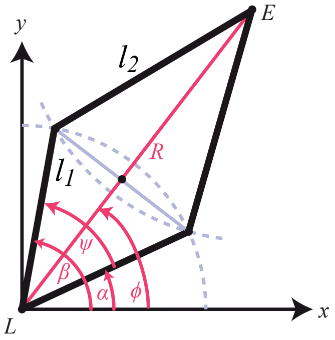


Figure 4: Geometrical SCARA-Model.

Fig. 4 shows the geometrical SCARA-model used to derive the kinematic equations. L represents the linear prepositioning unit, E the end-effector, and l_1 and l_2 the lengths of the inner and outer segments, respectively. If only the joints at the angles α and β are actively driven, the end-effector position $\underline{x}_E$ can be expressed as

$$\underline{x}_E = \underline{x}_L + \underline{x}_S = \begin{bmatrix} x_L \\ y_L \end{bmatrix} + \begin{bmatrix} \cos\left(\frac{\alpha+\beta}{2}\right) \cdot R \\ \sin\left(\frac{\alpha+\beta}{2}\right) \cdot R \end{bmatrix}, \quad (1)$$

where R, the radial travel, is calculated as

$$R = l_1 \cos\left(\frac{\beta-\alpha}{2}\right) + \sqrt{l_2^2 - l_1^2 \sin^2\left(\frac{\beta-\alpha}{2}\right)} \quad (2)$$

With this equation the direct kinematics of the manipulator is defined. The Jacobian of the manipulator $J(\underline{\gamma})$ on the configuration space $\underline{\gamma} = \begin{bmatrix}\alpha & \beta\end{bmatrix}^{\mathrm{T}}$, which will be used next for the control of the haptic interface, is defined as

$$J\left(\underline{\gamma}\right) = \frac{\partial \underline{x}_S\left(\underline{\gamma}\right)}{\partial \underline{\gamma}} \quad (3)$$

3.2 Control Structure

Fig. 5 illustrates the block diagram of the control scheme with force feedback, as well as the position control of the haptic display. The end-effector velocity of the haptic interface $\dot{\underline{x}}_E$ is transmitted via the communication channel and acts as reference velocity at the end-effector of the teleoperator $\dot{\underline{x}}_{E,ref,T}$. The environment reacts according to its impedance with a reaction $\underline{F}_T$, which is measured by the teleoperator, and transmitted to the haptic interface as reference input $\underline{F}_{T,ref,U}$. This architecture represents a two-channel force-velocity bilateral control.

In our system, the haptic interface is modelled as an admittance, which transforms $\underline{F}_{U,ref}$, the reference force to be displayed, into the reference motion of the end-effector as

$$\underline{F}_{U,ref} = M \cdot \ddot{\underline{x}}_{E,ref} + D \cdot \dot{\underline{x}}_{E,ref}, \quad (4)$$

where M is the desired mass matrix and D the desired damping matrix. The admittance control scheme is very well suited for systems with nonlinearities and large dynamic properties because the admittance model shapes the desired dynamic behaviour of the system by compensating the natural device dynamics (Ueberle and Buss, 2004).

The resolved acceleration control (J^{-1} *control*) is applied to calculate the commanded motor torque of the manipulator $\underline{\tau}_{ref}$:

$$\underline{\tau}_{ref} = \hat{M} \cdot J^{-1} \cdot \ddot{\underline{x}}_C + \underline{h}\left(\underline{\gamma}, \dot{\underline{\gamma}}\right) \cdot \dot{\underline{\gamma}} + \underline{g}\left(\underline{\gamma}\right), \quad (5)$$

where $\hat{M}$ is an approximation of the device joint inertia matrix and J is the Jacobian. The vectors $\underline{h}$, representing the friction effects, and $\underline{g}$ can be approximated through experimental identification.

The user, while moving the end-effector, applies a force consisting of a voluntarily applied force and a reaction force induced by the arm impedance. In order to reconstruct the applied force from the measured force $\underline{F}_U^*$ and the velocity of the end-effector, a model of the human arm impedance is applied:

$$\underline{F}_U = \underline{F}_U^* + M_u \cdot \ddot{\underline{x}}_E + D_u \cdot \dot{\underline{x}}_E + K_u \cdot \underline{x}_E \quad (6)$$

It is known that the arm impedance varies with the user and the arm configuration. Hence, the mean values of multiple users and planar configurations were used: $M_u = 2\,\mathrm{Kg}$, $D_u = 6\,\mathrm{Ns/m}$, and $K_u = 10\,\mathrm{N/\,m}$.

The reference position of the linear prepositioning unit $\underline{x}_{L,ref}$, which can be easily controlled in cartesian coordinates, is calculated by optimizing the manipulator's configuration according to some performance measure.

3.3 Prepositioning

When attaching the SCARA manipulator to the portal carrier, there is a redundancy in the planar directions that may be resolved by optimizing the manipulability of the SCARA. The manipulability is usually represented as

$$w\left(\underline{\gamma}\right) = \sqrt{det\left(J^T(\underline{\gamma}) \cdot J(\underline{\gamma})\right)}. \quad (7)$$

For $l_1 = 0.285\,\mathrm{m}$ and $l_2 = 0.708\,\mathrm{m}$, the SCARA robot's manipulability was found to be optimal when $\psi = \beta - \alpha = 2.048$.

Let's consider the polar coordinates of the end-effector's position: $R = l_1 \cos\left(\frac{\psi}{2}\right) + \sqrt{l_2^2 - l_1^2 \sin^2\left(\frac{\psi}{2}\right)}$, and $\phi = \frac{\alpha+\beta}{2}$. Since the manipulability w is independent of ϕ, another criterion must be found to optimize this parameter. It is also crucial to avoid collisions with the user, therefore the angle ϕ is chosen that maximizes the distance d between the user and the prepositioning unit by adopting $R^{opt} = R\left(\psi^{opt}\right)$. By designating $\underline{x}_{EH}$ the vector from the end-effector's position to the user's position, and θ the angle between this vector and the x-axis, the distance d can be expressed as

$$d = R^{opt\,2} + |\underline{x}_{EH}|^2 - 2R|\underline{x}_{EH}|\cos(\theta - \phi + \pi), \quad (8)$$

and it is maximal when $\phi^{opt} = \theta$, or in other words, when the linear prepositioning unit is situated in front of the user, and lies on the connecting line between the user's head and the end-effector. The optimal joint angles are finally $\alpha^{opt} = \phi^{opt} - \frac{\psi^{opt}}{2}$ and $\beta^{opt} = \phi^{opt} + \frac{\psi^{opt}}{2}$.

With $\underline{x}_S^{opt} = \begin{bmatrix}\cos(\phi^{opt}) \cdot R^{opt} & \sin(\phi^{opt}) \cdot R^{opt}\end{bmatrix}^{\mathrm{T}}$ being the optimal configuration of the manipulator, the reference position of the linear prepositioning results $\underline{x}_{L,ref} = \underline{x}_E - \underline{x}_S^{opt}$.

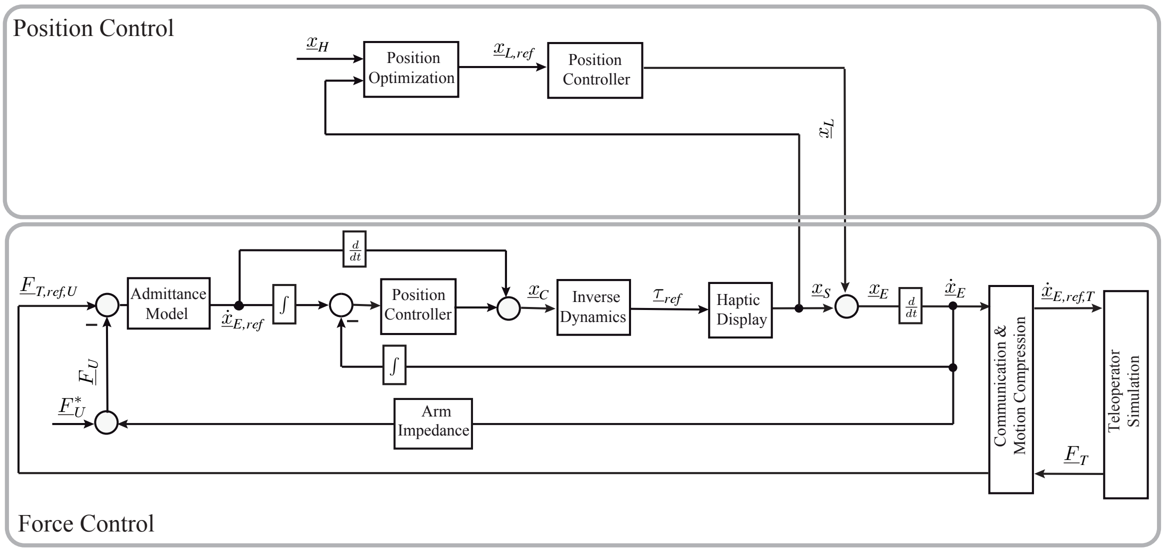


Figure 5: Control scheme of the haptic interface.

4 EXPERIMENTS

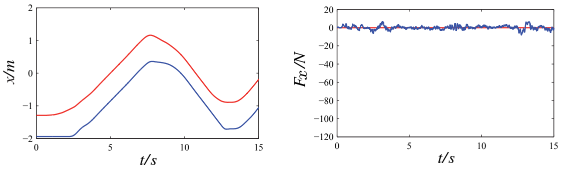
Two kinds of experiments were performed in order to evaluate the proposed haptic interface. First, the proposed force control was tested and second, the simultaneous wide area motion with haptic interaction was validated.

The force at the end-effector, and the positions of both, end-effector and prepositioning unit, were recorded during free motion and during a hard contact. In order to achieve transparency, the reference force during free motion is $\underline{F}_{U,ref} = 0\,\mathrm{N}$. An admittance of $M = 4\,\mathrm{kg}$ was simulated. The control gains of the prepositioning and the admittance position controller were obtained experimentally using standard Ziegler-Nichols.

Fig. 6 shows the force-position plots for the x-direction, when a user walks 15 seconds back and forth about 2 m in x-direction. Analogously, Fig. 7 represents the reference and the measured force when a user walks against a wall at position $-0.5\,\mathrm{m}$ with rigidity $K = 700\,\mathrm{N/m}$. The maximal displayed force is limited to 60 N. Both figures also show the motion of the linear positioning unit at an optimal distance of the end-effector.

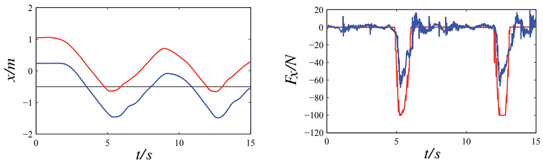
The main advantage of the admittance control is that the desired mass and damping of the device can be shaped. However, it is known that the admittance control reduces the force bandwidth of the haptic system.

The prepositioning was tested together with the haptic interaction to validate the entire concept of the haptic interface. For this purpose, the virtual and the user environment were supposed coincident, i.e. $4 \times 4\,\mathrm{m}^2$ large, and two virtual walls were placed inside. Fig. 8 shows the results of this experiment. The motion of the user can be divided into four segments: a) the user moves toward the wall, b) the user walks along the wall 1, c) the user walks along the wall 2, and d) the user turns on place. The haptic interface is always on the opposite side of the end-effector, so that the danger of a collision with the user is avoided. At the same time, the distance between the end-effector and the basis of the haptic interface is kept constant



(a) (b)

Figure 6: Position and force during free motion. (a) End-effector position x_E (red), linear system position x_L (blue). (b) Reference force $F_{x,ref} = 0\,\mathrm{N}$ (red), actual force F_x (blue).



(a) (b)

Figure 7: Position and force by hard contact. (a) End-effector position x_E (red), linear system position x_L (blue). (b) Reference force $F_{x,ref}$ (red), actual force F_x (blue).

on the optimal value that maximizes the manipulability of the haptic display.

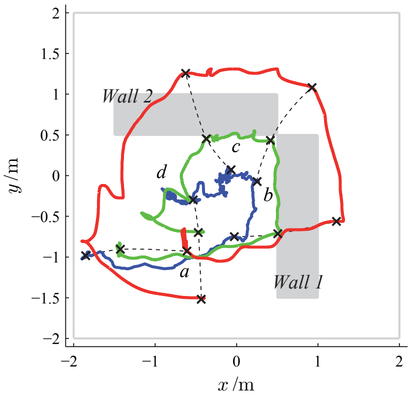


Figure 8: Positions of user x_H (blue), end-effector x_E (green) and linear system x_L (red) in presence of virtual walls during wide area motion.

5 CONCLUSIONS

This paper presents a novel multi-modal telepresence system for extended range telepresence, which enables the user to explore remote environments by natural walking. To achieve this goal, a novel haptic interface was developed, which consists of a haptic manipulator mounted on a linear prepositioning unit that follows the user by keeping the optimal configuration of the manipulator and avoiding collisions with the user. For the haptic feedback, a dedicated force control was implemented and tested. It uses an admittance model to shape the dynamics of the system, as well as a model of the impedances of arm and manipulator to compensate their undesired dynamics. Experiments show the suitability of this haptic interface for extended range telepresence.

The use of haptic information in extended range telepresence to improve the user guidance is a promising application of the presented haptic interface, which is currently being investigated. For this application, the simultaneous compression of head and hand motion represents a further challenge.

REFERENCES

Bergamasco, M., Allotta, B., Bosio, L., Ferretti, L., Perrini, G., Prisco, G. M., Salsedo, F., and Sartini, G. (1994). ”An arm exoskeleton system for teleoperation and virtual environment applications”. In *Proceedings of the IEEE Intl. Conference on Robotics and Automation*.

Bouguila, L., Ishii, M., and Sato, M. (2000). ”Multi-modal haptic device for large-scale virtual environment”. In *Proceedings of the 8th ACM Intl. Conference on Multimedia*.

Formaglio, A., Giannitrapani, A., Barbagli, F., Franzini, M., and Prattichizzo, D. (2005). ”Performance of mobile haptic interfaces”. In *Proceedings of the 44th IEEE Conference on Decision and Control and the European Control Conference*.

Hoogen, J. and Schmidt, G. (2001). ”Experimental results in control of an industrial robot used as a haptic interface”. In *Proceedings of the IFAC Telematics Applications in Automation and Robotics*.

Nitzsche, N., Hanebeck, U. D., and Schmidt, G. (2003). ”Design issues of mobile haptic interfaces”. *Journal of Robotic Systems*, 20(9):549–556.

Nitzsche, N., Hanebeck, U. D., and Schmidt, G. (2004). ”Motion compression for telepresent walking in large target environments”. *Presence*, 13(1):44–60.

Peer, A., Komoguchi, Y., and Buss, M. (2007). ” Towards a mobile haptic interface for bimanual manipulations”. In *Proceedings of the IEEE/RSJ Intl. Conference on Intelligent Robots and Systems*.

Richard, C. and Cutkosky, M. R. (1997). ”Contact force perception with an ungrounded haptic interface”. In *Proceedings of ASME IMECE 6th Annual Symposium on Haptic Interfaces*.

Rößler, P., Armstrong, T., Hessel, O., Mende, M., and Hanebeck, U. D. (2006). ”A novel haptic interface for free locomotion in extended range telepresence scenarios”. In *Proceedings of the 3rd Intl. Conference on Informatics in Control, Automation and Robotics*.

Rößler, P., Beutler, F., Hanebeck, U. D., and Nitzsche, N. (2005). ”Motion compression applied to guidance of a mobile teleoperator”. In *Proceedings of the IEEE Intl. Conference on Intelligent Robots and Systems*.

Ueberle, M. and Buss, M. (2004). ”Control of kinesthetic haptic interfaces”. In *Proceedings of the IEEE/RSJ Intl. Conference on Intelligent Robots and Systems, Workshop on Touch and Haptics*.

Ueberle, M., Mock, N., and Buss, M. (2003). ”Towards a hyper-redundant haptic display”. In *Proceedings of the International Workshop on High-Fidelity Telepresence and Teleaction*.

POSITION/FORCE CONTROL OF A 1-DOF SET-UP POWERED BY PNEUMATIC MUSCLES

Aron Pujana-Arrese, Kepa Bastegieta, Anjel Mendizabal, Ramon Prestamero and Joseba Landaluze
IKERLAN Research Centre, Arizmendiarrieta 2, E-20500 Arrasate (The Basque Country), Spain
{Apujana, Kbastegieta, Amendizabal, Rprestamero, JLandaluze}@ikerlan.es

Keywords: Pneumatic muscle, Robotic Arm, Position/Force Control, Hybrid Control, Impedance control.

Abstract: A one-degree-of-freedom set-up driven by pneumatic muscles was designed and built in order to research the applicability of pneumatic artificial muscles in industrial applications, especially in wearable robots such as exoskeletons. The experimental set-up is very non-linear and very difficult to control properly. This paper describes the control of this mechatronic system's interaction with its environment, controlling both its position and the force exerted against it. The classic position/force control techniques - hybrid control and impedance control - have been adapted to pneumatic muscles and applied to the experimental set-up developed. An alternative solution is also proposed whereby force or torque control is based on the calculation made by an estimator instead of on direct measurement by a sensor. The article presents a detailed analysis of the force and torque estimator used to close the control loops in the two position/force control schemes. Finally, the article concludes by presenting the experimental results obtained and the most outstanding conclusions of the study as a whole.

1 INTRODUCTION

The group of researchers from the IKERLAN technology centre working on the development of mechatronic systems have been involved for the last three years in the design and construction of an upper limb IAD (Intelligent Assist Device) (Martinez, 2007; Martinez, 2008). More specifically, the device is an exoskeleton for helping the user carry out routine tasks in the workplace (Figure 1). One of the requirements established from the start was to include non-conventional actuators as far as possible. Among the alternatives studied, artificial pneumatic muscles were considered to be the most suitable forms of actuation. In order to study the applicability of this type of actuators in biomechatronic systems a 1-DoF experimental set-up was built, driven by a pair of antagonistic pneumatic muscles. Initially, a dynamic model of the pneumatic muscle was created, and then used to make the full model of the experimental set-up. This model was experimentally validated (Pujana-Arrese, 2007).

Motivated by the high degree of non-linearity of the experimental prototype, the authors developed different solutions for robust control of the system's angular position: from a first initial attempt using basic controllers, to more advanced techniques achieved such as H_∞ or the non-linear sliding mode technique (Pujana-Arrese, 2008; Arenas, 2008).

This paper takes a step further as regards control of the mechatronic system's interaction with its environment, controlling not only the position but also the force exerted against it. An alternative

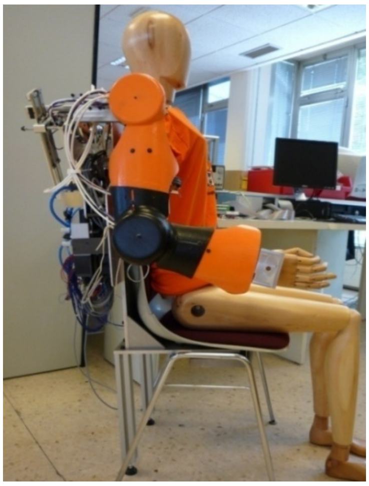


Figure 1: Exoskeleton ÎKO (IKerlan's Orthosis) worn by a dummy.

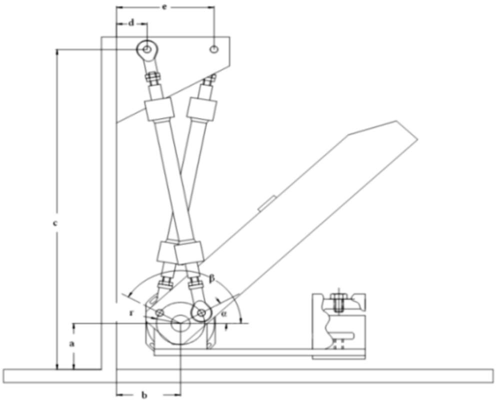


Figure 2: Geometric model of the 1-DoF robotic arm.

solution is also put forward whereby the force or torque control is based on the calculation made by an estimator instead of on direct measurement by a sensor.

A classic, in the field of the position/force control, is the Hybrid Controller strategy put forward by Raibert and Craig (1981). The controller carries out its action using selection matrixes which establish some spatial directions where position control must be carried out and some others where force must be controlled. In this way the force and position control actions are uncoupled by using the appropriate treatment of the spatial geometry where the manipulation task is being carried out. Another classic strategy is Impedance Control (Hogan, 1985), which does not control the position or the force but the dynamic relation between the two. This type of control strategy is deemed to be very suitable for IADs, although it needs to be adapted depending on the specific application.

The object of this paper is to present the algorithms implemented for controlling interaction with the environment, stressing the fact that an estimator requiring no direct measurement of either the torque or the force exerted by the mechatronic device has been developed for this purpose. The first point contains a brief description of the experimental set-up used for this study, and the paper then goes on to present a theoretical review of the control techniques most commonly used for these ends: Hybrid Control and Impedance Control. There then follows an in-depth analysis of the force and torque estimator used to close the control loops in both cases, and there is then a detailed description of the control diagram used for both the Hybrid Control and Impedance Control. Finally, the article concludes by presenting the experimental results obtained, and the most outstanding conclusions of the study as a whole.

2 SET-UP DESCRIPTION

A human arm orthosis-type application has been taken into consideration when designing the set-up. To this end and albeit with a single degree of freedom, it was considered that it should allow for the greatest angular displacement possible, and that it should be able to transport the greatest mass possible at the tip (emulating a weight borne by the hand). On the other hand, however, it needed to be confined to the length of the pneumatic muscles. In seeking a compromise between all the specifications, a displacement of around 60º and a maximum mass to be moved at the tip of 8 kg were set. By trying to minimize the length of the muscle required, the design focused on the mechanism that would enable the arm and inertias to rotate with good dynamics by means of the two muscles.

The pneumatic muscle that was chosen was the DMSP-20-200N manufactured by Festo. Figure 2 and Figure 3 show the resultant mechanism and a picture of the prototype. The parameter values that define the mechanism are:

a=5 mm; b=85 mm; c=491 mm; d=40.6 mm
e=129.4 mm; α=0º-60º; β=120º-180º; r=32 mm

From these values the distance L (mm) between the ends (joining points of the mechanism) of the pneumatic muscles is:

$$L = \sqrt{175059 + 2841.6 \cdot \cos\alpha - 26624 \cdot \sin\beta}$$

When the muscles are without pressure, the distance L is of 423 mm, with the length of the muscle fibre being 200 mm. The centre of the arm mass with regard to the centre of rotation is at a height of 17.6 mm and at a horizontal distance of 205 mm, considering that the arm is in the horizontal position. The arm mass is 0.987 kg. The centre of the additional masses placed on the end of the arm would be at a height of –24 mm and at a horizontal length of 367 mm with regard to the centre of rotation, always bearing in mind that the arm is in the horizontal position. The set-up may be rotated so that the arm moves in a horizontal plane and the effects of gravity are therefore cancelled out. The prototype (Figure 3) includes a *FAGOR S-D90* encoder, which supplies 180,000 pulses per turn, and a load cell on the lower stop of the model.

The schematic diagram of the set-up, which includes the control hardware, sensors and pneumatic circuit, is shown in Figure 4. As the figure shows, two *Festo MPYE-5-1/8HF* pneumatic servo-valve are used for actuation, each linked to

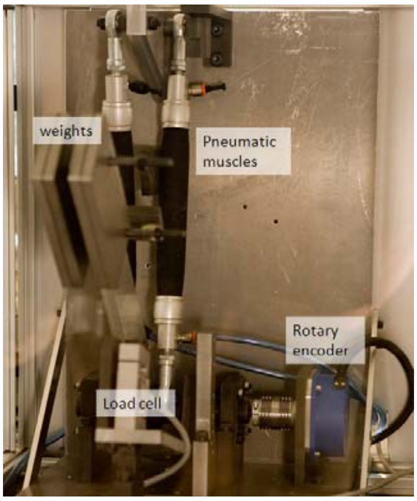


Figure 3: Picture of the experimental set-up.

one pneumatic muscle and controlled independently by the controller. The controller hardware is *PIP8*, an industrial PC made by the company MPL, which is very similar to The MathWorks' *xPCTargetBox*. A PC104 card (*Sensoray model 526*) was incorporated into the *PIP8* in order to read and write all the system signals. Control algorithms were implemented in *Simulink* and code was generated and downloaded in the aforementioned hardware by means of two of The MathWorks' tools: *RTW* and *xPCTarget*.

3 POSITION/FORCE CONTROL ALGORITHMS OVERVIEW

Research into pneumatic muscles has been carried out considering them as orthosis actuators. And an orthosis, or exoskeleton, is a wearable robotic device. In an initial approach, the basic control of an orthosis-type device can be considered to be based on position control, where the user creates the movement set-point and closes the loop aided by the human body's own sensors. Detecting the user's *intention* and creating the movement set-point on the basis of this is a key element. Another very important factor to be taken into account is the interaction with the environment, whether from the perspective of controlling the force exerted so as not to cause damage to persons within the robotic device's field of action, or because the device is being used as a force augmentation system. From this point of view, its functioning is similar to that of a robotic manipulator. There are two classic position/force schemes for robotic manipulators: hybrid control and impedance (or admittance) control. These schemes have been considered valid for the case of an arm orthosis, although they have some special characteristics that must be taken into account when the actuators are pneumatic muscles.

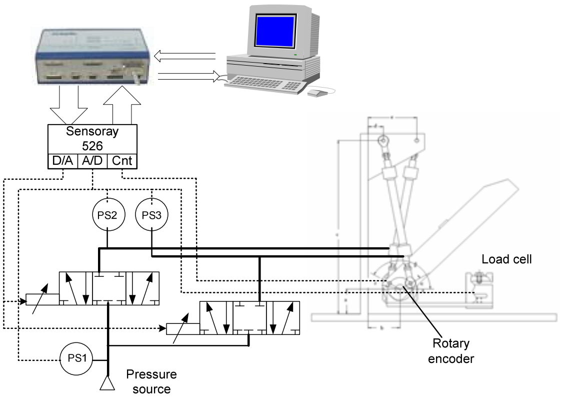


Figure 4: Schematic diagram of the set-up and pneumatic circuit.

3.1 Hybrid Position/Force Control

Hybrid control is a conceptually simple method for controlling both the position and the contact force generated at the tip of a manipulator during a task involving restricted movement. The method does not specify any feedback control law for regulating the errors, but rather a control architecture in which any position and force control techniques can be considered. The principle of hybrid control is based on the idea that each manipulation task can be described by specifying a set of contact surfaces. These surfaces serve to detail the restrictions existing in the system, which may be either *natural* or *artificial*.

The *natural restrictions* are connected with the system's particular mechanical and geometrical characteristics. *Artificial restrictions,* on the other hand, are connected with the control task objectives, and are specified in terms of position or force parameters.

Natural and artificial restrictions are defined within the space of the task, not within the space of the actuations. One natural and one artificial restriction may be specified for each degree of freedom of the task. In general, taking the task geometry into account, it is not difficult to determine the natural restrictions existing and decide on the most suitable way of dividing the space of the task on the basis of these.

For the case of an orthosis, normally *mixed* exercises are performed, which consist of both

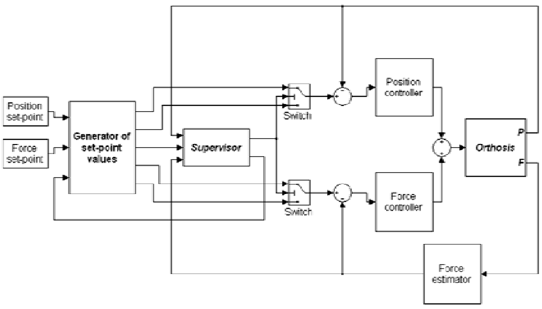


Figure 5: Structure of the hybrid position/force controller.

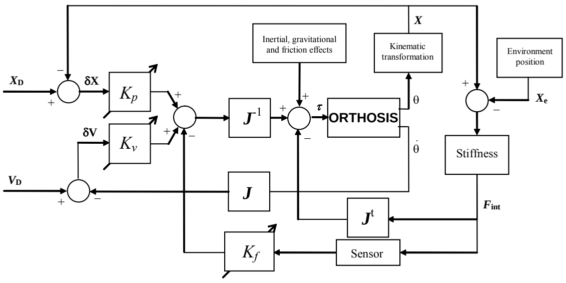


Figure 6: Structure of the impedance controller.

stages of free movement and stages of restricted movement. The restricted movement part can be of much less importance than in the case of a robotic manipulator, any may simply consist of a force control loop, which does not necessarily have to act at the same time as the position control in other directions. In any case, a *supervisor*, able to switch between the different configurations of the manipulator and the corresponding control laws, is a key element. A supervisor of this type must pay special attention to the impact between the manipulator and the environment, and to its separation.

Figure 5 shows a diagram of position/force hybrid control valid for both an orthosis and the 1-DOF set-up used to analyse the specific case of the pneumatic muscles. It basically consists of two independent controls, one for position and the other for force, and a supervisor that switches between one control type and another depending on the contact with the environment. The supervision is based on the information on the force exerted, which may be provided by a force sensor or an estimator. The supervisor, at the same time as the set-point generator, makes the transition between the controllers in a *soft* manner, to prevent rebounds and to assure the system's stability.

As already mentioned, the hybrid control scheme does not impose the control techniques that are used for the position controller and for the force controller.

3.2 Impedance Control

Impedance control is another classic force control scheme, and it is of great interest in the case of orthoses. It does not require a supervisor and it is able to take on the control of a composite task, with free and restricted movement stages, maintaining the system's stability without changing the control algorithm. It is based on the idea of controlling the dynamic relationship between the force and position variables of the physical systems. It is presumed that in any manipulation task the environment contains inertias and kinematical restrictions, i.e. systems accepting forces as input and responding by means of displacements (admittances). The manipulator in contact with the environment must accordingly behave as an impedance and respond with a determined force to the displacement of the environment. The general strategy may be established in terms of controlling a movement variable and at the same time providing the manipulator with a disturbance response in the form of impedance. Thus, the interaction between the manipulator and its environment can be modulated and controlled by acting on the impedance values.

In impedance control, the functional form of the torque of a manipulator's actuators is well-known:

$$\begin{aligned}\tau_{\text{act}} = {} & I(\theta)J^{-1}(\theta)M^{-1}K\left[X_0 - L(\theta)\right] + S(\theta) \\ & + I(\theta)\mathrm{J}^{-1}(\theta)M^{-1}B\left[V_0 - J^{-1}(\theta)\omega\right] + V(\omega) \\ & + I(\theta)J^{-1}(\theta)M^{-1}F_{\text{int}} - J^{\text{t}}(\theta)F_{\text{int}} \\ & + I(\theta)J^{-1}(\theta)G(\theta,\omega) + C(\theta,\omega)\end{aligned} \tag{1}$$

where each line of the second member represents a contribution of a different nature to the total torque. The first line corresponds to terms dependent on the position, the second to terms of speed, the third to terms of force, and the fourth to terms of inertial coupling. Within the field of the actuations, this equation expresses the behaviour that the controller should be able to induce in the manipulator, in the form of a non-linear impedance. The input variables are the desired Cartesian positions and speeds, and the terms – linear or not – that specify the required dynamic behaviour, characterised by the magnitudes *M*, *B*, and *K*. Figure 6 shows the typical impedance control structure for a robotic manipulator or for an orthosis, in which the feedback gains of the position, speed and force loops, K_p, K_v and K_f respectively, depend on the reference inertia and mass tensors and on the values designed for stiffness, *K*, and damping, *B*, and they are deduced from the control law (1).

The force feedback F_{int}, based on a measured force or estimated force, has the effect of changing the apparent inertia of the manipulator. However, the impedance control scheme can also be applied without a sensor or force estimator. In this case the force is not explicitly controlled, but, depending on the impedance values used in the controller design, the force the system exerts on the environment is limited.

4 TORQUE/FORCE ESTIMATOR

As already mentioned in the introduction, on most occasions different sensors are used to directly measure the force or torque exerted by the actuators in order to control the interaction between the mechatronic system and its environment (Tsagarakis, 2007; Jia-Fan, 2008).

This paper puts forward the idea of carrying out the control of the interaction between a mechatronic system driven by pneumatic muscles and its environment without any direct measurement at all of either the torque exerted by the pneumatic actuators or the force exerted by the arm. The torque and force are calculated on an estimated basis from the angular position of the arm and the pressures on the pair of muscles, as set out in the ensuing paragraphs.

The force exerted by each muscle can be modelled on the basis of its contraction and interior pressure according to equation (2) (Pujana-Arrese, 2007).

$$F_{up/down} = \left(D_1 + D_2 \cdot q_{up/down} + D_3 \cdot {q_{up/down}}^2\right) P_{up/down} + \varphi\left(q_{up/down}\right) \tag{2}$$

where

$$\varphi(q) = a + b \cdot q + c \cdot q^2 + d \cdot q^3 + e \cdot q^4 \tag{3}$$

q is the contraction of each muscle (*up* and *down*), P is the pressure exerted on each muscle, and the parameters D_1, D_2, D_3, a, b, c, d and e are constants obtained from empirical tests for characterising the behaviour of the muscles (Pujana-Arrese, 2007).

To calculate muscle contraction, trigonometric formulas are used to relate this contraction with the angle formed between the arm and the vertical (Figure 2). The torque exerted by the combination of the two pneumatic muscles (τ_{pres}) can thus be deduced as:

$$\tau_{pres} = F_{up} \cdot r \cdot \sin\left[\left({}^{\pi}/_{2} - \left(\theta_{top} - \theta\right)\right) + \left({}^{\pi}/_{2} - \alpha_{up}\right)\right] - F_{down} \cdot r \cdot \sin\left[\left({}^{\pi}/_{2} - \left(\theta - \theta_{down}\right)\right) + \left({}^{\pi}/_{2} - \alpha_{down}\right)\right] \tag{4}$$

where F_{up} is the force exerted by one pneumatic muscle, F_{down} is the force exerted by the other pneumatic muscle, and r is the distance between the rotation point and the lower joining points of the pneumatic muscles. The angle of the arm with respect to the vertical is designated as θ, while θ_{top} and θ_{down} are the angles corresponding to the physical limit stops of the prototype. α_{up} is the angle formed with respect to the horizontal by the muscle designated as *up*, while α_{down} is the angle formed by the other muscle with respect to the horizontal.

Despite the good results obtained on simulation, the experimental tests showed lack of accuracy of the estimated value with regard to the torque actually exerted by the muscles. The main reason for this is that equation (2) does not contemplate hysteresis, which is a feature of the pneumatic muscles. The error assumed on ignoring the effects of hysteresis means the estimator is not applicable in cases where the arm moves freely. However, when the arm is blocked by collision, the pressures, and consequently the torque, increase in such a way that the measurement error is not critical.

In order to obtain an estimator that behaves correctly for free movement, with the system moving at low torque values, a development based on Newton's laws of motion is proposed. This new estimator calculates the torque by means of equation (5), which is in fact the development of Newton's second law of motion.

$$\tau_{newt} = m \cdot g \cdot L.\sin(\theta - \beta) - N.m_p \cdot g \cdot L_p \cdot \sin\left(\theta - \beta_p\right) + \ddot{\theta} \cdot I_o \tag{5}$$

where m is the mass of the arm, L the distance between the rotation point and the arm's centre of gravity, θ the angle between the arm and the vertical, β the angle between the rotation point and the centre of gravity, and I_o its inertia on the rotation point. N is the number of extra masses placed on the tip, and m_p is the weight of each mass ($N{\cdot}m_p$ thus represents the mass placed at the tip of the arm).

This torque estimator functions correctly but has two negative aspects. The first is that if the structure has two or more degrees of freedom instead of one, calculating the equation becomes rather complicated. The other aspect is that if any kind of interaction is produced with the environment, e.g. a collision, when the arm movement is blocked equation (5) is no longer of use for estimating the torque exerted by the actuators.

Knowing that the torque estimator based on Newton's law of motion calculates the torque correctly in the case of free movement, and that the moment evaluated by the torque estimator based on the muscle pressures can be acceptable when an interaction occurs with the environment, it was decided to create a hybrid torque estimator. This hybrid estimator requires some kind of observer to indicate whether an interaction with the environment has occurred, so that the desired estimator can be selected at each time. In short, when free movements are made the estimator used will be the one based on Newton's laws (τ_{newt}), and in the case of limited movements or interactions with the environment the estimator used will be the one based on the pressures of each muscle (τ_{pres}), switching between them where necessary.

The switch between the torque estimator and the force estimator is practically instantaneous: all that is needed is to apply next equation (6):

$$F_{est} = {}^{\tau_{est}}/_{l} \qquad (6)$$

Where τ_{est} is equal to τ_{newt} in the case of no interaction with the environment, or equal to τ_{pres} if there is a collision. l is the distance between the rotation point and the tip of the arm on which the force is to be calculated.

5 HYBRID POSITION/FORCE CONTROL OF THE SET-UP

Figure 5 shows a diagram of the Hybrid Control implemented. It basically consists of a supervisor block that switches between position control and force control according to the status of the interaction between the arm and its environment (in this case, the lower limit stop). The position control is thus carried out separately from the force control, i.e. the system is controlled by the position algorithm until the supervisor block detects that a collision has occurred. When this happens, it switches between the controls as soft as possible, activating the force control.

5.1 Position Control Algorithm

The hybrid control structure enables different, independent algorithms to be implemented for position and force control. In the last few years the authors have conducted research into the position control of this same experimental device.

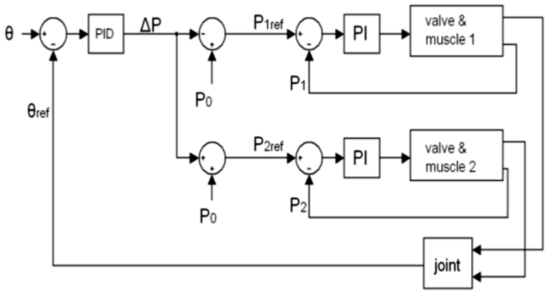


Figure 7: Diagram of the internal pressure loops based position control algorithm.

Owing to the fact that the results obtained with a classical PI controller were not good, due to its high non-linearity, other advanced control techniques were applied in order to correctly control the angular position. Firstly, a PID-based controller was enhanced with linear and non-linear internal loops. However, good performance requires the use of robust or non-linear control techniques (Thanh, 2006; Balasubramanian, 2007) and in this context, the application of different control techniques is found in the literature. Therefore, a robust linear control technique H∞ (Pujana-Arrese, 2008), and a robust non-linear technique, sliding-mode (Arenas, 2008), were applied.

Subsequently, based on an idea applied by Tsagarakis and Caldwell (2007), a new position controller was developed based on an internal pressure loop for each muscle. This new position algorithm requires the use of one servo-valve for each pneumatic muscle instead of one single valve for each DoF, as used with the position algorithms that were designed and implemented previously.

Although this new solution initially doubles the variables that have to be controlled for each degree of freedom, it can be considered as a single-variable approach for each joint. Based on the symmetrical co-contraction of the opposing muscles, an asymmetrical variation is set in the pressure of each muscle. Thus, based on an initial pressure (P_0) the setting is increased in one of the muscles and reduced by the same amount (ΔP) in the other.

$$P_{1ref} = P_0 - \Delta P; \quad P_{2ref} = P_0 + \Delta P \qquad (7)$$

Accordingly, from the control point of view, the position control problem is still SISO with the angular position of the joint (θ) as the output and the pressure variation (ΔP) as the input.

Figure 7 shows the position control schematic based on the internal loops that control the pressure

Table 1: Force PID parameters for the different impact points.

	64.8º	48.2º	37.5	27.2º
K_p	0.045	0.025	0.03	0.025
K_i	0.05	0.05	0.03	0.04
K_d	0.002	0.0002	0.001	0.0005

in each muscle, implemented by means of PI algorithms. As it has been already mentioned, the pressure set-point for each controller is set on the basis of an initial value (P_0), adding and subtracting the same quantity (ΔP). The value of this increase/reduction is the output of the most external loop of the controller (the position loop). This loop has also been implemented by means of a PID algorithm. The gains of both pressure loops were adjusted to the values K_p=4, K_i=4, and the gains of the position loop to K_p=0.21, K_i=1.2, K_d=0.04, being P_0=3 bar.

The experimental results for the hybrid position/force control shown later in this paper were obtained with the position algorithm based on the internal pressure loops tuned for a nominal load of 3 Kg placed at the tip.

5.2 Force Control Algorithm

Whereas direct measurement with the encoder located on the axis is the procedure used for position control, the value calculated by the force estimator is used to close the force control loop. The reaction of the device to any impact can thus be controlled without requiring the use of a sensor at the exact point of collision.

Unlike the case with position control, simulation-based tests confirm that to control the force correctly it is not necessary to design such complex algorithms. Also, the specifications of orthosis-type devices are much more restrictive for position control than they are for force control, in which it is normally sufficient for the force to be limited in the case of an inopportune collision.

The algorithm implemented is simply a PID tuned for a nominal load of 3 Kg. In any case, due to the non-linearity of the system, the response varies depending on the angular position in which switching between position control and force control occurs. With the aim of obtaining a more uniform response, four points distributed over the whole range of movement of the metal arm were selected. These five points divide the movement of the system into five different zones. By means of impacts applied to these points, four different PID algorithms were tuned. Finally, a gain scheduling type strategy was implemented, which linearly combines the outputs of the two PIDs delimiting the zone in which the collision occurs.

Table 1 shows the PID algorithm parameter values for each of the points expressed according to their angular situation with respect to the vertical plane.

6 IMPEDANCE CONTROL OF THE SET-UP

The classic impedance control formulation is based on the hypothesis that the actuation system is able to supply the torque required by the control algorithm, i.e. the impedance control output is the torque set-point for the actuator. This hypothesis is not so evident in the case of one degree of freedom actuated by means of an antagonistic pair of pneumatic muscles. Tsagarakis (2007) puts forward the same algorithm based on the independent pressure loops used for controlling the angular position, which, as can be observed in Figure 7, is the one that has been implemented. As explained above, the scheme has two separate PI controllers for controlling the internal pressure of each muscle and an external loop governed by a PID, which in this case serves to close the torque loop. Logically, the tuning of the pressure loops is the same as in the case of the position control, i.e. K_p=4, K_i=4, while the optimum values for the torque loop gains are K_p=0.12, K_i=0.6, K_d=0.0024.

The impedance control strictly speaking was implemented by adapting the general control law (1) to the experimental prototype presented in this study. The first step was to obtain a dynamic model of the system. In the field of robotics there are several methodologies for system modelling (including those of Newton-Euler and Lagrange-Euler), but in this case, given the mechanical simplicity of the prototype, the model was obtained by means of the physical equations.

First of all, the forces acting on the system needed to be identified. The intervening forces are the force of gravity and the two forces exerted by the pneumatic muscles. One of the muscles pulls upwards while the other pulls downwards. The resulting torque is the difference between them:

$$\sum \tau = I \cdot {}^{d\omega}\!/_{dt} \tag{8}$$

where I is the inertia of the body. When the existing forces had been established, the dynamic model was obtained by applying Newton's second law for the rotational movements:

$$\tau = -I \cdot {}^{d\omega}\!/_{dt} + m \cdot g \cdot L \cdot \sin\left[\theta - arctg\left(\frac{y_g}{x_g}\right)\right] \tag{9}$$

where m is the mass of the arm, θ the angle between the vertical and the arm, and L the distance between the centre of gravity and the rotation point. Finally, x_g and y_g are the coordinates of the centre of gravity with respect to the rotation point.

To fully complete the dynamic model of the system, all that remains is to calculate the inertia I. The inertia of the arm with respect to its centre of gravity should be calculated (10), and transformed with respect to the rotation point by means of Steiner's theorem (11).

$$I_z = \frac{m}{12} \cdot (l^2 + a^2) \tag{10}$$

$$I_o = I_z + d^2 \cdot m^2 \tag{11}$$

where l is the length and a the width of the arm, d is the distance between the rotation point and the centre of gravity, and m the mass of the arm.

It has not been taken into account up to now that plates with extra weight can be placed on one end of the arm. To do this, the terms corresponding to the extra masses should be inserted in the dynamic model.

$$\begin{aligned}\tau = -I \cdot {}^{d\omega}\!/_{dt} + m \cdot g \cdot L \cdot \sin\left[\theta - arctg\left(\frac{y_g}{x_g}\right)\right] + N \\ \cdot m_p \cdot g \cdot L_p \cdot \sin\left[\theta - arctg\left(\frac{y_p}{x_p}\right)\right. \\ \cdot \sin\left[\theta - arctg\left(\frac{y_p}{x_p}\right)\right]\end{aligned} \tag{12}$$

where N is the number of extra masses, m_p the weight of each extra mass, L_p the distance between the centre of gravity of the extra weights and the rotation point of the arm, and x_p and y_p are the coordinates of the centre of gravity of the extra weights with respect to the rotation point.

The next step is to establish the reference impedance (13). In this case, the parameters K, B and M are not matrixes but merely parameters simplifying the tuning process.

$$\tau_{int} + K(\theta_{ref} - \theta) + B(\omega_{ref} - \omega) = M\frac{d\omega}{dt} \tag{13}$$

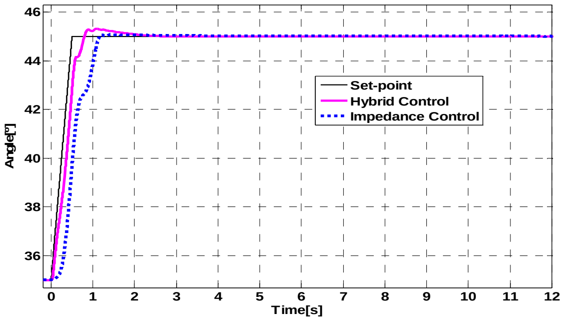


Figure 8: Experimental results in the intermediate displacement zone.

The impedance control law (14) to be applied, which is a simplified form of the general law (1), is obtained by equalling the term of the angular acceleration in (13).

$$\begin{aligned}\tau_{act} = -\frac{I}{M} \cdot K(\theta_{ref} - \theta) + S(\theta) - \frac{I}{M} \cdot B(\omega_{ref} - \omega) \\ -\frac{I}{M} \cdot \tau_{int}\end{aligned} \tag{14}$$

If the control law is analysed, apart from the parameters to be tuned (K, B and M), it can be observed that there is a term, τ_{int}, which is the torque measured when the arm interacts with the environment. This torque can be the estimated torque, as described for the hybrid control, but it is not essential in the control law. This term was not used in the law implemented, as the results were analogous. $S(\theta)$ represents the static part of the dynamic equation (12).

$$\begin{aligned}S(\theta) = m \cdot g \cdot L \cdot \sin\left[\theta - arctg\left(\frac{y_g}{x_g}\right)\right] + N \cdot m_p \cdot g \\ \cdot L_p \cdot \sin\left[\theta - arctg\left(\frac{y_p}{x_p}\right)\right]\end{aligned} \tag{15}$$

As can be observed, it is dependent on the mass at the tip, and it is therefore not robust to mass change. However, this mass can be estimated during the free movement phase. The design parameters K, B and M were experimentally adjusted and the following values were obtained: K=3.1, B=0 and M=0.5.

To close the torque loop, the most common procedure is to use the direct reading by means of a torque sensor placed on the actual rotation axis, as done by Tsasarakis (2007) and Jia-Fan (2008). However, as described in section 4 of this paper, an estimator was used to calculate the torque exerted by the antagonistic pair of muscles from the reading of their position and their internal pressures.

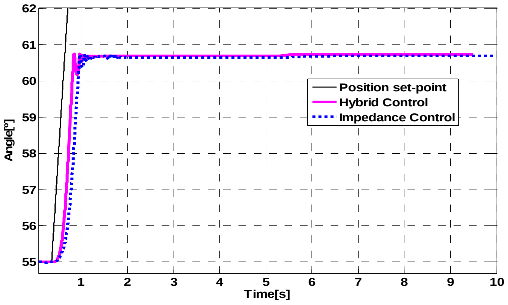


Figure 9: Position responses during a collision.

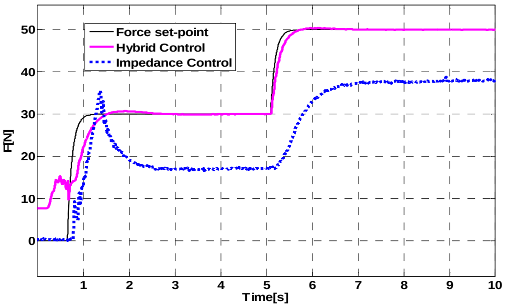


Figure 10: Estimated force responses during a collision.

7 EXPERIMENTAL RESULTS

Initially, controllers were tuned by simulation, using the non-linear model previously developed (Arrese-Pujana, 2007). Continuous-time controllers were discretized with sample time of 2 ms and then embedded into the control hardware.

With the objective of comparing the behaviour of both control techniques, Figure 8 shows the experimental response to a position step of 10º with a slope of 20º/s. This input signal has been applied to the intermediate zone within the motion range. Moreover, the arm has been loaded by attaching in its tip a weight of 3 Kg. This trial has been carried out in free movement without any collision with the environment.

The decoupling between the position and the force permits to adjust the hybrid control, thus it is possible to obtain a very fast position response which is almost identical in all the zones with a very little overshoot. Impedance control is not used to control explicitly the position, so, its response is more conservative because the same control law and the same tuning are used to control both the free movements and the collisions with the environment. Regarding the steady state error, both algorithms are able to eliminate it.

The trial showed in Figures 9 and 10 was performed in order to study the behaviour of both controllers with collisions. In this case, the arm is initially 55º away from the vertical plane and it is generated an ascendant position set-point with a slope of 20º/s. The lower limit stop was located at 61.5º, so when the arm tries to track the set-point and reaches this position a collision happens. Analyzing the dynamic of the position response (showed in Figure 9) it can be observed that the rebound happened after the collision has bigger amplitude with the hybrid control, due to the commutation between the position control and the force control. Figur 10 shows the estimate of the force executed by the pneumatic muscles against the lower limit stop when the collision happens. In fact, during the collision, controllers are not comparable to each other. Hybrid control carries out the tracking of a force set-point whereas the reaction force of the impedance control is proportional to the error between the position set-point and the collision point. Force set-point for the hybrid control is 30 N for the first 5 s and then increases up to 50 N. After some initial disturbances due to the impact and the commutation between the controllers, the estimated force increases smoothly until it reaches the set-point. This response could be faster, but the parameters were tuned to achieve an acceptable force control with a weight placed in the tip within the range of 0 to 6 Kg. In the impedance control the influence of the collision is lower, and after an initial peak the estimated forced is limited. After the first 5 s position set-point increases, and the control reacts to this rise increasing the torque as it is shown in the graph of the force.

The robustness of the controllers can be tested by changes in the weight placed at the extreme of the arm. In Figure 11 it can be observed that the tuning performed for the hybrid controller is suitable to manage collisions with different loads within the range 0 to 6 Kg, although in the upper limit of the range (with 6 Kg) a minor overshoot happens. Hybrid control offers a robust behaviour in free movements as well.

Impedance control does not offer a robust behaviour with changes in the load. In fact, the algorithm (14) needs to specify the weight of the mass placed at the extreme of the arm. To solve this problem, it was developed an initialization function which it is used to carry out a set of free movements are useful to estimate the extra weight. Thus, the control systems can work autonomously. Once the extra weight is estimated the results showed in Figure 12 are obtained for different tip masses during a collision and maintaining the same control parameters.

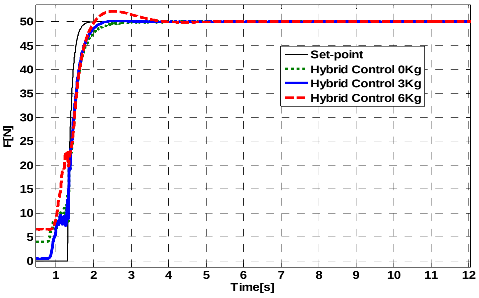


Figure 11: Estimated force response with different tip masses for hybrid control.

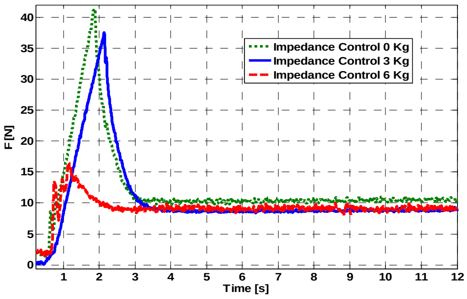


Figure 12: Estimated force response with different tip masses for impedance control.

8 CONCLUSIONS

This article is a study about the control issue of the interaction with the environment of a 1-DOF experimental set-up powered by pneumatic muscles. Due to the non linear behaviour of this kind of actuators the control of the mechatronic device is very complex both in free movements and when it comes into contact with an obstacle, having a different response depends on the movement zone or the position where the impact occurs. Moreover, the possibility of loading the extreme of the arm with extra weight requires using robust algorithms. The main contribution of this paper is the design and implementation of a torque/force estimator which is used to close the control loops. In spite of having a structural error derived from the equation used to model the force of the muscles, this inaccuracy only appears when the arm impacts with an obstacle and this is not critical because the dynamic is not affected.

ACKNOWLEDGEMENTS

The material used in this paper was partly supported by the Spanish Ministry of Education and Science and European FEDER Fund (research project DPI2006-14928-C02-01).

REFERENCES

Thanh, T-D.C., Ahn, K.K.,2006. Nonlinear PID control to improve the control performance of 2 axes pneumatic artificial muscle manipulator using neural network. *In Mechatronics,* no. 16, pp. 577-587.

Balasubramanian, K., Rattan, K.S., 2005. Trajectory tracking control of a pneumatic muscle system using fuzzy logic. *In NAFIP'2005, Annual Meeting of the North American Fuzzy Information Precessing Society*.

Pujana-Arrese, A., Riaño, S., Arenas, J., Martinez-Esnaola, A., Landaluze, J., 2008. H_∞ position Control of a 1-DoF Arm Powered by Pneumatic Muscles. *In the 8th Portuguese Conference on Automatic Control CONTROLO'2008*, 21-23 July 2008, Vila Real, Portugal.

Arenas, J., Pujana-Arrese, A., Riaño, S., Martinez-Esnaola, A., Landaluze, J., 2008. Sliding-mode Position Control of a 1-DoF Set-up based on Pneumatic Muscles. *In UKACC Control Conference CONTROL2008*, 2-4 Setember 2008, Manchester.

Tsagarakis, N.G., Caldwell, D.G., 2007. A compliant exoskeleton for multi-planar upper limb physiotherapy and trining. *In ICAR'07, The International Conference on Advanced Robotics,* 21-24 August 2007**,** Jeju Island, Korea.

Martinez, F., Retolaza, I., Lecue, E., Basurko, J., Landaluze, J., 2007. Preliminary design of an upper limb IAD (Intelligent Assist Device). *In the 9th* European Conference for the Advancement of Assistive *Technology, AAATE 2007,* October 3rd-5th, Donostia.

Martinez, F., Retolaza, I., Pujana-Arrese, A., Cenitagoya, A., Basurko, J., Landaluze, J., 2008. Design of a Five Actuated DoF Upper Limb Exoskeleton Oriented to Workplace Help. 2008. *In IEEE International Conference on Biomedical Robotics and Biomechatronics, IEEE BioRob 2008*, 19-22 October 2008, Scottsdale, Arizona, U.S.A..

Pujana-Arrese, A., Arenas, J., Retolaza, I., Martinez-Esnaola, A., Landaluze, J., 2007. Modelling in Modelica of a pneumatic muscle: Application to model an experimental set-up. *In the 21st European Conference on Modelling and Simulation ECMS2007*, 4-6 June 2007, Prague.

Raibert, M.H., Craig J.J., 1981. Hybrid Position/Force Control of Manipulators. *In Trans. of the ASME*, vol. 102.

Hogan, N., 1985. Impedance Control: An Approach to Manipulation : Part I - Theory. Part II - Implementation. Part III – Applications. *In Journal of Dynamic Systems, Measurement, and Control,* 107, 1-24.

Jia-Fan, Z., Can-Jun, Y., Ying, C., Yu, Z., Yi-Ming, D., 2008. Modeling and control of a curved pneumatic muscle actuatorfor wearable elbow exoskeleton. *In Mechatronics*, no. 18, pp. 448-457, 2008.

SCHEME FOR EVALUATION AND REDUCTION OF MOTION ARTIFACTS IN MOBILE VISION SYSTEMS

Christoph Walter[1], Felix Penzlin[2] and Norbert Elkmann[1]
[1] *Fraunhofer Institute for Factory Operation and Automation*
Sandtorstrasse 22, 39106 Magdeburg, Germany
[2] *Otto-von-Guericke-University Magdeburg*
Universitätsplatz 2, 39106 Magdeburg, Germany
{Christoph.Walter, Norbert.Elkmann}@iff.fraunhofer.de, felix.penzlin@st.ovgu.de

Keywords: Vision, Motion Blur, Mobile Robot, Congestion Control.

Abstract: Artifacts like motion blur are a common problem for vision systems on mobile robots, especially when operating under low light conditions or when using high-resolution sensors. In this contribution we present a scheme for estimating the degree of motion artifacts, especially motion blur, present in a stream of individual camera images. A single quality estimate is derived per frame using data from an inertial measurement unit. Considering limited image processing capacity of resource-constrained mobile robots, we show a number of data processing strategies that are based upon the idea of congestion control by adaptive image rejection.

1 INTRODUCTION

While the presence of motion artifacts in images from moving cameras can also be exploited in several ways, it is usually a troublesome effect. Objects may become unrecognizable because of blur; visual SLAM algorithms may yield poor results because of difficulties when finding corresponding image points or due to geometric distortion of the whole image.

At the same time, image processing tasks usually require significant resources and may easily exceed the capabilities of the computer hardware present on a mobile robot.

In the following sections we describe our approach to lessen the effects of both problems. At first we evaluate motion artifacts in greater detail. After discussing related work we present our method for estimating the image quality regarding the presence of motion artifacts. We then show data processing strategies including an approach to congestion control in persistent overload situations. We also present improvements of a specific vision task achieved with our system.

2 MOTION ARTIFACTS

Cameras acquire images by exposing a light-sensitive element for a given period of time. Camera movement while the sensor is exposed may result in a number of image artifacts. Lens distortion is considered to have a negligible impact and is therefore not modeled here.

2.1 Motion Blur

Motion blur can be induced by moving either objects in the camera’s field of vision or the camera itself. For simplicity we consider only a static scene and disregard any moving objects.

Camera Translation. We distinguish between two kinds of camera movement. On the one hand there is translation in direction of the optical axis; on the other hand there is motion in the plane orthogonal to that axis. In the second case, the magnitude of blur b on the image sensor for an object is in inverse proportion to the distance to the camera plane d_{cam} (See Figure 1).

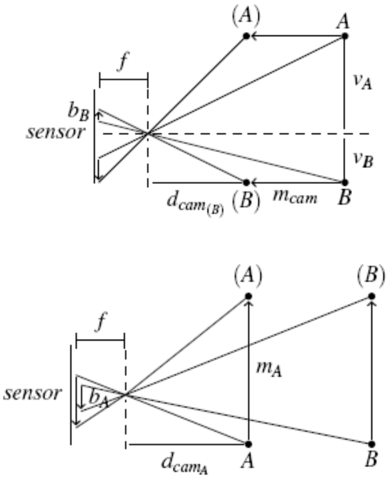


Figure 1: Motion blur in case of translation parallel to the optical axis (upper part) and vertical or horizontal translation (lower part).

For movements parallel to the optical axis the intensity of blur *b* for an object depends on its distances from line of view *v* and camera plane d_{cam} and the displacement m_{cam}. For a point at the optical axis, this kind of translation has no impact. If objects are relatively far away from the camera, translation becomes insignificant.

Camera Rotation. When rotating the camera, the magnitude of blur also depends on the position of a given object relative to the optical axis. Figure 2 shows that such a camera rotation results in a blur *b* roughly perpendicular to the axis of rotation. Its strength depends on the actual angle of rotation and on the angle between the rotation axis and the view direction. The distance to an object does not matter for rotational blur (See Figure 2).

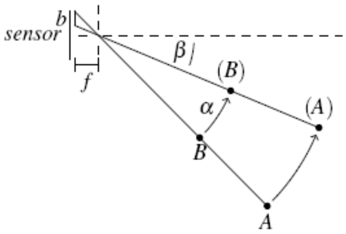


Figure 2: Motion blur in case of rotation.

2.2 Distortion

Geometrical image distortion is another common artifact that can be found with moving cameras. It occurs when different portions of the image sensor are exposed sequentially to light. This "rolling shutter"-mode is implemented in a number of CMOS-based cameras.

A sudden change in illumination may influence only portions of the image. If the camera is moved horizontally or rotated around the vertical axis, skew can be observed. Vertical lines appear to lean to the left for moving to the left or the right side for the opposite direction of movement. Vertical movements as well as rotations around the horizontal axis result in stretching respectively shrinking of objects vertically. Altering the direction of movement at a high speed (in case of vibrations) is called "wobble". When rotating the camera around the optical axis, straight lines get bent around the image center.

3 RELATED WORK

Two categories of techniques that are concerned with the problem of motion artifacts can be identified.

The first group of approaches is concerned with avoiding artifacts in the first place. Here, special hardware with accurate actuators is required. One solution is to stabilize the whole camera on a special platform, as shown in (Schiehlen, 1994). Other solutions are shiftable image sensors (Yeom, 2007), (Cardani, 2006), (Chi-Wei Chiu, 2007) that counteract camera shake or agile optical components like a variable fluid prism (Sato, 1993), a movable mirror (Günther, 2006) or an additional lens between scene and sensor.

A quite simple solution found in some recent hand-held cameras tries to delay image exposure in case of camera shaking. Camera movement is determined using acceleration sensors.

A different group of solutions is not concerned with preventing artifacts during image acquisition, but tries to undo artifacts at a later stage by using image processing techniques. For instance one approach (Choi, 2008) merges blurred and sharp but under-exposed images of the same scene to obtain an overall improved image. In general, a correction in software is time-consuming and consists of two steps. In the first step the artifacts are identified, in a second step they are removed from the image. A number of algorithms for global shutter (Ji, 2008), (Fergus, 2006) and for rolling shutter (Cho, 2008), (Nicklin, 2007), (Chun, 2008) cameras have been developed. In contrast to our implementation, most of these approaches are, beside their excessive computational cost, limited to simple motions.

4 IMAGE QUALITY ESTIMATION

We estimate motion artifacts by measuring the movement of the camera during image exposure. This eliminates the need for additional image processing to detect artifacts, which is beneficial for resource constraint mobile systems.

4.1 Sensor Configuration

In our system, motion tracking is done by an inertial sensor containing a MEMS gyroscope which can measure three-dimensional angular velocity and a MEMS three-axis-accelerometer. Measurements are discretized in time. As discussed in section 2, the effect of motion varies depending on the setup. Camera translation results only in marginal artifacts for objects at medium and large distances, which we consider prevalent in the vast majority of mobile robot scenarios. Additionally, the direction of view often is in coincidence with the driving direction. Here, camera movement as a result of the robot's linear motion can also be ignored. The position of the sensor relative to the camera is irrelevant for the measured angular speed as long as they are both firmly mounted on a rigid frame.

4.2 Tracing Motion Artifacts

In our approach we consider the intrinsic camera parameters and the position and orientation of the camera relative to the gyroscope. The direction of the view vector varies over the image. Therefore the effect of camera motion is different at every image point. It is calculated by first projecting points from image space into world space by using the camera's intrinsic matrix while considering its lens distortion (See Figure 3). The camera's depth of field could be used as a rough clue. As discussed earlier, rotation is the main cause of artifacts in many scenarios. Here, solely the view vector at the considered image points is relevant.

In a second step the measured motion is applied successively to the virtual camera while tracing the path in image space described by the projected point. The quality estimate for an image point is determined by the length of the path as well as by the maximum distance between any of two points on that path. We map the strength of motion artifacts to the interval from 0.0 to 1.0, where 1.0 stands for an immobile camera. The quality reaches 0.0 for infinitely high motion artifact strength.

As stated above, the direction of sight rays and thus the quality estimate varies over different image points. To get a single estimate for the entire image, several possibilities exist to combine the data. The simplest method is to only consider quality at the image center. Depending on the application, it is also possible to sample quality values over the entire image or from predefined areas of interest and then use the average or the lowest quality as the overall quality measure. When combining samples over a larger area it is possible to account for rotation around the optical axis, which could not be detected when only considering the image center.

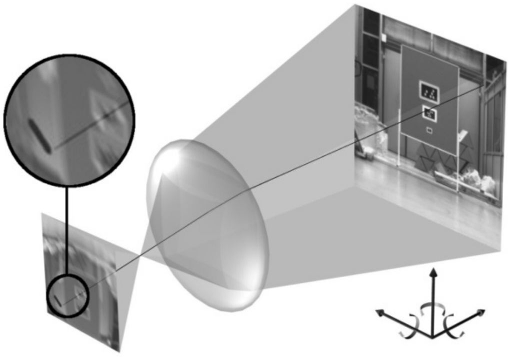


Figure 3: Basic principle of tracing motion artifacts.

5 DATA PROCESSING SCHEME

While researchers made various efforts to deal with motion artifacts, many of them are not well suited for mobile robot applications. Undoing blur is a slow and cumbersome process. Adaptive triggering of image acquisition depending on current camera movement is a promising and computationally inexpensive approach. However, a sudden increase in camera movement during exposure cannot be predicted.

In our system, we chose to continuously acquire images as well as motion data and apply a selection process at a later stage. One advantage is that the actual movement of the camera during image exposure is known for every individual image. Another advantage is that 'bad' pictures are not prevented from being acquired. A scheme for rejecting individual frames at an early processing stage is applied instead. Depending on the application, blurry images may still be used if continuous image degradation happens for a prolonged period of time.

5.1 Image Acceptance Test and Congestion Control

We consider mobile platforms to have in general a limited computing capacity. At the same time we assume that image processing tasks consume significant resources. Here, the frame rate of a camera may easily exceed the image processing capacity.

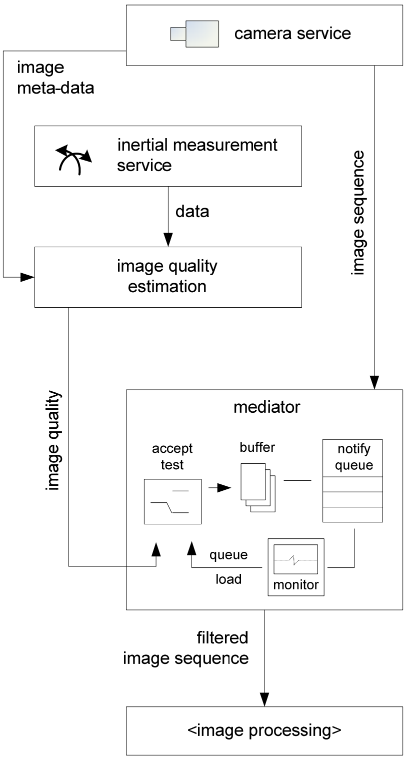


Figure 4: Structure of a system containing a mediator service for congestion control.

As shown in Figure 4, instead of dropping random frames, we can apply a simple, yet effective congestion control scheme. Having a normalized image quality value available for every frame enables us to compare it directly to the current system load. Images are only accepted for further processing if their image quality is greater than the system load indicator. This implies that in case the system is idle, even images containing heavy motion artifacts are accepted for further processing. This ensures that it is not possible to reject every image for an extended period of time. The system load indicator is derived from the percentage of utilized space in the image queue right before the image processing stage mapped to the interval from 0.0 to 1.0.

5.2 Data Processing Strategies

The basic data processing scheme has been described in the previous section. However, as in the computation of a global quality estimate for the entire image, some variations or extensions are also possible at this stage.

Minimum Quality. If an application requires that the degree of motion artifacts doesn’t exceed a certain degree, it is possible to specify a minimum quality value at the mediator to prevent it from accepting low quality images.

Binning Images. Scenarios where degraded images can still be useful if they are processed in an alternative way compared to artifact-free images can also be supported. Here, the mediator shown in Figure 4 can be extended to route images with quality estimates below a certain threshold to another processing module. An example of such a scenario may be a system for visual odometry where images are too blurry to match corresponding features in successive frames but can be used to derive camera motion from blur instead. Another scenario is the combination with deblurring algorithms. To prevent starvation of image processing in case of unacceptable blurry images for a long period of time, it is possible to route individual images through a deblurring stage when the load indicator becomes low.

Reusing Quality Estimates. It may also be desirable of course to include the quality estimate or even the computed paths of movement at individual image locations in later processing stages. The overall quality measure can be easily included in the image metadata at the mediator stage. If access to more detailed data is required, it is more suitable to establish an additional connection between the quality estimation module and the processing stage and access desired data on demand.

6 SYSTEM EVALUATION

In this section we show results achieved with our approach to image quality estimation. We also present improvements of a scenario where markers are to be detected by a mobile robot while driving on a bumpy floor.

6.1 Evaluation of Motion Artifact Detection

In a first experiment we examined the correlation between motion blur in camera images and the computed quality based on angular rate measurement. A mobile robot was equipped with a front facing camera and was driving towards a board placed in front of it. The robot passed various bumps of a maximum height of 1 cm, which resulted in displacement as well as rotation of the robot. Figure 5 shows the correlation between blur radius at the image center and computed quality values. The blur radius was measured manually in each individual image.

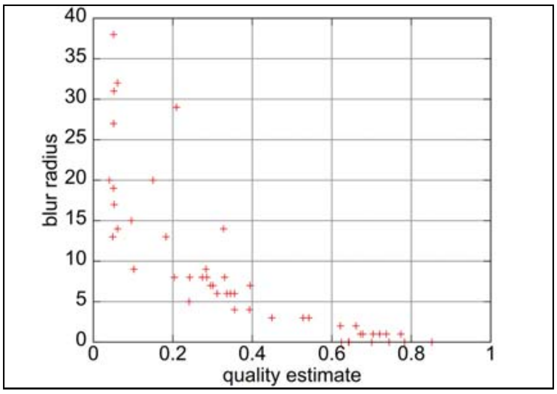


Figure 5: Correlation between quality estimate and actual strength of motion blur.

In Figure 6 small parts of a blurry and a non-blurry image is shown. The red dots indicate the motion calculated using inertial data only.

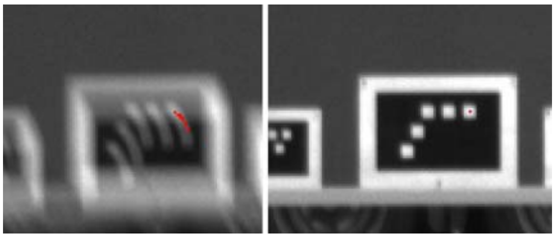


Figure 6: Motion blur traced by inertial data.

6.2 Improving a Marker Detection Scenario

We applied our approach to a scenario where optical markers were to be recognized by a moving mobile robot. The computing capacity onboard the robot is limited. Therefore not all images acquired by the onboard camera can be processed. The lights were set to about 300 lux, which resulted in an average integration time of 60 ms. A 2/3 inch monochrome CCD sensor and a 4.8 mm fixed focus C-Mount lens were used in the experiment.

The robot was approaching a board from a distance of approximately 13 meters. Markers of different sizes were attached to the board (See Figure 7). The goal when approaching the board was to recognize the markers as frequently as possible.

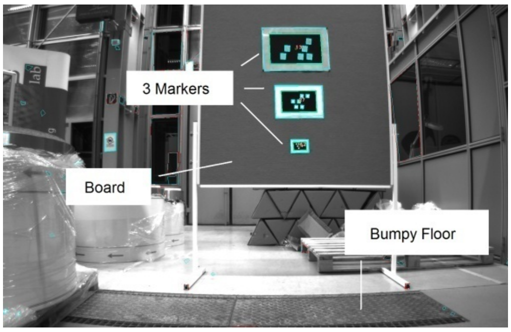


Figure 7: Test setup of the marker detection scenario.

Images and motion data were recorded in order to compare results achieved with uncontrolled frame drops due to queue overflow against results with dynamic congestion control. The total number of images acquired during the approach was 319. The average processing time per frame required by the marker detection algorithm was approximately 1.6 times the interarrival time of new images. Table 1 shows the number of images in which a marker could be identified for one particular approach. In general, markers could not be recognized in all 319 frames because at first they were too far away, went partially outside of the image, or they were obscured by motion blur. It can be seen that the improvement in the total number of images with recognized markers increases with the decreasing size of the marker. This is because smaller markers are easily obscured by blur.

Table 1: Improvements of recognition results when applying dynamic filtering.

Marker	Large	Medium	Small
recognizable	246	272	99
recognized(uncontrolled)	154	159	62
recognized(filtered)	161	178	71
Improvement [pct.]	4.5	11.9	14.5

7 CONCLUSIONS

Here we presented an approach to improve the performance of image processing tasks on mobile robots equipped with common fixed focus, low-cost cameras. The basic idea presented was to improve the quality of images processed by arbitrary vision algorithms by estimating the amount of motion artifacts for every image and rejecting bad ones while also considering a system load indicator.

Our system is suitable for resource-constrained robots where the camera's frame rate usually exceeds the processing capabilities of the onboard computer. Based on improvements we have seen in an example scenario, we are confident that the performance of a number of different image processing tasks can be improved through this approach.

ACKNOWLEDGEMENTS

This work has been funded in part by the German Federal Ministry of Education and Research under grant 01IM08002.

REFERENCES

Schiehlen, J., Dickmanns, E. D.: Design and control of a camera platform for machine vision, In: Proc. of IEEE Int. Conf. on Intelligent Robots and Systems '94, pp. 2058-2063, 1994.

Yeom, D., Park, N., Jung, S.: Digital controller of novel voice coil motor actuator for optical image stabilizer, In: Proc. of Int. Conf. on Control, Automation and Systems '07, pp. 2201-2206, 2007.

Cardani, B.: Optical image stabilization for digital cameras, In: IEEE Control Systems Magazine vol. 26 no. 2, pp. 21-22, April 2006.

Chi-Wei Chiu, Chao, P. C. - P., Din-Yuan Wu: Optimal Design of Magnetically Actuated Optical Image Stabilizer Mechanism for Cameras in Mobile Phones via Genetic Algorithm, In: IEEE Transactions on Magnetics vol. 43 no. 6, pp. 2582-2584, June 2007.

Sato, K., Ishizuka, S., Nikami, A., Sato, M.: Control techniques for optical image stabilizing system, In: IEEE Transactions on Consumer Electronics vol. 39 no. 3, June 1993.

Günthner, W., Wagner, P., Ulbrich, H.: An inertially Stabilised Vehicle Camera System. Hardware, Algorithms, Test Drives, In: Proc. of IEEE Conf. on Industrial Electronics '06, pp. 3815-3820, 2006.

Ji, H., Liu, C.: Motion blur identification from image gradients, In: IEEE Conf. on Computer Vision and pattern Recognition '08, pp. 1-8, 2008.

Fergus, R., Singh, B., Hertzmann, A., Roweis, S. T., Freeman, W. T.: Removing camera shake from a single photograph, In: Proc. of ACM SIGGRAPH '06, pp. 787-794, 2006.

Cho, W. H., Hong, K.-S.: A fast CIS still image stabilization method without parallax and moving object problems, IEEE Transactions on Consumer Electronics vol. 54 no. 2, May 2008.

Nicklin, S. P., Fisher, R. D., Middleton, R. H.: Rolling Shutter Image Compensation, In: Lecture Notes in Computer Science vol. 4434/2007, Springer, 2007.

Chun, J.-B., Jung, H., Kyung, C.-M.: Suppressing rolling-shutter distortion of CMOS image sensors by motion vector detection, IEEE Transactions on Consumer Electronics, vol.54, no.3, 2008.

Choi, B.-D., Jung, S.-W., Ko, S.-J.: Motion-blur-free camera system splitting exposure time, IEEE Transactions on Consumer Electronics, vol.54, no.3, 2008.

MODEL-DRIVEN DEVELOPMENT IN INDUSTRIAL AUTOMATION

Automating the Development of Industrial Automation Systems using Model Transformations

Mathias Maurmaier and Peter Göhner
Institute of Industrial Automation and Software Engineering, Universität Stuttgart
Pfaffenwaldring 47, 70550 Stuttgart, Germany
{mathias.maurmaier, peter.goehner}@ias.uni-stuttgart.de

Keywords: Model-driven Development, MDE, Model Transformation, Domain-specific Modeling Language, Industrial Automation Systems.

Abstract: The complexity of modern automation systems is growing steadily. In software engineering, model-driven development proved that it contributes significantly to cope with this complexity in development, while increasing efficiency and the quality of the development results. However, hardware-software dependencies, different types of requirements that must be considered in development and the large number of modeling languages are specific challenges for a model-driven approach in automation technology. In this paper a concept of model-driven system development is presented that takes into account these challenges, and thus provides the possibility to leverage model-driven development in industrial automation technology.

1 INTRODUCTION

Because of the integration of new functions and technologies necessary to fulfill customer requirements, environmental regulations or safety standards, the complexity of automation systems increases steadily (Ramebäck, 2003). Therefore, adequate development methods are needed providing the means to increase efficiency in development and to increase the quality of the development results. In software engineering, model-driven development has proven that it offers mechanisms to cope with the increasing complexity and to boost efficiency in development (Schmidt, 2006).

In this paper we analyze, why model-driven development is rarely used for the development of industrial automation systems and present a concept based on extended model transformations that allows to benefit from the advantages of model-driven development in industrial automation technology. Therefore we discuss specific challenges in development of industrial automation systems in chapter 2. Starting with a theoretical consideration of system development and the theory of model-driven development in chapter 3, the limiting factors regarding the application for automation systems development are presented in chapter 4. Based on the limitations a concept of model-driven development for industrial automation systems is deducted.

2 CHALLENGES IN INDUSTRIAL AUTOMATION

Industrial automation systems are complex hardware-software systems, whose objective is the control and supervision of a technical process. Many different disciplines as software engineering, hardware development, or electrical engineering are involved in the development of such systems. Modeling is seen as an important lever for coping with the complexity in, since any model allows to focus on specific aspects. Therefore, for system development, many different models are used. Since dependencies between the various models are not automatically managed, there is a high manual effort for multiple entries of the same information and for ensuring consistency of the models (Schenk and Schlereth, 2008). Therefore, we need concepts to manage the dependencies between the various models used during development.

In contrast to software engineering, there is no established, universal modeling language such as the UML in industrial automation technology. This is due to the various disciplines involved in development, using specific modeling languages tailored to their needs. Thus, the development method must be open to the use of different modeling languages.

The hardware of an industrial automation system consisting of sensors, actuators and processing units, and the software are highly integrated, leading to a large number of dependencies in development. In order to increase efficiency and the quality of the developed industrial automation systems, reusable partial solutions should be used whenever possible. These partial solutions consist both of hardware parts and a specification or implementation of the related software parts. Hence, the development method for industrial automation systems must provide reusable partial solutions and has to offer concepts to manage hardware-software dependencies.

In the development of an industrial automation system different types of requirements must be fulfilled. First of all, the technical process needs to be controlled and monitored. Further specific requirements for the automation system arise from the realization of the technical system, from existing legacy systems as well as from legal and economic constraints. These requirements can demand a specific property of the overall automation system as for example reduced energy consumption or a property of an individual subsystem, e.g. the manufacturer of a specific subsystem or a specific bussystem to be used. These requirements vary between different automation systems that realize the same technical process. Any development method in industrial automation technology must support these different kinds of requirements.

3 SYSTEMS DEVELOPMENT

3.1 Basic Definitions

According to Smith and Browne (1993) development is the creation of a system or artifact to solve a given problem. Therefore representations of the real world are used. In the representation of the problem R(P) the problem is described using the vocabulary, concepts and metrics of the problem space. Starting from this representation the representation of the solution R(L) is developed in several development steps. Based on the representation of the solution R(L), the system can be produced. There may be representations of different formalization degrees. A textual requirements specification, for example, is an informal representation of the problem to be solved; the source code of the automation software is a formal representation of the solution.

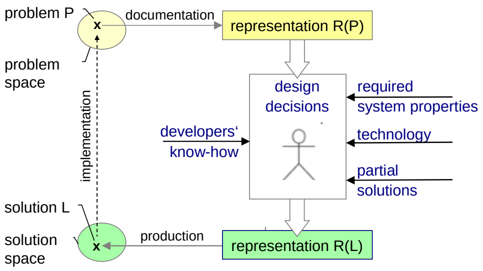


Figure 1: Systems development.

The system to be developed consists of a set of individual elements, which are related to each other and interact in a way, that they fulfill a common goal (Sommerville, 2007). The elements can be atomic or subsystems, i.e. in this case they are systems as well. The system properties comprise both the properties of the individual elements, e.g. the manufacturer of the element or subsystem, and the properties that arise from the interplay of the elements as for example energy consumption or reliability. The latter are known as global system properties.

Considering the requirements to be fulfilled by a system, they can focus on different aspects: First of all, there are requirements describing the problem to be solved by the system. Other requirements focus on the global properties, the system must prove. Examples for this class are requirements concerning energy consumption. A third class of requirements prescribes properties of individual elements. These latter two classes of requirements restrict the number of possible solutions of the problem.

During development, a multitude of design decisions have to be made, which affect the solution. These decisions are made by the engineer combining his own knowledge and expertise with the required system properties, available technologies and reusable partial solutions. In decision-making the engineer has to consider all interdependencies between the technologies and partial solutions. If there is a modification in the problem to be solved, in required system properties or in the partial solutions, all decisions must be checked manually for correctness under the new circumstances.

The aim of model-driven development is to automatically generate the representation of the solution R(L) from the representation of the problem R(P). Therefore the basics of model-driven development are described in the following section.

3.2 Model-driven Development

In model-driven development, the course of the development from problem analyses to design and implementation is defined by models (OMG, 2003), whereas a model is a formalized representation of some aspects of the problem to be solved or the system to be developed. The formalized representation of the problem is called conceptual model M(P). This model has a high level of abstraction. As shown in figure 2, transformations are used for the automated generation of the model of the solution M(L) out of the conceptual model M(P). A transformation is the conversion of a source model into a target model based on a formalized specification. As prerequisite for an automated transformation, all factors affecting the solution must be formalized.

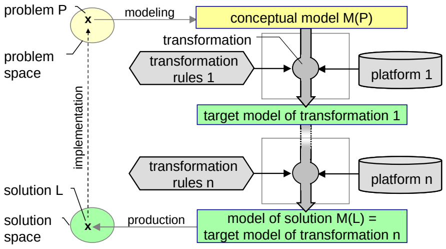


Figure 2: Model-driven systems development.

To implement a transformation, a platform and transformation rules are needed. The platform contains all available reusable partial solutions of one dedicated level of abstraction, whose interfaces and properties are described in a formalized way.

The platforms used in the various transformations determine much of the system properties. In reverse requirements concerning system properties such as high reliability or the operating system to be used determine the platforms to be chosen for the development of the system.

Transformation rules formalize the part of the developer’s knowledge needed for the selection, linkage and configuration of the partial solutions of the platform, which are needed to implement the problems that can be modeled in the source model. When executing a transformation, the transformation rules link the problem modeled in the source model with the partial solutions of the platform.

Since the transformation rules link the model elements of the source model with the partial solutions of the platform, they depend on the modeling language of the source model and the description of the partial solutions. This is why standards for the specification of transformation rules such as QVT (OMG, 2008) define the structure of the rules using metamodels. These standards are very generic and require high efforts to define the necessary transformation rules when introducing model-driven development in a specific domain. Hence, currently transformations are usually realized for specific modeling languages and platforms within integrated development environments. Since in software engineering there are standard modeling languages as for example the UML or signal flow diagrams, model-driven development techniques are applied especially in application domains in software engineering that traditionally use these modeling languages.

The identification of the necessary transformations and the modeling languages used within a domain as well as the development of platforms and transformation rules takes place in a preceding, project-independent development phase, called infrastructure development. In this phase, domain engineering activities are executed in order to develop the artifacts, which will be reused within various projects later in the system development phase. In the system development phase, the platforms and transformation rules are reused.

In model-driven development dependencies between different views and partial solutions are used to define transformations allowing an automated generation of more detailed models, or source code from other usually more abstract models. In the following chapter, we discuss the limitations of the current transformation concepts regarding the application for the development of industrial automation systems and propose concepts allowing to leverage the advantages of model-driven development in industrial automation technology.

4 MODEL-DRIVEN DEVELOPMENT OF AUTOMATION SYSTEMS

4.1 Model-driven Development and Challenges of Industrial Automation Technology

In model-driven development platforms are used to manage reusable partial solutions, which are described in a formalized way and stored in a clear structure within the platforms. Partial solutions consisting of hardware-software parts, as needed in automation technology, are not supported by existing model-driven approaches. Therefore, an extension of the classical model-driven development approach is needed to support partial solutions with hardware and software parts and to be able to manage hardware-software dependencies.

In classical model-driven development, the solution is built based on the conceptual model M(P) and the selected platforms. Platforms and transformation rules are not adaptable in system development. Hence, besides the conceptual model M(P) and the selection of the platforms, there are no further possibilities to influence the solution in the system development phase. Requirements concerning system properties can only be fulfilled, if there is a platform, which has exactly the required properties. As in automation technology there are a lot of requirements focusing on system properties, we would need an enormous number of platforms for any combination of required system properties. As this is not feasible in practice, we propose an extension of the classical model-driven development approach allowing to specify required system properties and thus to influence the solution.

Existing implementations of transformations cannot be used in automation technology due to the variety of modeling languages and the close hardware-software dependencies. To reduce the efforts in infrastructure development and to make the concept applicable to engineers, the generic concepts for the definition of transformation rules must be concretized. Standards as AutomationML (Drath et al., 2008) define metamodels for the description of automation technical solutions. As engineers are getting familiar with these standards, the description of the platform elements is based on these standards.

The integration of these approaches into a concept for hardware-software integration is described in the following paragraph in more detail.

4.2 Platforms for Hardware-Software Integration

In industrial automation technology, reusable partial solutions are composed of hardware and software parts. On the level of models, this means that the use of a partial solution has an impact both on hardware models (e. g. circuit diagram) and on software models such as source code. One partial solution has representations in several models. In order to integrate the representations belonging to one partial solution the concept of platforms is extended by views. The platform for model-driven development in automation technology consists not only of software components but integrates partial automation technical solutions of one abstraction level. An automation technical partial solution is created by the encapsulation of the representations of the partial solution for the hardware and the software models and a general description. For example, the partial solution representing a speed sensor is composed of a representation for the circuit diagram, for the simulation model and a software driver.

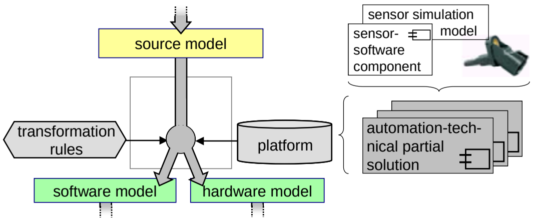


Figure 3: Transformation generating the hardware and software model.

All automation technical partial solutions of one abstraction level are merged into one platform. If, during the execution of a transformation, a partial solution is selected by a transformation rule, the corresponding representation of the partial solution is instantiated in any model generated by the transformation. Figure 3 illustrates the course of a transformation, which generates several target models from one source model.

4.2.1 Unified Metamodel of Automation Technical Partial Solutions

The knowledge, which partial solution should be selected during the execution of a transformation, how it must be connected and configured, is encapsulated in the transformation rules. The selection, linkage and configuration dependent on many parameters. The required properties of a partial solution are extracted from the source model. Then, the suitable partial solution is selected from the platform. To avoid that for any combination of required properties a new transformation rule must be created, the properties of any partial solution are subsumed within its formalized description. Thus transformation rules can select partial solutions by a set of required properties. This leads to a greatest possible decoupling between the transformation rules and the partial solutions allowing to add new partial solutions to the platform without having to modify the transformation rules. The description of each partial solution follows the metamodel presented in figure 4. Any partial solution disposes of features that are either fixed (property), optional (option) or can be configured (parameters).

The representations of a partial solution in different types of target models are encapsulated in the partial solution (target model representation).

Each representation is assigned to a specific view (View). A partial solution is linked with other partial solutions using ports (Port).

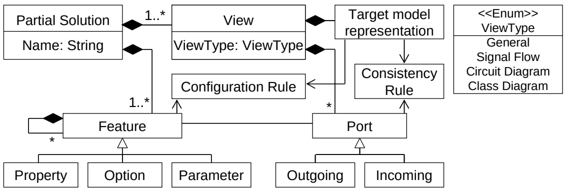


Figure 4: Metamodel of an automation technical partial solution within a platform.

Since dependencies between the hardware and software parts of a partial solution are known best by the developer of the partial solution, these can be integrated as consistency rules. Configuration rules adapt the representations within the different types of target models to the actual configuration of the partial solution. For example, if there is a parameter allowing to adjust the value range of a sensor, the configuration rules set the configuration values in the source code of the driver and adjust the values of the series resistor in the circuit diagram. When a partial solution is selected, linked or configured during the execution of a transformation, these rules are evaluated. This allows automatic internal adjustments of the partial solution to ensure consistency between the different views.

4.2.2 Transformation Rules

One benefit of the unified metamodel of the partial solutions within the platforms is the possibility to define a general structure of the transformation rules in automation technology. There are three major types of transformation rules:

Transformation rules that encapsulate the knowledge to define the structure of the target model, which allows solving the problem modeled in the source model, are called structure-defining (SD) rules. They extract the information from the source model, which is relevant for the structure of the target model, and create a possible structure of the target model. In further steps, this structure has to be detailed using partial solutions from the platform.

Selective (SL) transformation rules are used for the selection of the correct partial solutions from the platform. They encapsulate the knowledge about the relevant properties, a partial solution needs to have in order to be deployed in a certain position within the structure of the target model. Selective transformation rules define in a first step a complete requirements specification for any position in the structure of the target model that has to be detailed by a partial solution. In the second step, the requirements aggregated in the first step are used to select the correct partial solution from the platform.

Transformation rules of the third type, called configurational (CF) rules, ensure a consistent configuration of the selected partial solutions within all target models.

When a transformation is executed, the structure-defining rules are applied first, followed by the selective rules and in the end the configurational rules. If a set of transformation rules produces several possible solutions for a given source model, there is the possibility to optimize the solution concerning specific system properties.

Platforms and transformation rules are developed in infrastructure development preceding the systems development, where they are reused within many systems. This allows increasing efficiency in development and shortening the development time within individual projects. Crucial disadvantages are the high initial costs for the project-independent development of transformation rules and platforms, and the inflexibility regarding the consideration of requirements concerning individual system properties in the transformations. A solution to this problem is presented in the following section.

4.3 Adaptability of Transformations

In order to be able to specify the different kinds of requirements and thus to fulfill them, the concept of transformations of the classical model-driven development is extended by two additional variation points. These variation points can be used to modify the solution. In analogy to framework theory, these variation points are called hot spots of the transformation. The first hot spot allows to adapt the transformation rules to the requirements of a specific project. This adaptation allows to model and thus to respect required global system properties within the transformation, i.e. properties that arise from the interaction of the system elements. During the tool-supported execution of the transformation different architectural variants of the solution can be analyzed in terms of meeting the required global system properties. Finally, the one that meets best the requirements is selected. For example, a reliable temperature measurement can be realized using one highly reliable sensor or a redundant measurement with two or even more standard sensors. The knowledge of the possible architectural variants and the calculation of the reliability is formalized in the transformation rules. Using the hot spot, these transformation rules are configured or even completed by new rules formalizing the know-how on the reliability that should be reached in the current project, on possible architectural alternatives and on their properties. The decision, which alternative is best suited within the specific project, is made during the execution of the transformation. If the properties concerning reliability of the available sensors are known, this decision for one architectural variant can be made automatically depending on the required reliability.

Required properties of individual system elements such as the supplier of a specific partial solution can be specified using the second hot spot. This hot spot allows to access all properties of the partial solutions of the platform and thus to select or exclude specific partial solutions by any property. The project-specific adaptation of the platform at this hot spot prepares the set of partial solutions from the platform, which may be used in the specific development project.

Furthermore, the concept of hot spots allows upgrading the transformation rules or the platform within a development project. Thus, model-driven development can be introduced iteratively, since the missing transformation rules and partial solutions can be added at the corresponding hot spot.

5 CONCLUSIONS

In model-driven development the course of development is defined by models using the principle of abstraction to focus only on the relevant aspects in any development steps. Transformations are used to transform the abstract models of the problem to more detailed models of the solution.

The mature concepts of model-driven software development can not be applied directly for the development of automation systems, since they miss concepts for the high variability concerning the requirements on specific system properties and hardware-software dependencies. Furthermore, a concept for model-driven development in industrial automation technology must be adaptable to different modeling languages as there are no standard languages as for example the UML.

In this paper we presented a concept for model-driven development of industrial automation systems. Central aspects of this concept are platforms built on automation technical partial solutions, hot spots in the transformations and a metamodel based definition of automation technical partial solutions. The platforms allow reusing partial solutions consisting of hardware and software parts. Hardware-software dependencies are modeled by the developer of the partial solution and are taken into account when using a partial solution within a transformation. Hot spots enable the developer to adapt the transformation in order to fulfill requirements concerning specific system properties. The metamodel-based definition of the partial solutions allows to reuse transformation rules and to adapt them to different modeling languages.

The presented concept allows applying model-driven development to the development of industrial automation systems and leads to a better controllability of the complexity, increases efficiency and shortens the development time within individual industrial automation projects.

REFERENCES

Drath, R., Peschke, J., Lips, S. (2008). *AutomationML Top-level Architecture Document*. AutomationML consortium, Stuttgart.

International Organization for Standardization (1997). *Flow diagrams for process plants -- General rules*. ISO 10628:1997, Geneva: ISO.

Maurmaier, M. (2007). Model-driven system development in product automation. In *VDI-Bericht Nr. 1980 mit CD 'GMA-Kongress 2007 Automation im gesamten Lebenszyklus' ISBN 978-3-18-091980-5*. VDI-Verlag, Düsseldorf.

Maurmaier, M. (2008). Leveraging Model-driven Development for Automation Systems Development. In *Proceedings of IEEE Conference on emerging technologies and factory automation*. Hamburg.

Object Management Group (2003). *MDA Guide Version 1.0.1*.

Object Management Group (2008). *Meta Object Facility (MOF) 2.0 Query/View/Transformation*, Version 1.0.

Ramebäck, C. (2003). Process automation systems-history and future. In *Proceedings of IEEE Conference on emerging technologies and factory automation*. Lisbon.

Schenk, B., Schlereth, M. (2008). Model Driven Development applied to Automation Engineering. In *VDI-Bericht Nr. 2032 mit CD 'AUTOMATION 2008 - Lösungen für die Zukunft' ISBN 978-3-18-092032-0*. VDI-Verlag, Düsseldorf.

Schmidt, D. (2006). Model-Driven Engineering. In: IEEE Computer 39 (2), 25–31.

Smith, G., Brown, G. (1993). Conceptual Foundations of Design Problem Solving. In *IEEE Transactions on Systems, Man, and Cybernetics*; 23 (5), 1209-1219.

Sommerville, I (2007). *Software Engineering*. 8th ed., Pearson Education Limited, Harlow.

APPEARANCE-BASED DENSE MAPS CREATION

Comparison of Compression Techniques with Panoramic Images

Luis Payá, Lorenzo Fernández, Óscar Reinoso, Arturo Gil and David Úbeda
Departamento de Ingeniería de Sistemas Industriales, Miguel Hernández University
Avda. de la Universidad s/n. 03202, Elche (Alicante), Spain
{lpaya, o.reinoso}@umh.es

Keywords: Robot Mapping, Appearance-based Methods, Omnidirectional Vision, Spatial Localization.

Abstract: The visual information captured by omnidirectional systems is very rich and it may be very useful for a robot to create a map of an environment. This map could be composed of several panoramic images taken from different points of view in the environment, and some geometric relationships between them. To carry out any task, the robot must be able to calculate its position and orientation in the environment, comparing his current visual information with the data stored in the map. In this paper we study and compare some approaches to build the map, using appearance-based methods. The most important factor of these approaches is the kind of information to store in order to minimize the computational cost of the operations. We have carried out an exhaustive experimentation to study the amount of memory each technique requires to build the map, and the time consumption to create it and to carry out the localization process inside it. Also, we have tested the accuracy to compute the position and the orientation of a robot in the environment.

1 INTRODUCTION

When a robot or a team of robots have to carry out a task in an environment, an internal representation of it is usually needed so that the robot can estimate its initial position and orientation and navigate to the target points. Omnidirectional visual systems are a widespread sensor used with this aim due to their low cost and the richness of information they provide. Extensive work has been carried out in this field, using the extraction of some natural or artificial landmarks from the images to build the map and carry out the localization of the robot (Thrun, 2003). However, these processes can be carried out just working with the images as a whole, without extracting landmarks nor salient regions.

These appearance-based approaches are useful when working in unstructured environments where it may be hard to create appropriate models for recognition, and offer a systematic and intuitive way to build the map. Nevertheless, as we do not extract any relevant information from the images, an important problem of such approaches is the high computational cost they suppose.

Different researchers have shown how a manifold representation of the environment using some compression techniques can be used. A widely extended method is PCA (Principal Components Analysis), as (Kröse, 2004) does to create a database using a set of views with a probabilistic approach for the localization inside this database. Conventional PCA methods do not take profit of the amount of information that omnidirectional cameras offer, because they cannot deal with rotations in the plane where the robot moves. (Uenohara, 1998) studied this problem with a set of rotated images, and (Jogan, 2000) applied these concepts to an appearance-based map of an environment. The approach consists in creating an eigenspace that takes into account the possible rotations of each training image, trying to keep a good relationship between amount of memory, time and precision of the map. Other works rely on Fourier Transform to compress the information, as (Menegatti, 2004), that defines the concept of Fourier Signature and presents a method to build the map and localize the robot inside it, or (Rossi, 2008), that uses spherical Fourier transform of the omnidirectional images.

The representation of an environment with appearance-based approaches can be separated in a low-level map, that represents a room with several images and a high-level map, which tries to modelize the spatial relationships between rooms and between rooms and corridors. (Booij, 2007) shows how these concepts can be implemented through a graph representation whose nodes are the images and whose links denote similarity between them and (Vasudevan, 2007) uses a hierarchy of cognitive maps where place cells represent the scenes through a PCA compression and the information provided by a compass is used to compute the connectedness between them. Appearance-based techniques constitute a basis framework to other robotics applications, as in route-following, as (Payá, 2008) shows.

2 REVIEW OF COMPRESSION TECHNIQUES USING OMNIDIRECTIONAL IMAGES

In this section, we outline some techniques to extract the most relevant information from a set of panoramic images, captured from several positions in the environment to map.

2.1 PCA-based Techniques

When we have a set of *N* images with *M* pixels each, $\vec{x}^j \in \Re^{Mx1};\ j = 1 \ldots N$, each image can be transformed in a feature vector (also named 'projection' of the image) $\vec{p}^j \in \Re^{Kx1};\ j = 1 \ldots N$, being *K* the PCA features containing the most relevant information of the image, $K \le N$ (Kirby, 2000). The PCA transformation can be computed from SVD of the covariance matrix *C* of the data matrix, *X* that contains all the training images arranged in columns (with the mean subtracted). If *V* is the matrix containing the *K* principal eigenvectors and *P* is the reduced data of size *K x N*, the dimensionality reduction is done by $P = V^T \cdot X$, where the columns of *P* are the projections of the training images, $\vec{p}^j$.

However, the database built in this way contains information only for the orientation the robot had when capturing each image but not for all the possible orientations on each point. (Jogan, 2000) presents a methodology to include this orientation information but acquiring just one image per position. When we work with panoramic images, we can artificially rotate them by just shifting the rows. This way, from every image $\vec{x}^j \in \Re^{Mx1};\ j = 1 \ldots N$ we can build a submatrix $\mathbf{X}^j \in \Re^{MxQ}$ where the first column is the original image, and the rest of them are the shifted versions of the original one, with a $2\pi/Q$ rotation between them.

When having a set of *N* training images, the data matrix is composed of *N* blocks, and the covariance matrix has the following form:

$$\mathbf{X} = \left[\mathbf{X}^1 \middle| \mathbf{X}^2 \middle| \ldots \middle| \mathbf{X}^N\right]$$
$$\Rightarrow\ C = X^T X = \begin{bmatrix} X^{11} & X^{12} & \ldots & X^{1N} \\ X^{21} & X^{22} & \ldots & X^{2N} \\ \vdots & \vdots & \ddots & \vdots \\ X^{N1} & X^{N2} & \ldots & X^{NN} \end{bmatrix} \qquad (1)$$

Where $\mathbf{X}^{ik} \in \Re^{QxQ}$ are circulant blocks. The eigenvectors of a general circulant matrix are the *Q* basis vectors from the Fourier matrix (Ueonara, 1998):

$$\begin{gathered} \vec{\omega}_i = \begin{bmatrix} 1 & \gamma^i & \gamma^{2i} & \ldots & \gamma^{(Q-1)i} \end{bmatrix}^T \\ i = 0, \ldots, Q, \quad \gamma = e^{-2\pi j/Q}, \quad j = \sqrt{-1} \end{gathered} \qquad (2)$$

This property allows us to compute the eigenvectors without necessity of performing the SVD decomposition of *C* (this would be a computationally very expensive process). This problem can be solved by carrying out *Q* decompositions of order *N*. The eigenvectors of *C* shall be found among vectors of the form:

$$\vec{v}_i = \left[\alpha_i^1 \vec{\omega}_i^T, \alpha_i^2 \vec{\omega}_i^T, \ldots, \alpha_i^N \vec{\omega}_i^T\right]^T, i = 0, \ldots, N \qquad (3)$$

where $\vec{\alpha}_i = \left[\alpha_i^1, \alpha_i^2, \ldots, \alpha_i^N\right]^T, i = 0, \ldots, N$ are the eigenvectors of the following matrix:

$$\Lambda = \begin{bmatrix} \lambda_i^{11} & \lambda_i^{12} & \ldots & \lambda_i^{1N} \\ \lambda_i^{21} & \lambda_i^{22} & \ldots & \lambda_i^{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_i^{N1} & \lambda_i^{N2} & \ldots & \lambda_i^{NN} \end{bmatrix} \qquad (4)$$

where λ_i^{jk} is an eigenvalue of X^{jk} corresponding to the eigenvector $\vec{\omega}_i$. As the matrix Λ has *N* eigenvectors, if we repeat this process for every $\vec{\omega}_i$ we can obtain *Q·N* linearly independent eigenvectors of *C*.

2.2 Fourier-based Techniques

2.2.1 2D Discrete Fourier Transform

When we have an image *f(x,y)* with N_y rows and N_x columns, the 2D discrete Fourier Transform is defined through:

$$\Im[f(x,y)] = F(u,v) = \\ = \frac{1}{N_y}\sum_{x=0}^{N_x-1}\sum_{y=0}^{N_y-1} f(x,y)\cdot e^{-2\pi j\left(\frac{ux}{N_x}+\frac{vy}{N_y}\right)} \qquad (5)$$

$$u = 0,\ldots,N_x - 1, v = 0,\ldots,N_y - 1$$

The components of the transformed image are complex numbers so it can be split in two matrices, one with the modules (power spectrum) and other with the angles. The most relevant information in the Fourier domain concentrates in the low frequency components. Furthermore, removing high frequency information can lead to an improvement in localization because these components are more affected by noise. Another interesting property when we work with panoramic images is the rotational invariance, which is reflected in the shift theorem:

$$\Im[f(x-x_0, y-y_0)] = F(u,v)\cdot e^{-2\pi j\cdot\left(\frac{ux_0}{N_x}+\frac{vy_0}{N_y}\right)} \qquad (6)$$

$$u = 0,\ldots,N_x - 1, v = 0,\ldots,N_y - 1$$

According to this property, the power spectrum of the rotated image remains the same of the original image and only a change in the phase of the components of the transformed image is produced, whose value depends on the shift on the x-axis (x_0) and the y-axis (y_0). It means that when two images are acquired from close points of the environment but with different headings for the robot, then, the power spectrum is very similar, and studying the difference in phases we could estimate the angle between the two orientations, using eq. (6).

2.2.2 Fourier Signature of the Image

If we work with panoramic images, we can use another Fourier-based compact representation that takes profit of the shift theorem applied to panoramic images (Menegatti, 2004). It consists in expanding each row of the panoramic image $\{a_n\} = \{a_0, a_1, \ldots, a_{N_y-1}\}$ using the Discrete Fourier Transform into the sequence of complex numbers $\{A_n\} = \{A_0, A_1, \ldots, A_{N_y-1}\}$.

This Fourier Signature presents the same properties as the 2D Fourier Transform. The most relevant information concentrates in the low frequency components of each row, and it presents rotational invariance. However, it exploits better this invariance to ground-plane rotations in panoramic images. These rotations lead to two panoramic images which are the same but shifted along the horizontal axis (fig. 1). Each row of the first image can be represented with the sequence $\{a_n\}$ and each row of the second image will be the sequence $\{a_{n-q}\}$, being q the amount of shift, that is proportional to the relative rotation between images. The rotational invariance is deducted from the shift theorem, which can be expressed as:

$$\Im[\{a_{n-q}\}] = A_k e^{-j\cdot\frac{2\pi qk}{N_y}}; \qquad k = 0, ..., N_y - 1 \qquad (7)$$

where $\Im[\{a_{n-q}\}]$ is the Fourier Transform of the shifted sequence, and A_k are the components of the Fourier Transform of the non-shifted sequence. According to this expression, the amplitude of the Fourier Transform of the shifted image is the same as the original transform and there is only a phase change, proportional to the amount of shift q.

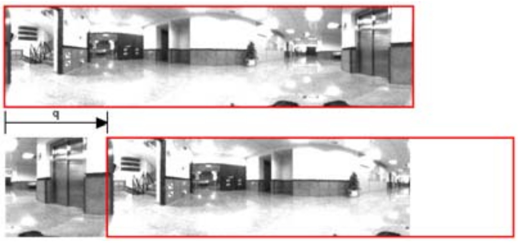


Figure 1: A robot rotation on the ground plane produces a shift in the panoramic image captured.

3 MAP BUILDING

To carry out the experiments, we have captured a set of omnidirectional images on a pre-defined 40x40 cm grid in an indoor environment, including an unstructured room (a laboratory) and a structured one (a corridor). We work with panoramic images with size 56x256 pixels. Once we have all the panoramic images, we have used the compression methods exposed in the previous section. Fig. 2(a) shows a bird's eye view of the grid used to take the images and two examples of panoramic images.

Before using the compression methods, a normalisation and a filtering process have been carried out to make the map robust to changes in illumination. There have been significant differences when using each one of the compression methods, regarding the elapsed time and the amount of memory the map takes up once it is built.

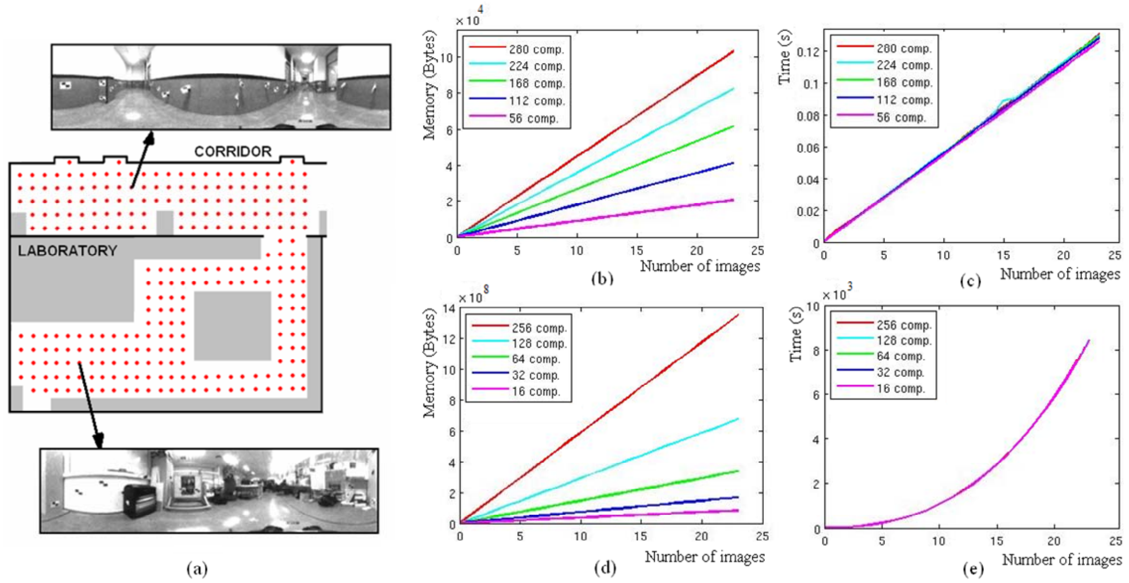


Figure 2: (a) Grid used to capture the training set of images, (b) amount of memory taken up and (c) time elapsed to build the map with the Fourier-based approaches, (d) amount of memory and (e) time elapsed with the PCA-based approach.

Fig. 2(b) shows the amount of memory the map constructed takes up, depending on the number of images in the map, and the number of Fourier Components we retain (those of lower frequency). When we work with the Fourier Signature, taking 224 components implies we retain 4 Fourier Components per row (56 rows in the image). In the case of 2D Fourier transform, 224 components means we take the firs 7 rows and 32 columns (where the main information is concentrated).

Fig. 2(c) shows the elapsed time to build the database, depending on the number of images in the map and the number of Fourier components we retain. The time elapsed is very similar when we use the Fourier Signature and the 2D Fourier Transform. On the other hand, fig. 2(d) and 2(e) show the results when using the PCA compression technique for spinning images. Fig 2(d) shows the amount of memory required. The PCA map is composed of the matrix $V \in C^{KxM}$, which contains the K main eigenvectors and the projections of the training images $\vec{p}_j \in C^{Kx1};\ j = 1 \ldots N$.

Although the training images have been artificially rotated to add the orientation information in the database, it is not necessary to store the projections of all the rotated images but only the projections of one image per training position. This is due to the fact that a rotation in the image results in the change of angle of the PCA coefficients of this image, but not in the module. So, if we have the coefficients for one representative viewpoint, the coefficients of the rotated images can automatically be generated through a rotation in the complex plane. So the module of the projections can be used to compute the position of the robot, and the phase is useful to know the orientation. Anyway, as we can see on fig. 2, PCA is a computationally more expensive process comparing to Fourier Transform.

4 LOCALIZATION AND ORIENTATION RECOVERING

To test the validity of the previous maps, the robot has captured several test images in some half-way points among those stored in the map. We have captured two sets of test images, the first one, at the same time we took the training set and the second one a few days later, in different times of the day (with varying illumination conditions) and with changes in the position of some objects. The objective is to compute the position and orientation of the robot when it took the test images, just using the visual information in the maps.

4.1 PCA-based Techniques

The PCA map is made up of the matrix $V \in C^{KxM}$, which contains the K main eigenvectors and the projections of the training images $\vec{p}^{\,j} \in C^{Kx1}$ (one per position, as explained in the previous section), that have been decomposed in two vectors, $\vec{p}_m^{\,j} \in R^{Kx1}$ containing the modules of the components of the projections and $\vec{p}_{ph}^{\,j} \in R^{Kx1}$ containing the phases.

To compute the location where the robot took each test image, we have to project the test image $\vec{x}^i \in R^{Mx1}$ onto the eigenspace, $\vec{p}^{\,i} = V^T \cdot \vec{x}^i \in C^{Kx1}$. Then, we compute the vector of modules $\vec{p}_m^{\,i} \in R^{Kx1}$ and compare it with all the vectors $\vec{p}_m^{\,j}$ stored in the map. The criterion used is the Euclidean distance. The corresponding position of the robot is extracted as the best matching. Once this position is known, we make use of the phases vector $\vec{p}_{ph}^{\,i}$ to compute the orientation of the robot.

Table 1 shows the results we have obtained when computing the position and the orientation when the training set is taken over a 30x30 cm grid and table 2 shows the same results for an 40x40 cm grid, depending on the number of eigenvectors (*K*) we retain. In these tables, p_1 is the probability that the best match is the actually nearest image (geometrically), p_2 is the probability that the best match is one of the two actually nearest images, and p_3 is the probability it is one of the three nearest images. At last, e_θ is the average error in the orientation estimation.

Table 1: Accuracy in the estimation of the position and orientation with PCA methods. 30x30 cm grid training set.

K	p_1	p_2	p_3	e_θ
112	81.8	96.7	97.1	3.21º
224	82.6	96.8	98.8	2.89º
448	87.2	96.8	98.8	5.29º

Table 2: Accuracy in the estimation of the position and orientation with PCA methods. 40x40 cm grid training set.

K	p_1	p_2	p_3	e_θ
112	96.3	97.1	98.4	8.07º
224	96.7	97.5	99.3	5.95º
448	97.9	98.8	99.5	4.79º

Fig. 3(a) shows the time taken up by this method to compute the position and orientation, depending of the number of images stored in the map, and the number of eigenvectors.

4.2 Fourier-based Techniques

To compute the position and orientation of the robot for each test image, we compute the Fourier Transform (with the two methods described in the previous section) and then, we compute the Euclidean distance of the power spectrum of the test image with the spectra stored in the map. The best match is taken as the current position of the robot.

On the other hand, the orientation is computed with eq. (6), when we work with 2D Discrete Fourier Transform (assuming $y_0 = 0$) and with the expression (7), when we work with the Fourier Signature. We obtain a different angle per row so we have to compute the average angle. Tables 3 and 4 show the accuracy we obtain in position and orientation estimation with the Fourier methods.

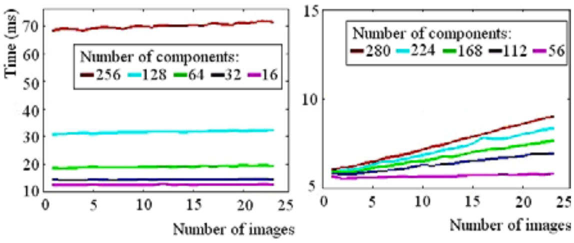
Table 3: Accuracy in the estimation of the position and orientation. Fourier methods. 30x30 cm grid training set.

K	p_1	p_2	p_3	e_θ
2x56	76.5	94.6	96.3	3.13º
4x56	78.5	95.5	97.5	3.04º
8x56	83.9	97.4	98.4	2.91º
16x56	85.5	98.4	98.4	2.89º

Table 4: Accuracy in the estimation of the position and orientation. Fourier methods. 40x40 cm grid training set.

K	p_1	p_2	p_3	e_θ
2x56	94.2	96.7	97.2	4.36
4x56	94.6	97.5	98.3	4.35
8x56	96.7	98.8	100	4.40
16x56	97.5	99.6	100	4.37

Fig. 3(b) shows the time elapsed since the robot captures the omnidirectional image until the position and orientation of the robot are obtained, depending on the number of images stored in the map and the number of Fourier components we retain. This approach clearly outperforms the PCA methods in accuracy and time consumption.



(a) (b)

Figure 3: Time consumption to compute position and orientation with (a) PCA and (b) Fourier methods.

5 CONCLUSIONS

In this work, we have exposed the principles of the creation of a dense map of a real environment, using omnidirectional images and appearance-based methods. We have presented three different methods to compress the information in the map. The mathematical properties of these methods together with the rich information the omnidirectional images pick up from the environment permit the robot to compute its position and orientation into the map.

The Fourier Transform method (both the 2D Discrete Fourier Transform and the Fourier Signature) has proved to be a good method to compress the information comparing to PCA regarding both the time and the amount of memory, and the accuracy in position and orientation estimation. Another important property is that the Fourier Transform is an inherently incremental method. When we work with PCA, we need to have all the training images available before carrying out the compression so this method cannot be applied to tasks that require an incremental process (e.g. a SLAM algorithm where the information of the new location must be added to the map while the robot is moving around the environment). The Fourier Transform does not present this disadvantage because the compression of each image is carried out independently. These properties make it applicable to future tasks where the robots have to add new information to the map and localize themselves in real time.

This work opens the door to new applications of the appearance-based methods in mobile robotics. As we have shown, the main problem these methods present is the high requirements of memory and computation time to build the database and make the necessary comparisons to compute the position and orientation of the robot. Once we have studied in deep some methods to compress the information and separate the calculation of position and orientation, the next step should be to test their robustness to changes in illumination and in the position of some objects in the scene. Also, their robustness and simplicity make them applicable to the creation of more sophisticated maps, where we have no information of the position the robot had when he took the training images.

ACKNOWLEDGEMENTS

This work has been supported by the Spanish government through the project DPI2007-61197. ‘Sistemas de percepción visual móvil y cooperativo como soporte para la realización de tareas con redes de robots’.

REFERENCES

Artac, M.; Jogan, M. & Leonardis, A., 2002. Mobile Robot Localization Using an Incremental Eigenspace Model, In *Proceedings of IEEE International Conference on Robotics and Automation*, Washington, USA, pp. 1205-1030, IEEE.

Booij, O., Terwijn, B., Zivkovic, Z., Kröse, B., 2007. Navigation using an Appearance Based Topological Map. In *IEEE International Conference on Robotics and Automation*, pp. 3297-3932 IEEE Press, New York.

Jogan, M., Leonardis, A., 2000. Robust Localization Using Eigenspace of Spinning-Images. In *Proc. IEEE Workshop on Omnidirectional Vision*, Hilton Head Island, USA, pp. 37-44, IEEE.

Kirby, M., 2000. *Geometric data analysis. An empirical approach to dimensionality reduction and the study of patterns*, Wiley Interscience.

Kröse, B., Bunschoten, R., Hagen, S., Terwijn, B. Vlassis, N., 2004. Household robots: Look and learn. In *IEEE Robotics & Automation magazine*. Vol. 11, No. 4, pp. 45-52.

Menegatti, E.; Maeda, T. Ishiguro, H., 2004. Image-based memory for robot navigation using properties of omnidirectional images. In *Robotics and Autonomous Systems*. Vol. 47, No. 4, pp. 251-276.

Payá, L., Reinoso, O., Gil, A., Sogorb. J., 2008. Multi-robot route following using omnidirectional vision and appearance-based representation of the environment. In *Lecture Notes in Artificial Intelligence. Hybrid Artificial Intelligence Systems*, Vol. 5271, pp. 680-687 Springer.

Rossi, F., Ranganathan, A., Dellaert, F., Menegatti, E., 2008. Toward topological localization with spherical Fourier transform and uncalibrated camera. In *Proc. Int. Conf. on Simulation, Modeling and Programming for Autonomous Robots*. Venice (Italy), pp. 319-330.

Thrun, S., 2003. Robotic Mapping: A Survey, In *Exploring Artificial Intelligence in the New Milenium*, pp. 1-35, Morgan Kaufmann Publishers, San Francisco, USA.

Ueonara, M., Kanade, T, 1998. Optimal approximation of uniformly rotated images: relationship between Karhunen-Loeve expansion and Discrete Cosine Transform. In *IEEE Transactions on Image Processing*. Vol. 7, No. 1, pp. 116-119.

Vasudevan, S., Gächter, S., Nguyen, V., Siegwart, R., 2007. Cognitive maps for mobile robots – an object based approach. In *Robotics and Autonomous Systems*. Vol. 55, No. 1, pp. 359-371.

ROBUST CONTROL FOR AN ARTIFICIAL MUSCLES ROBOT ARM

S. Boudoua, M. Chettouh and M. Hamerlain
The Advanced Technologies Development Centre (CDTA), Baba Hassen, Algiers, Algeria
{sboudoua, mhamerlain}@cdta.dz, chettouh@yahoo.com

Keywords: Neural Network, Reinforcement Learning, Variable Structure System, Pneumatic Artificial Muscle, Manipulator Robot Arm.

Abstract: We are concerned with the control of a 3-DOF robot arm actuated by pneumatic rubber muscles. The system is highly non-linear and somehow difficult to model therefore resorting to robust control is required.The work in this paper addresses this problem by presenting two types of robust control. One uses neural network control, which has powerful learning capability, adaptation and tackles nonlinearities; in our work the learning performed on-line is based on a binary reinforcement signal without knowing the nonlinearities appearing in the system and no preliminary off-line learning phase is required. The other control law is a Classical variable structure which is robust against parameters variations and external disturbances. Experimental results together with a comparative study are presented and discussed.

1 INTRODUCTION

For most robotic applications, the common actuator technology is electric with very limited use of hydraulics or pneumatics but electrical systems suffer from relatively low power/weight ratio, especially in the case of human-friendly robot or human coexisting and collaborative systems, such as in medical and welfare fields. Therefore, sharing the robot working space with its environment is problematic. Conversely, the human arm is not very accurate, but its lightness and joint flexibility due to the human musculature give it a natural capability for working in contact. A novel pneumatic artificial muscle (PAM) actuator (Caldwell et al., 1993; Bowler et al.), which has achieved increased popularity to provide these advantages, has been regarded during the recent decades as an interesting alternative to hydraulic and electric actuators and applied to construct a therapy robot where high level of safety for humans is required. However, the complex nonlinear dynamics of the PAM manipulator makes it a challenging and appealing system for modeling and control design. As a result, a considerable amount of research has been devoted to the development of various position control systems for the PAM manipulator. The fine control performance could be obtained by using some control strategies such as sliding mode control (Cai and Yamaura, 1997; Carbonell et al., 2001; Tondu and Lopex, 2000; Hamerlain, 1995), adaptive control and so on. However, these systems were based on the assumption that the process to be controlled should be linear and past of the research results are just considered with step reference input. Furthermore, intelligent control techniques have emerged to overcome some deficiencies in conventional control methods in dealing with complex real-world systems in more recent years. Fuzzy controllers (Balasubramanian and Rattan, 2003) have been successfully implemented for many linear and nonlinear processes. However, there were obviously steady-state error, and it also was very hard to implement in practice because of the difficulty in constructing the control rule’s bases. In addition, neural network control has been successfully used in many commercial and industrial applications in recent years. An adaptive controller based on the neural network was applied to the artificial hand, which was composed of the PAM (Folgheraiter et al., 2003). Nonlinear PID control to improve the control performance of 2 axes pneumatic artificial muscle manipulator using neural network (NN) has been proposed by Tu Diep (Thanh and Kwan, 2006).

The work in this paper addresses this problem by showing the ability of the NN to learn unmodeled nonlinear dynamics through reinforcement learning.

In this paper, we will explore a new type of reinforcement learning algorithm (Kim and Lewis, 1997), in which the learning signal is merely a binary "+1" or "-1", from a critic rather than an instructive correction signal. Compared with existing NN learning methods, where learning is performed in a trial-and-error manner, the NN weights in our scheme are tuned on-line, with no off-line learning phase required, in such a fashion that closed-loop performance is guaranteed. The experiments were carried out in practical 3 axes PAM manipulator and the effectiveness of the proposed control algorithm was demonstrated and compared with sliding mode control, which suggests its superior performance and disturbance rejection.

2 ACTUATOR AND MECHANICAL STRUCTURE

The three degrees of freedom (DOF) of the robot manipulator prototype illustrated in figure 1 are considered. It consists of a base joint, a shoulder joint and an elbow joint, all of which are revolute.

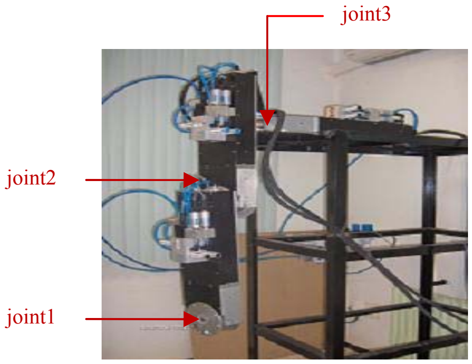


Figure 1: Experimental robot arm.

Since the pneumatic artificial rubber muscles (PAM) are contractile devices, in order to have a bidirectionally actuated revolute joint, two PAM have to be used in what is generally called an antagonistic setup. This is illustrated in figure 2.

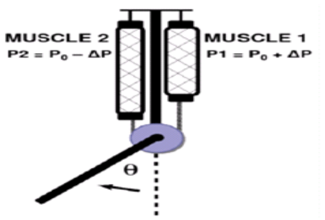


Figure 2: Working principle of a joint.

The muscles in this application were designed to function as biceps. As the internal air pressure increases, the actuator expands in its radial direction and contracts its length. The (PAM) selected as the actuator for this robot arm is the MAS-40 fluidic muscle manufactured by FESTO (Pomiers, 2003).

3 DIRECT REINFORCEMENT ADAPTIVE LEARNING NEURAL NETWORK CONTROL

3.1 Neural Networks

Here we employ a simple “two-layer” feedforward neural network (NN) to approximate a general smooth non linear function on a compact set R^n (Sadegh, 1993). According to the NN approximation property:

$$f(x) = W^T\sigma\left(V^T x\right) + \varepsilon(x) \qquad (1)$$

where x= [*1 x1 x2* ... *xn*] is the input to NN, σ( ) is an active function, W and V are defined as the collection of respectively, NN weights for output and hidden layer and *ε(x)* is the NN approximation error.

The NN in the remainder of the paper is considered with the first layer weight V fixed. This makes the NN linear in the parameters. Selecting a constant V result in the NN output $y = W^T\sigma(\chi)$.

There exist constant weights W so that the nonlinear function to be approximated can be represented as:

$$f(x) = W^T\sigma(\chi) + \varepsilon(x) \qquad (2)$$

with $\|\varepsilon(x)\| < \varepsilon_N$; ε_N is a known value.

Then, the functional estimate can be given by $\hat{f}(x) = \hat{W}^T\sigma(\chi)$ Where $\hat{W}$ is provided by a certain tuning algorithm. In particular in Barron’s paper (Barron, 1993) it was shown that neural networks can serve as universal approximators for continuous functions more efficiently than traditional functional approximators, even though there exists a fundamental lower bound on the functional reconstruction error of order $(\frac{1}{N_k})^{\frac{2}{n}}$ where N_k is the number of neurons in the hidden layer.

3.2 Controller Design

In this paper, the detailed system dynamics and the nonlinearities in the controlled system are assumed to be unknown. It is only supposed that the system belongs to a general class having a canonical structure:

$$\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= x_3 \\
&\vdots \\
\dot{x}_n &= g(x) + d(t) + u(t) \\
y &= x_1
\end{aligned} \qquad (3)$$

with state $X = [x_1\ x_2 \dots x_n]^T$,$u(t)$ is the control input to the plant, *d(t)* the unknown disturbance with a known upper bound *bd*, *g(x)* an unknown smooth function and output y.

Define the reference signal as $X_d = [x_d\, \dot{x}_d \dots\dots \overset{(n-1)}{x_d}]^T$ A standard use in robotics is the filtered tracking error $r(t) = \Lambda^T e(t)$ Where $\Lambda^T = [\lambda_1\, \lambda_2 \dots \lambda_n]$ is an appropriately chosen coefficient vector such that $s^{n-1} + \lambda_{n-1} s^{n-2} + \cdots + \lambda_1$ is Hurwitz ($e(t) \to 0$ exponentially as $r(t) \to 0$).

The tracking error vector is defined as $e(t) = X_d - X$. The full filtered tracking error $r(t)$ is not allowed to be used for tuning the action generating NN weights. Only a reduced reinforcement signal *R* is allowed.

$$R = \text{sgn}(r);\ \text{sgn}(x) = \begin{cases} +1 & if\ x \geq 0 \\ -1 & otherwise \end{cases}$$

The time derivative of the measured performance signal can be written as:

$$\dot{r} = g(X, X_d) + u(t) + d(t) \qquad (4)$$

where $g(X, X_d)$ is a fairly complex nonlinear function of X and X_d .The control input $u(t)$ used to control the plant is given by (Kim and Lewis, 1997):

$$u(t) = -K_v r - \hat{g}(X, X_d) + v(t) \qquad (5)$$

where $\hat{g}(X, X_d)$ is provided by the NN. The performance measurement gain matrix is $K_v = K^T{}_v$ and $v(t)$ is a robustifying vector that will be determined later to offset the NN functional reconstruction error $\varepsilon(x)$ and disturbances $d(t)$.

From (4), the time derivative of the performance measure signal $r(t)$ can be rewritten as:

$$\dot{r} = -K_v r + \tilde{g}(X, X_d) + d(t) + v(t) \qquad (6)$$

where $\tilde{g}(X, X_d) = g(X, X_d) - \hat{g}(X, X_d)$

The continuous nonlinear function $g(X, X_d)$ can be represented by a NN with some constant "ideal" weight W and some sufficient number of input basis function $\sigma(\ )$ as:

$$g(X, X_d) = W^T \sigma(\chi) + \varepsilon(x) \qquad (7)$$

with $\|\varepsilon(x)\| < \varepsilon_N$.

We assume that the ideal weight W is bounded by known positive values (Lewis et al., 1995; Kosmatopoulos, 1990) so that $\|W\|_F \leq W_M$ where W_M is a known value.

Let the NN functional estimate for the continuous nonlinear function $g(X, X_d)$ be given by:

$$\hat{g}(X, X_d) = \hat{W}^T \sigma(\chi) \qquad (8)$$

where the current value $\hat{W}$ is provided by the weight tuning algorithm. From (3) and (4) we have the following performance measure:

$$\dot{r} = -K_v r + \tilde{W}\sigma(\chi) + \varepsilon(x) + d(t) + v(t) \qquad (9)$$

with the weight estimation error $\tilde{W} = W - \hat{W}$.

The robustifying term is given by (Kim and Lewis, 1997):

$$v(t) = -K_z \frac{R}{\|R\|} \qquad (10)$$

with $K_z \geq bd$ And reinforcement learning rule for tuning the action generating NN weights is given by (Kim and Lewis, 1997):

$$\dot{\hat{W}} = F\sigma(\chi)R^T - kF\hat{W} \qquad (11)$$

with $F = F^T$ for the learning rate and $k > 0$ for the speed of convergence. Then the errors r and $\tilde{W}$ are Uniformly Ultimately Bounded (UUB) (Kim and Lewis, 1997). Moreover, the performance measure $r(t)$ can be made arbitrarily small by increasing the fixed control gain K_v .

Proof (Kim and Lewis, 1997). Define the Lyapunov function candidate:

$$L = \sum_{i=1}^{m} |r_i| + \frac{1}{2} tr(\tilde{W}^T F^{-1} \tilde{W})$$

Differentiation yields:

$$\dot{L} = \sum_{i=1}^{m} \text{sgn}(r)^T \dot{r} + tr(\tilde{W}^T F^{-1} \dot{\tilde{W}})$$

Substituting now from the error system (9) and using (11) gives:

$$\dot{L} \le \sum_{i=1}^{m} - R^T K_v r + R^T \varepsilon(x) + tr(\tilde{W}^T \hat{W})$$

Using the inequality:

$$tr(\tilde{W}^T \hat{W}) = tr\left\{\tilde{W}^T (W - \tilde{W})\right\} \le \|\tilde{W}\|_F (W_M - \|\tilde{W}\|_F)$$

and $\|\text{sgn}(r)^T\| \le \sqrt{n}$ results in:

$$\dot{L} \le \sqrt{n}\, \lambda_{\min}(K_v)\|r\| - k(\|\tilde{W}\|_F - \frac{W_M}{2})^2 + k\frac{W_M^2}{4} + \sqrt{n}\, \varepsilon_N$$

which is guaranteed negative as long as either:

$$\|r\| \ge \frac{k\frac{W_M^2}{4} + \sqrt{n}\, \varepsilon_N}{\sqrt{n}\, \lambda_{\min}(K_v)}$$

Or

$$\|\tilde{W}\|_F \ge \frac{W_M}{2} + \sqrt{\frac{W_M^2}{4} + \frac{\sqrt{n}\, \varepsilon_N}{k}}$$

According to a standard Lyapunov theory extension (Lewis et al., 1993; Narendra and Annaswamy, 1987), this demonstrates the *UUB* of both $\|r\|$ and $\|\tilde{W}\|_F$.

4 VARIABLE STRUCTURE CONTROL

Sliding mode control (SMC) is a type of variable structure control where the dynamics of a nonlinear system is changed by switching discontinuously on time on a predetermined sliding surface with a high speed, nonlinear feedback (Young et al., 1999). Actually, sliding mode controller design has two steps: the first step involves obtaining a sliding surface for desired stable dynamics and the second step is about providing the control law to reach this sliding surface. The system trajectories are sensitive to parameter variations and disturbances during the reaching mode whereas they are insensitive in the sliding mode (Hung et al., 1993). Although CVS (Classical Variable Structure) control is robust against modelling errors, it however requires an approximate model. Knowledge of the assumed model parameter variation bounds is also required.

The identification of each joint dynamics is based on the estimation of coefficients of a presumed linear model. This is achieved by fitting the best linear curve to the input-output data using an ARX model (Autoregressive with exogenous input) in MATLAB. Joint dynamic parameters are identified using various step input signals. The measured response for the joint angle variation θ (radians) corresponding to various step of the pressure between the two muscles is shown in (figure 3).

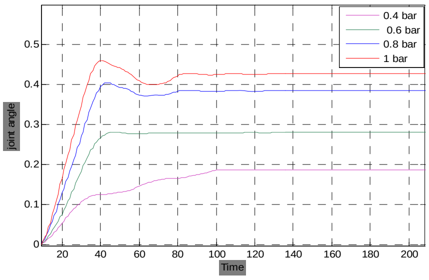


Figure 3: Step response of robot arm (joint 1).

In a linear approximation, the decoupled model for the system dynamics is given in the following form:

$$\ddot{q} + A_1.\dot{q} + A_2.q = B.u \tag{12}$$

Where $q = [q_1, q_2, q_3]^T$ is the displacement vector A_1, A_2 and B are the estimated gain matrices of velocity position and control. These for a decoupled system are:

$$A_1 = diag\ [0.16 \quad 0.17 \quad 1.25]$$
$$A_2 = diag\ [1.19e-2 \quad 0.94e-2 \quad 0.55]$$
$$B = diag\ [0.57e-2 \quad 0.27e-2 \quad 2.23e-3]$$

The sliding mode occurs on a switching surface $S(x) = 0$, which forces the original system to behave as a linear time invariant system, which can be designed to be stable. The switching surfaces are chosen as:

$$S_i(e_i, \dot{e}_i) = \lambda_i e_i + \dot{e}_i \tag{13}$$

$1 \le i \le 3$

Where $\lambda_i \succ 0$, $e_i = q_i - q_{id}$ with q_{id} is the desired position. For ideal sliding to occur, the invariance conditions $S_i(e_i,\dot{e}_i)=0 \; and \; \dot{S}_i(e_i,\dot{e}_i)=0$ must be satisfied. This yields the equivalent control:

$$U_{ieq} = b_i^{-1}[(a_{i1}-\lambda_i)\dot{e}_i + a_{i2}e_i + a_{i2}q_{id} + a_{i1}\dot{q}_{id} + \ddot{q}_{id}] \qquad (14)$$

Now, due to modelling errors, the estimated equivalent control is given by

$$\overset{*}{U}_{ieq} = \overset{*}{b}{}_i^{-1}[(\overset{*}{a}_{i1}-\lambda_i)\dot{e}_i + \overset{*}{a}_{i2}e_i + \overset{*}{a}_{i2}q_{id} + \overset{*}{a}_{i1}\dot{q}_{id} + \ddot{q}_{id}] \qquad (15)$$

where $\overset{*}{b}_i, \overset{*}{a}_{ij}$ are estimated mean parameters.

The control U_i is then fixed as $U_i = \overset{*}{U}_{ieq} + \Delta U_i$ while ΔU_i is the high frequency component which ensures the sliding mode and consequently the system insensitivity to parameter variations ,errors modelling and perturbations.

The control U_i is discontinuous across the switching surfaces $S_i(e_i,\dot{e}_i) = 0$

$$U_i = \begin{cases} U_i^+ = U_{ieq}^* + \Delta U_i^+ & \text{if } \mathrm{S_i}(e_i,\dot{e}_i) > 0 \\ U_i^- = U_{ieq}^* + \Delta U_i^- & \text{if } \mathrm{S_i}(e_i,\dot{e}_i) < 0 \end{cases}$$

The discontinuous component can take several forms in literature the form retained is established by Harashima et al. (Harashima et al., 1986) as:

$$\Delta U_i = (\alpha_i|e_i| + \beta_i|\dot{e}_i| + \gamma_i).\mathrm{sgn}(S_i) \qquad (16)$$

5 EXPERIMENTAL RESULTS

Experimental results of both DRAL and CVS control laws applied to a 3-DOF robot arm driven by pneumatic artificial muscles are presented.

5.1 Tracking Trajectory

We present a simultaneous control of all three robot axes for tracking a sinusoidal reference trajectory; joint coupling is significant.

Number of hidden neurons is 20 and activation functions are sigmoid. Experimental parameters are as follows:

$$K_{vi} = \begin{pmatrix} 0.5 \\ 2 \\ 0.5 \end{pmatrix}; \lambda = \begin{pmatrix} 0.02 \\ 0.08 \\ 0.12 \end{pmatrix}; F_i = \begin{pmatrix} 0.2 \\ 0.3 \\ 0.1 \end{pmatrix}; K = \begin{pmatrix} 0.5 \\ 0.5 \\ 0.5 \end{pmatrix}; K_{zi} = \begin{pmatrix} 0.1 \\ 0.1 \\ 0.1 \end{pmatrix}.$$

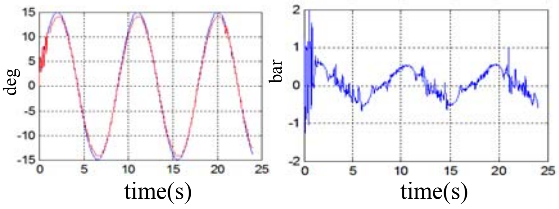


Figure 4: Position and signal of control of joint 1.

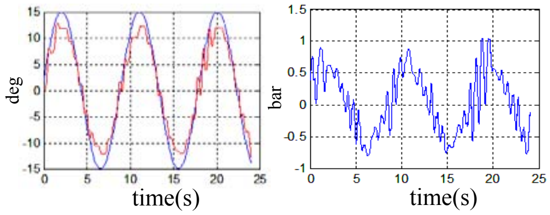


Figure 5: Position and signal of control of joint 2.

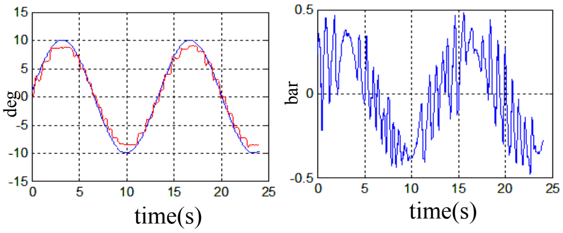


Figure 6: Position and signal of control of joint 3.

The performance of the DRAL controller shows that the trajectory following ability is fairly good. Due to its position in the robot arm the second joint is more difficult to control because of interactions between axes (see Figure 1), moreover, the tracking errors converge to small values as expected from the stability analysis. Though robot non linearity and system dynamics are completely unknown to the DRAL, the algorithm has good properties to cancel the nonlinearities in the robot system, it can also be improved by supplying NN with more input signals (in this work we have considered that NN have to approximate unknown second order dynamics).

5.2 Comparative Study

In order to show the ability of the DRAL to control unknown highly non linear systems our experimental results are compared with those obtained using sliding mode control. Both reference and tracking are considered.

We summarize our concluding remarks in the tables below.

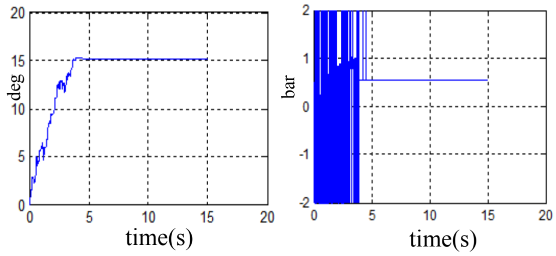


Figure 7: Position and SMC signal control joint 1.

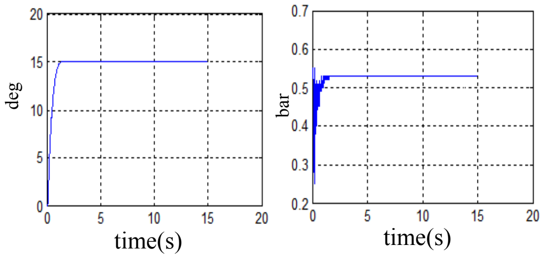


Figure 8: Position and DRAL signal control joint 1.

Table 1.

	DRAL	VSS
Response Time	1.5s	4 s
Chattering	Insignificant	Exist in the transient part
Control	Not energetic Umax=0.53bar	Energetic Umax=2bar
Static error	0.02 degree	0.26 degree

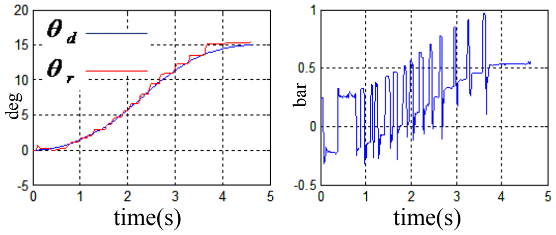


Figure 9: Position and SMC signal control joint 1.

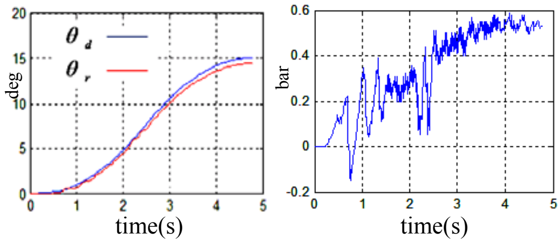


Figure 10: Position and DRAL signal control joint 1.

Table 2.

	Control	Trajectory
DRAL	Umax=0.59bar	Smooth
VSS	Energetic Umax=0.9bar	Incremental

Among the disadvantages of pneumatic artificial muscles we can underline frictions between a rubber tube and the synthetic braid which result on incremental trajectory tracking as shown with VSS control (Fig 9), conversely with DRAL we have attenuated this drawback since the trajectory following is fairly smooth (Fig 10), which proves the ability of neural network to learn unmodeled nonlinear dynamics.

6 CONCLUSIONS

Due to nonlinearities and uncertainties the exact dynamic characteristics of PAM robot manipulator are very difficult to obtain, therefore resorting to robust control is required. Neural network has powerful capability of learning, adaptation and tackling nonlinearity, the proposed neural network controller using reinforcement learning for on line identification of plant dynamics are simple to apply to any control system in order to minimize the position error without knowledge of the plant to be controlled, the algorithm does not require any off-line training or learning phase, the algorithm has proven its performances through experiments and comparative study with sliding mode control. Since the traditional SMC design is a model-based control approach, the partial knowledge of model dynamics deteriorates the control performance; on the other hand we have proven in this work that NN can approximate any unknown complicated nonlinear dynamics consequently, our future investigation will focus on implementation of hybrid control law combining these two methods.

REFERENCES

D. G. Caldwell, G. A. Medrano-Cerda, M. J. Goodwin,"Braided pneumatic actuator control of a multi-jointed manipulator," in *Proc. IEEE int. conf. Systems, man and cybernetics,* Le Touque, France 1993, pp. 423–428.

C. J. Bowler, D. G. Caldwell, G. A. Medrano-Cerda,"Pneumatic muscle actuators Musculature for an anthropomorphic robot arm," in *Proc. IEE colloquium. Actuator technology current practice and new developments*, London, pp. 8/1–8/5.

D. Cai, H. Yamaura. "A VSS control method for a manipulator driven by an McKibben artificial muscle actuator," *Electron, Commun, Japan*, vol. 80, no. 3, pp. 55-63, 1997.

P. Carbonell, ZP. Jiang, DW. Repperger. "Nonlinear control of a pneumatic muscle actuator: backstepping vs. sliding-mode," *in Proc. IEEE int. Conf. Control applications, Mexico City, Mexico 2001, pp. 167-172.*

B. Tondu, P. Lopex. "Modeling and control of McKibben artificial muscle robot actuators" *in Proc.of the IEEE .Int Conf. Control Syst* Mag 2000, vol.20, no.1, pp.15-38.

M. Hamerlain, "An anthropomorphic robot arm driven by artificial muscles using a variable structure control," *in Proc. IEEE/RSJ int Conf. Intelligent Robots and Systems*, 1995, vol.1, pp. 550-555.

V. Balasubramanian, KS. Rattan, "Feedforward control of a non-linear pneumatic muscle system using fuzzy logic," *in IEEE int. Conf. Fuzzy Systems*, 2003, vol.1, p. 272–277.

M. Folgheraiter, G. Gini, M. Perkowski, M. Pivtoraiko, "Adaptive reflex control for an artificial hand" *in Proc SYROCO 2003, symposium on robot control*, *Holliday Inn*, Wroclaw, Poland, 2003.

T. D. C. Thanh, A. K. Kwan, "Nonlinear PID control to improve the control performance of 2 axes pneumatic artificial muscle manipulator using neural network," Science Direct. Mechatronics 16, 577-587, 2006.

Y. H. Kim, F. L. Lewis, "Direct-Reinforcement-Adaptive-Learning Neural Network Control for Nonlinear Systems," *Proceedings of the American Control Conference* Albuquerque, New Mexico June 1997.

P. Pomiers."Modular robot arm based on pneumatic artificial rubber muscles (PARM)", in CLAWAR 2003, Catania, Italy, 17-19 Sept 2003.

N. Sadegh, "A perceptron network for functional identification and control of nonlinear systems," *IEEE Trans. Neural Networks,* vol.4, no. 6, pp. 982-988, 1993.

A. R. Barron, "Universal approximation bounds for superposition of a sigmoidal function," *IEEE Trans. Inform. Theory*, vol.39, no. 3, pp. 930-945, 1993.

F. L. Lewis, A. Yesildirek, and K. Liu, "Neural net robot controller with guaranteed tracking performance," *IEEE Trans. Neural Networks*, vol. 6, no. 3, pp. 703-715, 1995.

E. B. Kosmatopoulos, M. M. Polycarpou, M. A. Christodoulou, P. A. loannou, "High-order neural network structures for identification of dynamical systems," *IEEE Trans. Neural Networks*, vol. 6, no. 2, pp. 422-431, 1990.

F. L. Lewis, C. T. Abdallah, and D. M. Dawson, Control of Robot Manipulators. MacMillan, New York, 1993.

K. S. Narendra and A. M. Annaswamy, "A new adaptive law for robust adaptation without persistent excitation," *IEEE Trans. Automat. Control*, vol. 32, no.2, pp. 134-145, 1987.

K. D. Young, V. I. Utkin, and U. Ozguner, "A Control Engineer's Guide to Sliding Mode Control," *IEEE Trans. Control Systems Technology*, vol. 7, no. 3, pp. 328-342, May 1999.

J. Y. Hung, W. Gao, and J. C. Hung, "Variable structure control:A survey" *IEEE Trans. Industrial Electronics*, vol. 40, no.1, pp. 2-22, 1993.

F. Harashima, H. Hashimoto, K. Maruyama, "Practical robust control of robot arm using variable structure system". *in Proc.of the IEEE .Int Conf.on Robotics and Automation San Fransisco 1986*, 532-538.

APPLYING A SOFTWARE FRAMEWORK FOR SUPERVISORY CONTROL OF A PLC-BASED DISCRETE EVENT SYSTEM

B. Curto, V. Moreno
Department of Informática y Automática, Salamanca University, Plaza de los Caídos, Salamanca, Spain
{bcurto, vmoreno}@usal.es

C. Fernández-Caramés, R. Alves, A. Chehayeb
Department of Informática y Automática, Salamanca University, Plaza de los Caídos, Salamanca, Spain
{carlosfc, ralves}@usal.es

Keywords: Supervisory Control, Automated Manufacturing Systems, Software Framework, Discrete Event System.

Abstract: In this paper we propose a software framework where the main aim is to make easier the implementation of supervisory control. The main idea is that functionalities must be offered with no effects on the robustness of the system. We prove that our methodology has a solid base, so the approach can be applied to any kind of industrial process unit. In this way, a real application of a developed framework is presented. The implementation is done over well-known devices (such as PLC with a conveyor belt) so the technical feasibility of the procedure is guaranteed.

1 INTRODUCTION

Due to the market evolution, manufacturing industry suffers some pressures in order to reduce product prices, to increase the model complexity and to support a model proliferation (Beck, 2000). To achieve these goals, it is necessary to accomplish shorter production cycles and lower manufacturing costs. More flexible and intelligent workcells are needed in order to find a competitive manufacturing process. Thus, as a part of a Computer Integrated Manufacturing (CIM) goal, Flexible Manufacturing Systems (FMS) offer the best promise of reducing costs and increasing flexibility.

Current robotic workcell systems can generally be classified as large, monolithic and centralized systems. The main reason comes from the fact that, in a typical industrial workcell, a Programmable Logic Controller (PLC) acts as an operation sequence controller of the cell. Several problems arise with these systems: limited functionality and flexibility, low levels of intelligence, and so on. As a solution we will propose the use of a distributed architecture design.

In (Curto, 2001), a proposal of a software architecture is realized where the development of distributed systems is taken into account. In a natural way, the development of a framework is the next step where some wrong situations appear when the provided services at the system becomes blocked due to an erroneous sequence of service invocations. Thus, it is necessary to achieve that these blocked situations will be avoided.

We propose the use of the Supervisory Control Theory (SCT) (Ramadge, 1987) that is proposed initially by P. J. Ramadge and W. M. Wonham in the late 80s. This theory has received a special attention from the academic environment in such a way it has reached a great evolution. Nevertheless, actually just a few applications at industrial environment can be found that applies the SCT due to mainly a complete model of the system (with a sound mathematical load) and a corresponding implementation is needed.

Although some works exist where the real implementation of controller is obtained using the SCT, they are not too much. Some of the most representative are (Chandra, 2000) and (Mušić, 2002). The first one describes the way to design a control system for assembling line whereas the second one is focused on the implementation of a concrete task on a PLC. In both cases, the controller formally obtained using the SCT is restricted to reach a particular goal at a concrete task.

This approach presents two main disadvantages. First, if a modification of the task is made, it is necessary to perform the complete redesign and implementation of the controller. Second, a unique controller is not feasible when the task implies a large set of sensors and actuators. In this situation, the physical restrictions imposed by the communications and a possible overload of the controller are two questions that have to be taken into account. Consequently, we propose to make a separation between the particular device control and the task to be performed at the plant.

In (Chandra, 2000), when the restriction of the plant behaviour has to be done, two kinds of specifications can be distinguished: security specifications and progress specifications. The first ones are considered to prevent that no plant section takes undesirable actions and the second ones are concerned to reach the finalization of the task. Security specifications can affect to a unique plant element or several elements.

In this work we propose to perform the controller design for each device taking into account the service that the component provides with no consideration on the task that has to be done. The key element will be the development and use of a software framework where it will be necessary to model and implement a set of specifications.

The rest of the paper is organized as follows: In section 2 we review the Supervisory Control Theory. Next, the main relevant topics of the proposed architecture and the resulting framework will be presented. In section 4, in order to prove the validity of our proposal we will present a case of study where a real industrial element will be considered. Finally the main conclusions are presented.

2 MODELING DISCRETE EVENT SYSTEMS

The SCT of Discrete Event Systems is based on the use of automata and formal language models. Under these models the main interest is on the order in which the different events occur. In this way, a plant (Phoha, 2004) is assumed to be the generator of these events. The behaviour of the plant model is described by event trajectories over the finite event alphabet Σ. These event trajectories can be thought of as strings over Σ, so $L \subseteq \Sigma^*$ represents the set of those event strings that describe the behaviour of the plant.

The SCT restricts the behaviour of a plant *G* by disabling temporarily certain events that can be created by Σ, so the goal is that the plant cannot create undesired or illegal event chains in *L(G)*. In the following, a few basic definitions will be posed.

The plant *G* will modelled by the deterministic finite state automaton (DFSA)

$$G = (X, \Sigma_G, f_G, \Gamma_G, x_0, X_m)$$

where,

- X is the set of states;
- Σ_G is the finite set of events over G;
- $f_G : X \times \Sigma_G \to X$ is the state transition function;
- $\Gamma_G : X \to 2^{\Sigma_G}$ is the active event function;
- $x_0 \in X$ is the initial state;
- $X_m \subseteq X$ is the set of marked states, which represent the completion of a certain task or a set of tasks.

Σ_G^* is used to denote the set of all finite length strings over Σ_G including the empty string ε. f_G can be extended from $X \times \Sigma_G$ to the $X \times \Sigma_G^*$ domain by means of recursion: $f_G(x, \varepsilon) = x$, $f_G(x, se) = f_G(f_G(x, s), e)$ for $x \in X$, $e \in \Sigma_G$ and $s \in \Sigma_G^*$.

The SCT distinguish between the controllable set (Σ_c), which can be disabled, and the non-controllable set (Σ_{nc}) which cannot be disabled. The following relations are fulfilled: $\Sigma = \Sigma_c \cup \Sigma_{nc}$ and $\Sigma_c \cap \Sigma_{nc} = \varnothing$.

In order to restrict the behaviour of the plant *G*, one or more specifications can be defined, that are usually modelled by a DFSA

$$H = (Y, \Sigma_H, f_H, \Gamma_H, y_0, Y_m)$$

A restriction limits the behaviour of the plant by means of one of the two composition operations defined by the SCT. Informally, it can be said that a composition operation allows two DFSA to run synchronously, which means that certain events will only be created if both DFSA are capable of doing so. In this work we will consider (synchronous) parallel composition of *G* and *H*, defined as:

$$G \,||\, H = Ac(X \times Y, \Sigma_G \cup \Sigma_H, f_{G||H}, \Gamma_{G||H}, (x_0, y_0), X_m \times Y_m)$$

where

$$f_{G\|H}((x,y),e)=\begin{cases}(f_G(x,e),f_H(y,e)) & \text{if } e\in\Gamma_G(x)\cap\Gamma_H(y)\\ (f_G(x,e),y) & \text{if } e\in\Gamma_G(x)\,\text{and}e\notin\Sigma_H\\ (x,f_H(y,e)) & \text{if } e\notin\Sigma_G\,\text{and}e\in\Gamma_H(y)\\ \text{notdefined} & \text{othercase}\end{cases}$$

$$\Gamma_{G\|H}(x,y)=[\Gamma_G(x)\cap\Gamma_H(y)]\cup[\Gamma_G(x)-\Sigma_H]\cup[\Gamma_H(y)-\Sigma_G]$$

Ac() denotes the accessible automaton part, excluding thus the states which cannot be accessed from x_0 state.

3 OVERVIEW OF DEVELOPED FRAMEWORK

Traditionally, when changes in the production process occur, the PLC that control the process need to be reprogrammed as well. In order to get highly flexible manufacturing systems, we propose a framework with a core element: a service repository. In this proposal, the services can be invoked dynamically based on the production needs. Therefore, if the production process is altered, then the sequence of invocation of the services will be modified. In this way, the main advantage is that the software running on the PLC does not need any modification at all, if all the required services have been taken into account.

In our framework, if an input event e_i is triggered by an external entity for a certain service, then $e_i\in\Sigma_c$; if an internal event e_i is created by the plant, then $e_i\in\Sigma_{nc}$. When certain services are invoked, the system can stop working properly. In order to avoid this, our proposal considers supervising the behaviour of the system using SCT. As stated in the previous section, once the plant *G* is modelled, its behaviour is restricted by means of one or more specifications H_k that can temporarily disable certain input events. Consequently, no undesired actions will ever take place.

Our framework (Figure 1) consists of the following components:

- *Interface with external entities*. Clients invoke the services provided by the system through this interface, and the corresponding input event will be triggered.
- *Supervision*: it is made up of a) the active event function Γ of *A*, being $A=\{G\cup H_1\cup\ldots\cup H_k\}$ and b) the synchronous composition $G\parallel H_1\parallel\ldots\parallel H_k$, where the input events can be disabled if needed.
- State transition functions, which evolve in accordance with 1) the input events that have not been disabled previously in the supervision and 2) the internal events of *G*. In the plant, if an input event causes a transition f_G, it will activate the execution of the corresponding service.
- Services provided that gather a specified functionality. Due to design criteria they are isolated from the supervision role so it is possible to perform modifications on the functionality regarding supervision consideration. This characteristic can be seen as the main advantage.

In the following, the main behaviour will be described. When an input event $e_i\in\Sigma_c$ is triggered, it is processed by the supervisor. The composition operation, taking into account Γ_A for each DFSA, decides if e_i must be disabled or not. If e_i is enabled, it is then processed by the f_A of each DFSA, including *G*. Therefore, if a transition is defined in f_G for e_i and the current state of *G*, the corresponding service will be run. Otherwise, if e_i is disabled, the transition does not happen, and the requested service will never take place. As a result, we make sure that the system behaves according to its specifications at all times.

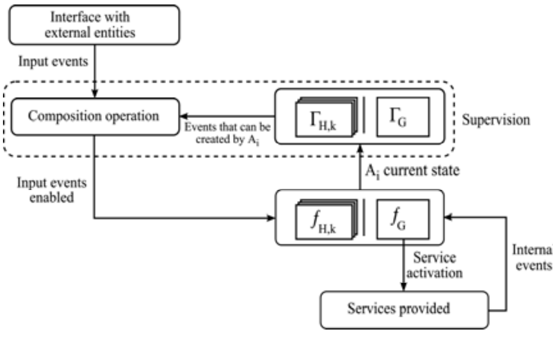


Figure 1: Global overview of the framework.

4 A CASE OF STUDY

In what follows we discuss how to use our proposed framework in a typical element of a FSM cell: a conveyor belt. We have chosen it because it is a component found in every cell instance and because the working specifications are changed frequently, given that the conveyed elements must be moved to different target positions using different velocities as well.

The plant is made up of a BOSCH conveyor belt with encoder, dynamo and position sensors. The control is carried out by the Allen Bradley SLC 500 series with the IMC 110 motion control module.

Depending on the operation mode of the IMC, the conveyor belt can be moved manually or automatically. Movements in manual mode are used for initialization and maintenance by an operator and are always executed at constant velocity. Movements in automatic mode are performed when some of the MML programs loaded in the IMC 110 memory is running, and it is possible to modify both the velocity and the acceleration of the motion. In order to work in automatic mode and know where the tray is located it is needed to perform a home operation. For security reasons, emergency stops must be considered when in automatic mode.

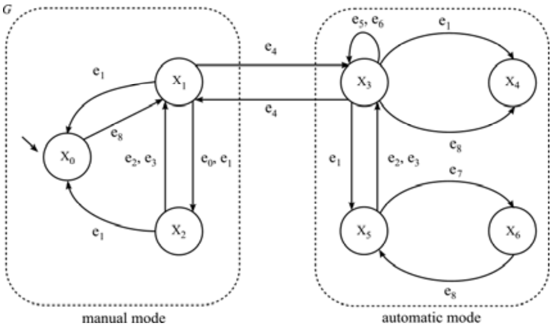


Figure 2: Plant operation modelled by a DFSA *G*.

The DFSA *G* (Figure 2) models both the manual and automatic mode of the conveyor belt. This plant model DFSA is given as the 5-tuple with x_0 as initial state. In the definition of the state set $X = \{x_0, \dots, x_7\}$ (Table 1) three factors have been considered:

- IMC operation mode (automatic or manual)
- belt state (moving or not)
- the presence of an emergency stop.

So, in this case, $X = X_m$.

Respect to Σ (Table 2), the set $\Sigma_{NC} = \{e_2, e_7, e_8\}$ is generated from the information provided by the IMC from sensors, $\Sigma_C = \{e_0, e_1, e_3, e_4, e_5, e_6\}$ corresponds with the events provided by the system.

The behaviour of the plant is controlled by two restrictions:

- R1) the movement to a target position (*e_1* event) is not allowed prior to completing a successful homing operation
- R2) the change to automatic mode (*e_4* event) is not allowed prior to completing a successful homing operation.

Table 1: *G* plant states.

State	Mode	Conveyor belt state	e-stop
x_0	Manual	Stopped	Yes
x_1	Manual	Stopped	No
x_2	Manual	Moving	No
x_3	Automatic	Stopped	No
x_4	Automatic	Moving	Yes
x_5	Automatic	Moving	No
x_7	Automatic	Moving	Yes

Let H_1 and H_2 be the two DFSA that model the restrictions R1 and R2, respectively.

According to the H_1 state transition diagram (Figure 3), the state y_1 represents the action "performing home operation". If it is completed successfully (*e_2* event), the state changes to y_2. Given that $e_1 \in \Gamma_{H1}(y_2)$, then it is not possible to perform any type of movement to any position.

Table 2: Defined events in *G* (C - controllable, NC - non controllable).

Event	Description	Type
e_0	Request for performing a home operation	*C*
e_1	Request for moving until the target position is reached	*C*
e_2	Movement completed successfully	*NC*
e_3	Request for stopping the current movement	*C*
e_4	Request for changing the IMC 110 operation mode	*C*
e_5	Request for setting the velocity of automatic movements	*C*
e_6	Request for setting the acceleration of automatic movements	*C*
e_7	Emergency stop occurs	*NC*
e_8	Emergency stop situation is finished	*NC*

Taking into account the state transition diagram (Figure 4) for H_2, it is only possible to change to automatic mode (*e_4* event) if a home operation has been completed successfully.

Diagrams H_1 and H_2 are structurally identical, and both restrictions over G could have been modelled by only one specification *H*. However, modelling several modular specifications H_i, structurally simpler than *H*, is easier to verify and understand (Cassandrass, 2007).

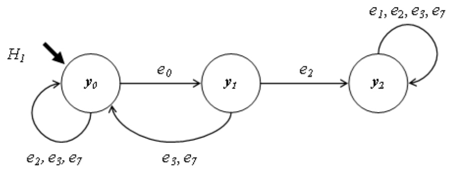


Figure 3: State transition diagram for H_1.

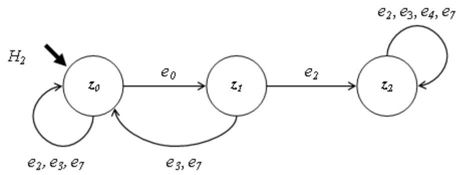


Figure 4: State transition diagram for H_2.

In order to implement the supervisor system (Figure 5), we have followed the proposed framework, taking into account G, H_1 and H_2 defined previously.

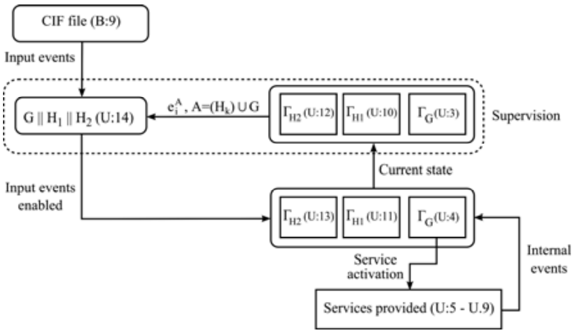


Figure 5: Implemented Supervisor System.

5 CONCLUSIONS

In this paper we propose a software framework where the main aim is to make easier the implementation of a supervisory control system over a PLC. With this framework we want to emphasize on the idea that the functionalities must be offered with no effects on the robustness of the system operation.

The system can be viewed as a well defined set of services which are requested depending on the production needs. Changes in the production process can affect the order in which the services are requested but not their implementation.

On the other hand, the flexibility obtained must have nothing to do with the robustness demanded by this kind of systems. Therefore we take into account the possibility that certain event sequences could put at risk the good working order of the system.

In order to avoid these possible incidents, we use the Supervised Control Theory. The framework we propose allows modelling one or more specifications which guarantee that the system behaves properly at all times. This is achieved by temporarily disabling the input events which could put the system at risk. Our framework also differentiates between the specifications and the functionality provided. Thus, it is possible to modify both parts independently.

To prove the main features of our framework, some implementation issues must be solved, due to the synchronous nature of PLC.

ACKNOWLEDGEMENTS

The results presented at this paper have been obtained from the tasks of research that have been partially supported by the Spanish Ministery of Science and Innovation through DPI2007-62267 proposal and by the Castilla y León Council with SA030A07 project.

REFERENCES

Beck, J., Reagin, M., Sweeny, T., Anderson, R., Garner, T. 2000. Applying a Component-Based Software Architecture to Robotic Workcell Applications. In *IEEE Transactions in Robotics and Automation*, vol. 16, pp. 207-217.

Curto, B., García, F. J., Moreno, V., González, J., Moreno, A., 2001. Experience of a CORBA based architecture for Computer Integrated Manufacturing. In *ETFA'01, 8th IEEE International Conference on Emerging Technologies and Factory Automation*, pp 765-769.

Ramadge, P. J., Wonham, W. M., 1987. Supervisory Control of a Class of Discrete Event Processes. In *SIAM J. Control and Optimization*, vol. 25, no. 1, pp. 206-230.

Chandra, V., Mohanty, S. R., Kumar, R., 2000. Automated control synthesis for an assembly line using discrete event system control theory. In *IEEE Transactions on Robotics and Automation.*

Mušić, G., Matko, D., 2002. Discrete Event Control Theory Applied to PLC Programming. In *AUTOMATIKA: Journal for Control, Measurement, Electronics, Computing and Communications*, vol. 43 no. 1-2.

Phoha, V. V., Nadgar, A., Ray, A., Phoha, S., 2004. Supervisory Control of Software Systems. In *IEEE Transactions on Computer*, vol.53, nº9.

Cassandrass, C. G., Lafortune, S., 2007. Introduction to Discrete Event Systems, Springer, 2nd edition.

EFFICIENT SIMULATION OF THE FLUID-STRUCTURE INTERFACE

Luis Gerardo de la Fraga[1], Ernesto Olguín-Díaz[2] and Fernando García-Arreguín[1]
[1]*Computer Science Department, CINVESTAV, Av. I.P.N. 2508. CP 07360, México City, México*
fraga@cs.cinvestav.mx
[2]*Robótica y Manufactura Avanzada, CINVESTAV, Carr Saltillo-Monterrey Km 13.5 CP 25900, Saltillo, México*
ernesto.olguin@cinvestav.edu.mx

Keywords: Fluid-structure simulation, Haptics.

Abstract: In this work, an alternative method to simulate fluid interactions with compliant surfaces is described. The simulation of the fluid phenomena is performed by the Smooth Particle Hidrodynamics (SPH) method. In the normal SPH method, solid interfaces are modeled with sets of static particles covering the boundaries. In the proposed alternative method, a compliant-solid interface is modeled as a polygon allowing to substitute the particles that represent a surface in the normal SPH method. Therefore, by considering less particles, a simplification on simulations is achieved, and the alternative method still describes the general behavior of the phenomena. Furthermore, a deformable object, this is, a time variant object, can be modeled as a polyhedron, with a mass-spring-dashpot system in each of its edges, and with each polygon as a compliant-solid interface. Bidirectional dependence on the alternative method for fluid simulation and the deformable model gives a new method for the simulation of compliant solids partially or fully immersed in an incompressible fluid. This new method is not intended to have a high accuracy in the numerical results but to have a perceptual high qualitative behavior and fast numerical response, to be applied in visual/haptic simulators.

1 INTRODUCTION

Understanding the dynamic of fluids has been a quest for many engineers and physicists mainly because its knowledge can help in the design process of many fluid applications such as hydroelectric dams (chuan Bai et al., 2007), water supply nets (Chen et al., 1995), aerodynamics of an aircraft (Eddy et al., 2006) or the behavior of an alien corp in blood stream of a patient (Tai et al., 2007). Some of these problems have been solved with approximate solutions of the well known Navier-Stokes equations, which describe with partial equations the behavior of a fluid under specific circumstances.

When the interaction of the fluid with a solid structure needs to be computed (for the case of the container or an object inside it) the problem becomes even more complicated due to the fact that the border conditions on the partial differential equations are dependent on time. The complexity of these partial differential equations plus the geometric complexity of volume containing the fluid lead to very complex and large amount of simultaneous equations that can only be solved approximately using numerical solutions, and finite element methods. This problem is known in the literature as the Fluid-Structure problem One possible, yet very expensive, way of compute the overall solution can be that of computing the conservation of momentum for every particle in the fluid and apply Newton's second law. The big problem with this approach is that there become too many simultaneous equations to solve three dimensional variables (interaction forces between the particles) plus the time varying border conditions of the working volume. So even when this method may lead to more accurate solutions, it is impractical for online –realtime– calculations. Fluid dynamics simulations used in haptics (kinesthetic feedback to the user) need faster numerical solutions but preserving the qualitative behavior of the fluid physical system, not only within the fluid but also with the container or bodies moving along. The big problem reduces to how the Navier-Stokes equations can be simplified to still represent, with some degree of accuracy, the fluid dynamic so that a common computer would be able to calculate the algorithm fast enough to be used in real time applications.

We propose a system where the fluid dynamics is solve with the Smoothed Particle Hydrodynamics algorithm (SPH) (Monaghan, 2005), and the border effects are solved by a compliant model composed by plane surfaces. We are using a deformable model composed by simplex meshes and a simple mechanical system in every edge on the mesh (Trejo and de la Fraga, 2005). The contribution of this work lies in the fluid-structure interface modeling that is used in the SPH as forces on the particles at the borders and as pressure in every mesh face. Equivalences between forces and pressures are given by the time varying known areas of the mesh faces.

This paper is organized as follow: in Sec. 2 the SPH method to model a fluid is described. In Sec. 3 the model of a compliant body and the way how it is integrated with the SPH to build the complete simulator are presented. In Sec. 4 the implementation of the simulator is discussed and some results presented. Finally, Sec. 5 presents the conclusions of this work.

2 FLUID DYNAMIC MODELING USING SPH

SPH use a Lagrangian approximation to the fluid mechanics in the context that it calculates the evolution of variables associated with the particles within the fluid, instead of inertial referenced positions (as done in the Euler method). This is the strongest characteristic of the method.

In order to explain the SPH method, we can start with the continuity equation, which refers to the particle's mass in the form of density (Streeter and Wylie, 1988):

$$\frac{d}{dt}\rho = -\nabla \cdot \rho\mathbf{v} \tag{1}$$

The particle's acceleration is given by the pressure's gradient as $\frac{d}{dt}\mathbf{v} = -\frac{1}{\rho}\nabla p$, where $\mathbf{v}$ is the velocity if particle ($\mathbf{v} = \frac{d}{dt}\mathbf{r} = \dot{\mathbf{r}}$) and $\mathbf{r}$ stands for the position of the fluid particle. By using the chain rule of $\nabla\left(\frac{p}{\rho}\right) = \frac{\nabla p}{\rho} - \frac{p}{\rho^2}\nabla\rho$, particle's acceleration can be written as

$$\frac{d}{dt}\mathbf{v} = -\nabla\left(\frac{p}{\rho}\right) - \frac{p}{\rho^2}\nabla\rho, \tag{2}$$

On the other hand, the interpolation function

$$\langle f(r)\rangle = \int_D f(r')W_{r,h}(r')dr' \cong \sum_{j=1}^{N} f(r_j)W_{r,h}(r_j)V_j \tag{3}$$

is the Monte Carlo numerical integration of equation $\langle f(r)\rangle = \int_D f(r')W_{r,h}(r')dr'$ in order to obtain discrete points from the information of finite set of known points r_j. The function $\langle f(r)\rangle$ is the average of function $f(r)$ around point r, where D is the domain of this function, $W(r)$ is the *kernel* and V_j is the volume of point r_j.

Using (3) in (2), a particle's movement equation is obtained as (Pérea et al., 2005):

$$\frac{d}{dt}\mathbf{v} = -\sum_j m_j\left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} + \Pi_{i,j}\right)\nabla_i W_{i,j} + \bar{\mathbf{f}}_i \tag{4}$$

Two terms are added in (4): the first is an artificial viscosity $\Pi_{i,j}$, which is a dissipation term that should tend to decrease the kinetic energy of the particles when no external forces are acting on the set of particles. The second term is indeed the external force, $\bar{\mathbf{f}}_i$, applied to every particle i.

Densities are calculated as:

$$\frac{d}{dt}\rho_i = \sum_j m_j \mathbf{v}_{i,j} \cdot \nabla_i W_{i,j} \tag{5}$$

where $\mathbf{v}_{i,j} \triangleq \mathbf{v}_i - \mathbf{v}_j$ is the relative velocity of particles i and j.

Kernel. The kernel function determines the interaction among the different particles of the fluid. Each particle interacts only with its nearest neighbor particles, inside its influence zone. We use B-splines (Monaghan and Lattanzio, 1985) defined as:

$$W_{i,j}(\mathbf{r}_i,\mathbf{r}_j,h) \triangleq \begin{cases} C\left(1-\frac{3}{2}q^2+\frac{3}{4}q^3\right), & \text{if } 0 \le q < 1 \\ C\left(\frac{1}{4}(2-q)^3\right) & \text{if } 1 \le q < 2 \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where h is a half of the influence radio, $q = r_{i,j}/h$ is the normalized distance $r_{i,j}$, and this last one is the magnitude of the relative position of particles i and j as $r_{i,j} = \|\mathbf{r}_{i,j}\|$, where $\|\cdot\|$ stands for the norm of any vector, i.e: $\|\mathbf{a}\| = \sqrt{a_x^2+a_y^2+a_z^2}$ and $\mathbf{r}_{i,j} = \mathbf{r}_i - \mathbf{r}_j$. The constant parameter C is equal to $2/(3h)$ for one dimension, $10/(7h^2)$ for 2D and $1/(\pi h^3)$ for 3D.

Artificial Viscosity. The main purpose of the artificial viscosity term in equation (4) is to avoid the presence of oscillations or nonphysical collisions in the simulation procedure. It is calculated as:

$$\Pi_{i,j} \triangleq \begin{cases} \frac{-\alpha\bar{c}_{i,j}\mu_{i,j}+\beta\mu_{i,j}^2}{\bar{\rho}_{i,j}} & \text{for } \mathbf{v}_{i,j}\cdot\mathbf{r}_{i,j} < 0; \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

where $\mu_{i,j} \triangleq \frac{h\mathbf{v}_{i,j}\cdot r_{i,j}}{{r_{i,j}}^2+0.001h^2}$ is a normalized version of the relative velocity between two particles, with an artificial term in order to avoid numerical ill conditioning; $\bar{\rho}_{i,j} \triangleq (\rho_i+\rho_j)/2$ is the average of densities of the two particles; and $\bar{c}_{i,j} \triangleq (c_i+c_j)/2$ is the average of sound's velocity due to their different density. The constants α and β stand for the laminar and turbulent flow dissipation and according to (Monaghan, 1994) they take values of $0.1-0.01$ and 0, respectively.

To use SPH with incompressible fluids, *slightly compressibility* must be considered to allow dynamic simulation of these fluids. This is achieved considering the atmospheric pressure negligible as:

$$p = B\left(\left(\frac{\rho}{\rho_0}\right)^{\gamma} - 1\right) \tag{8}$$

where ρ_0 is a reference density (e.g. for water is 1000). This relationship is known as the *state equation* (which in this context refers to the physical phase of the fluid and not to the dynamic *state* as usually referred in control engineering jargon).

If constants γ and B are high enough (for example, take 7 and 3000 respectively), state equation (8) computes constrained variation on pressures (under 10^5 atmospheres for water in the example). In this case, sound's velocity is sufficiently high and variations in the relative density are small:

$$\frac{|\delta\rho|}{\rho} \approx \frac{v^2}{c_s^2} \tag{9}$$

where v is the maximum fluid velocity. Moreover, if $v/c_s < 0.1$, it can be assumed that $|\delta\rho|/\rho \approx 0.01$. In this case, sound's velocity can be calculated as $c_s^2 = (\gamma B)/\rho_0$. Thus, if $B = 100\rho_0 v^2/\gamma$, then the variations on the relative velocity can take values of the order of 0.01. The calculations finish by approximating the maximum fluid velocity by $v^2 = 2gH$, where g is the gravity constant and H is the fluid's working area (Monaghan, 1994).

3 COMPLIANT SOLID MODELING

In contrast with rigid bodies, compliant solids can not be represented by dynamical lumped equations. This means that the order of the time varying dynamical equation should be infinite. Reduction of the order of this type of equations, for practical simulation purposes, yields to the so called Finite Element Methods (FEM). These methods consist basically in discretisize the body on a finite number of small simple mechanical models. Then a set of simultaneous but not so complex equations may be solve by different numerical methods. The new problem is then determined by the border or boundary conditions that exist in the new compliant solid.

In the next paragraphs, we propose a new method to calculate these boundary conditions.

3.1 Modeling with Simplex Meshes

Simplex meshes are used to represent surfaces in the three-dimensional space. These meshes have similarities with triangulations, in fact a 2-simplex mesh is the topological dual of the triangulation, but they are not geometric duals. This means that, we can not build a geometric transformation between triangulations and simplex meshes. An important property of simplex meshes is their constant vertex connectivity: each vertex in the mesh has three and only three neighboring vertices. This condition allows to use the same deformation engine to solve four differential equations for the four mass-spring-dashpot systems attached on each simplex. In addition, it has the advantage that allows smooth deformations in a simple and efficient manner. In this work we used the model of a sphere built as is presented in (Flores and de la Fraga, 2004).

A surface made with simplex meshes can be visualize it as a mesh of hexagons, and it is easy to represent each hexagon with four triangles. Then, each triangle can be modeled as a single compliant solid surface. The result is a compliant solid body with arbitrary three-dimensional geometry.

3.2 Fluid Particles Interaction

The interaction of a fluid particle and a surface can be modeled by the interaction forces or pressures. The same force, in opposite sense, is exerted by the surface to the particle.

The contact force that the surface exert to the particle can be modeled in two orthogonal components as $\mathbf{f}_c = \mathbf{f}_d + \mathbf{f}_f \in \mathbb{R}^3$, where $\mathbf{f}_d$ is the deformation force due to the elastic stresses and mechanical deformation of the body surface and in this work is considered to be strictly perpendicular (normal) to the surface. The friction component $\mathbf{f}_f$ is considered to be strictly tangential to the surface.

The Normal Operator. Lets be $s(x,y,z) = 0$ the function in the Euclidean space defining the surface with whom the particle is contact at point $\mathbf{r}_c = (x_c, y_c, z_c)^T$, where the variables x_c ,y_c and z_c are the Cartesian coordinates of the contact point. The unit vector $\lambda_N \in \mathbb{R}^3$ is defined as the normalized gradient of the surface at point $\mathbf{p}_c$:

$$\lambda_N \triangleq \frac{\nabla s(\mathbf{p}_c)}{\|\nabla s(\mathbf{p}_c)\|} \tag{10}$$

The deformation can be calculated as the normal component at the relative position of the particle $\mathbf{r}(x,y,z)$ with respect to the contact point at the surface $\mathbf{x} \triangleq \mathbf{r} - \mathbf{p}_c \in \mathbb{R}^3$. The normal component $\mathbf{x}_N$ is obtained from the next equation

$$\mathbf{x}_N = N\mathbf{x} \tag{11}$$

where the Normal Operator N is defined as

$$N \triangleq \lambda_N \lambda_N^T \tag{12}$$

The normal component of the relative velocity can be calculated using the Normal Operator N as

$$\dot{\mathbf{x}}_N = \lambda_N \lambda_N^T \dot{\mathbf{x}} \tag{13}$$

where $\dot{\mathbf{x}} = \mathbf{v} - \dot{\mathbf{p}}_c$, is the relative velocity between the particle and the surface, $\mathbf{v}$ and $\dot{\mathbf{p}}_c$ are the linear velocities of the particle and the contact point of a rigid surface, respectively [1].

The Tangential Operator. As well as for the normal component of the deformation or velocity to the surface, a linear operator $T : \mathbb{R}^3 \mapsto \mathbb{R}^3$ shall exist that gives the tangential projection of these vectors. This Tangential Operator must be of the form

$$\dot{\mathbf{x}}_T = T\dot{\mathbf{x}} \tag{14}$$

In the same way that Normal Operator has been defined in (12), The Tangential Operator is :

$$T \triangleq \lambda_T \lambda_T^T \tag{15}$$

where λ_T stands for any unit vector tangent to the surface s at point $\mathbf{p}_c$. One possible way is defining it in the direction of the tangent velocity as:

$$\lambda_T \triangleq \frac{\dot{\mathbf{x}}_T}{\|\dot{\mathbf{x}}_T\|} \tag{16}$$

By construction, operators N and T shall be orthogonal complements and fulfill next properties

$$NT = TN = [0], \text{ and } N + T = I \tag{17}$$

Then the Tangential Operator defined as

$$T = -[\lambda_N \times]^2 \tag{18}$$

fulfill properties (17), where the operator $[\mathbf{a}\times]$ stands for the matrix representation of the cross product: $\mathbf{a} \times \mathbf{b} = [\mathbf{a}\times]\,\mathbf{b}$.

Deformation Normal Force. This force, normal to the surface can be modeled as a simple second order, hence continuous system, from the compliant model of the surface. That is, the normal force exerted by a particle of mass m_p, when colliding with a surface s at point p_c with velocity $\dot{\mathbf{x}}$ is given by the following equation

$$\mathbf{f}_p = k_s \mathbf{x}_N + b_s \dot{\mathbf{x}}_N \tag{19}$$

where the constant k_s is the elastic modulus of the surface material, b_s is an artificial damping coefficient and variables $\mathbf{x}_N$ and $\dot{\mathbf{x}}_N$ are the normal projections of the relative position and velocity respectively of the particle in the contact point.

By Newton's laws, the reaction force, i.e. the one felt by the particle and induced by the surface is

$$\mathbf{f}_d = -\left[\lambda_N {\lambda_N}^T\right](k_s \mathbf{x} + b_s \dot{\mathbf{x}}) \tag{20}$$

In order to simulate a non absorbent material, the artificial damping coefficient b_s should be chosen sufficiently small with the restriction $b_s << 2\sqrt{k_s m_p}$.

Friction Tangential Force. This force can be modeled by a simple Coulomb friction as

$$\mathbf{f}_f = \mu_k \|\mathbf{f}_d\| \,(-\lambda_T) \tag{21}$$

where μ_k is the dynamic friction coefficient between the particle and the contact surface. The $\mathbf{f}_f$ direction is given by the unit tangent vector if it is defined as in (16). Then, equation (21), can be rewritten as:

$$\mathbf{f}_f = -\mu_k \|\mathbf{f}_d\| \frac{\dot{\mathbf{x}}_T}{\|\dot{\mathbf{x}}_T\|} \tag{22}$$

where $\dot{\mathbf{x}}_T$ is calculated by (14).

3.3 Integration

The sum of the deformation force (20) and the friction one (22) is the total contact force $\mathbf{f}_c$ that each particle feels when colliding with a surface. It shall be normalized by the particle's mass in order to be included in equation (4).

To be included in the model of the compliant body made up of simplex meshes the forces of all the particles that collide with a specific planar surface (a triangle in our case) are added and then normalized by the area of that triangle in order to include these interface forces as border pressure. The resulted force is applied at the three triangle's vertices.

4 IMPLEMENTATION DETAILS

The next algorithm shows our implemented SPH method:

```
Set initial time, $t \leftarrow t_0$,
Set time increment $\Delta$,
Set stop time, $T \leftarrow t_f$,
while $t \leq T$ do
    for all $p_i$ particle do
        Find the neighbor particles to $p_i$, using (6).
        Calculate the acceleration with (4), and
```

[1] Note: If the surface is considered uncompliant and with no movement, velocity at the contact point is null, $\dot{p}_c = 0$

```
8:        Calculate the density change for $p_i$ using
          continuity expression (5).
9:    For each particle, update its position, velocity
      and density, using leapfrog numerical method.
10:   For each particle, calculate its pressure (8).
11:   $t+=\Delta$
```

The most time consuming step is finding the neighbor particles to the given one in line 6 of the SPH algorithm, which is the calculation of equation (6). For n particles, $(n)(n-1)/2$ calculations of distance among each particle pair must be performed. Therefore, to reduce the execution time in the 3D simulations, we create a grid of cubes of size $2h \times 2h \times 2h$. In each cube there is a list of belonging particles. When a particle moves, its indexes are recalculated to a new position, and the particle is moved to the corresponding list. In this manner, we reduce the search of neighbors of the whole set of particles into a neighborhood of 26 cubes around the given one. However, the neighbor finding algorithm is still of complexity $O(n^2)$; this part of the algorithm could be improved with more complex data structures.

For locating if a particle it is inside the kernel, the workplace, or in a side of a plane, we use a simple collision detector which consists in testing the sign of the value given by the implicit equation of the interface.

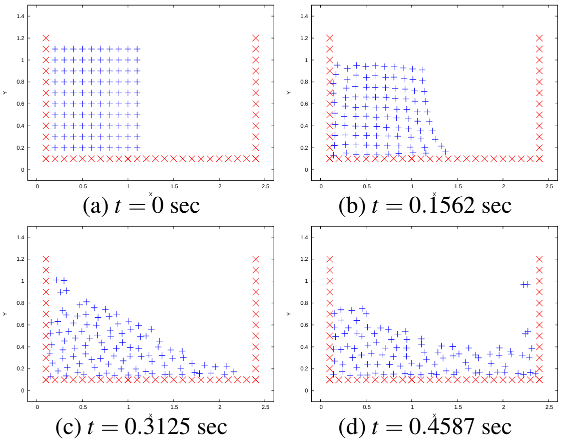
For the deformable object we use SOLID collision detector (van den Bergen, 2004), which it is optimized to calculated the collision among a ray and a set of triangles.

4.1 Results

We perform two experiments to test our simulator. The first experiment is a classical simulation of rupture of a dam. The second experiment is a deformable sphere inside a tube in which is circulating an incompressible fluid. We are going to describe in detail each experiment.

We performed the first experiment with border particles in order to test our SPH implementation. In Fig. 1(a), we can see the initial state of the dam. The working space is made up with the static particles marked as "×". The set of particles marked with "+" form the incompressible fluid. A border to the right of the fluid (the dam's border) is removed at time t_0. All fluid's particles suffer a gravity force. In Fig. 1(b)-(d) we can see the carried simulation. The simulation works as expected, and it is similar to the showed in (Monaghan, 1994).

The second experiment uses compliant simplex meshes surfaces to model a tube where an incompressible fluid is running and the surface of a deformable sphere is placed inside the moving fluid. The fluid moves by the action of an horizontal force equal to the gravity force. It is difficult to check by visualization that the simulation is running ok in this scenario. Thus, values along the fluid are taken: the average particle velocity through a transverse tubes' section, and the average pressure through longitudinal tubes' sections. These values are shown on Fig. 2, and we can see that simulation's results are correct.


(a) $t = 0$ sec

(b) $t = 0.1562$ sec

(c) $t = 0.3125$ sec

(d) $t = 0.4587$ sec

Figure 1: Simulation of a dam's rupture.

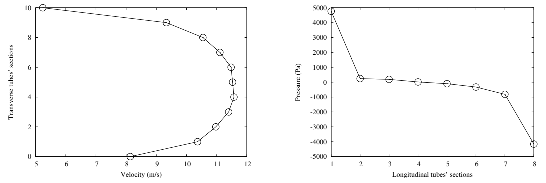


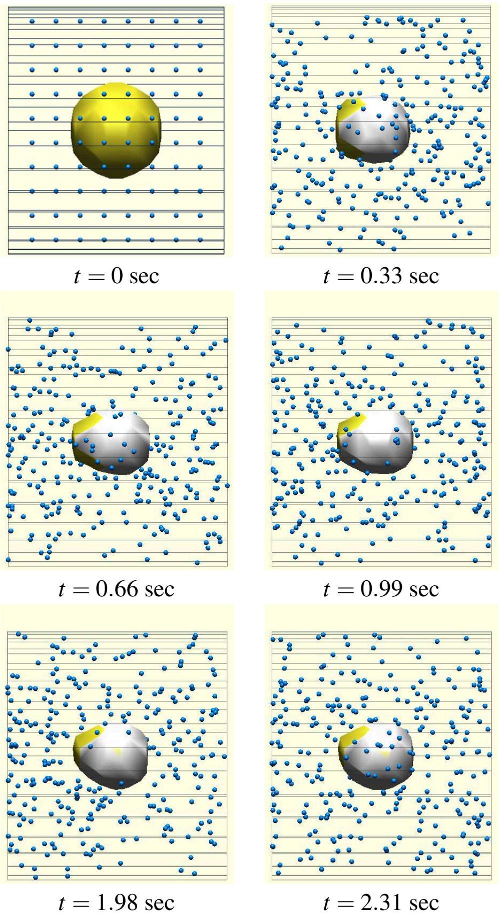
Figure 2: Left curve shows the variation of the mean velocity (horizontal axis) taken at a transverse section of the tube (vertical axis). Right curve shows the average pressure (horizontal axis) along the tube (horizontal axis).

Finally, in Fig. 3, we see several views of the simulation of the deformable sphere inside the fluid. The deformable surface takes the form of an ovoid, as it is expected.

5 CONCLUSIONS

We developed a fluid dynamics simulator based in the Smooth Particle Hydrodynamics method plus the use of compliant bodies modeled with simplex meshes. The interface interaction has been modeled by orthogonal forces produced by each particle colliding with the surface and simple models of 2^{nd} order compliant deformation and Coulomb friction.

We can simulated containers or deformable objects contact with slightly compressible fluids. Our



$t = 0$ sec

$t = 0.33$ sec

$t = 0.66$ sec

$t = 0.99$ sec

$t = 1.98$ sec

$t = 2.31$ sec

Figure 3: Simulation of a deformable sphere immerse in a fluid contained within a tube.

simulator uses a simplified model of a compliant body that allows fast computations. Our simulator can be used in visualization of a real phenomena like to cut a vein and to show how the blood fluids.

We showed bidimensional simulations of the rupture of a dam and a three-dimensional simulation of a compliant sphere completely immersed in a running incompressible fluid inside a tube. The results simulate qualitative behaviors of the proved systems.

We are convinced that our method can be used as a first approximation, or to show graphically the behavior, of partial immersed bodies on compressible or incompressible fluids, like ships or submarines at the see surface.

Also, the forces produced in immersed complex mechanism, like robot arms attached to a ROV (Remotely Operated Vehicle) can be better understood if more accurate hydrodynamical damping models can be obtained. This method can be a good approach to this end.

Finally, some work has to be done yet to reduce the computational cost when a complex deformable model is used, in order to obtain real time simulations.

ACKNOWLEDGEMENTS

This work has been partially supported by project SEP-CONACyT 80965, México.

REFERENCES

Chen, S., Johnson, D. B., and Raad, P. (1995). Velocity boundary conditions for the simulation of free surface fluid flow. *Journal of Computational Physics*, Volume 116(Issue 2, February):262–276.

chuan Bai, Y., Xu, D., and q. LU, D. (2007). Numerical simulation of two-dimensional dam-break flows in curved channels. *Journal of Hydrodynamics*, Ser. B, Volume 19(6):726–735.

Eddy, P., Natarajan, A., and Dhaniyala, S. (2006). Subisokinetic sampling characteristics of high speed aircraft inlets: A new cfd-based correlation considering inlet geometries. *Journal of Aerosol Science*, Volume 37(Issue 12, December):1853–1870.

Flores, J. R. and de la Fraga, L. (2004). Basic three-dimensional objects constructed with simplex meshes. In *First International Conference on Electrical and Electronics Engineering*, pages 166–171. IEEE Pres.

Monaghan, J. and Lattanzio, J. (1985). A refined method for astrophysical problems. *Astron. Astrophysics*, 149:135–143.

Monaghan, J. J. (1994). Simulating free surface flows with sph. *Journal of Computational Physics*, 110:399–406.

Monaghan, J. J. (2005). Smoothed particle hydrodynamics. *Rep. Prog. Phys.*, 68:1703–1759.

Pérea, S. A., González, F. F., and Iglesias, A. S. (2005). Utilización del método sph en la simulación del sloshing. Technical Report No. 822, Universidad de la Rioja.

Streeter, V. L. and Wylie, E. B. (1988). *Fluid Mechanics*. Mc. Graw Hill.

Tai, C. H., Liew, K. M., and Zhao, Y. (2007). Numerical simulation of 3d fluid-structure interaction flow using an immersed object method with overlapping grids. *Computers & Structures*, Volume 85(Issues 11-14, June-July):749–762.

Trejo, C. R. and de la Fraga, L. G. (2005). Animation of deformable objects built with simplex meshes. In *Second International Conference on Electrical and Electronics Engineering*, pages 36–39. IEEE Pres.

van den Bergen, G. (2004). *Collision Detection in Interactive 3D Environments*. Morgan Kaufmann.

SELF-LOCALIZATION OF A TEAM OF MOBILE ROBOTS BY MEANS OF COMMON COLORED TARGETS

Patricio Nebot and Enric Cervera
Department of Computer Science and Engineering, Jaume-I University, Campus de Riu Sec, Castellón de la Plana, Spain
pnebot@icc.uji.es, ecervera@icc.uji.es

Keywords: Mobile robots, Visual localization, Color vision, Common objects.

Abstract: Robot localization is one of the fundamental problems in mobile robotics. Using sensory information to localize the robot in its environment is the most fundamental problem that has to be solved in order to provide a mobile robot with autonomous capabilities, But, if robots can detect each other, there is the opportunity to do better. In this paper, it is explained how one robot, the leader, with a pan-tilt-zoom camera mounted on it can localize a team of robots. Camera images are used to detect other robots and to determine the relative position of the detected robot and its orientation with respect to the leader. Each robot carries a colored target that helps the leader to recognize it and calculate their position and orientation. Moreover, the zoom is used to enhance the perception and get a higher accuracy and a larger field of view.

1 INTRODUCTION

Cooperation among a group of robots has been a topic of very much study during the last years. To have a cooperative system it is necessary for more than one organism to have a relationship with another or anothers. So, to implement cooperation in a robotic system, it is necessary to have more than one robot working in the same environment, that is, a multirobot system.

There are several applications in which having more than one robot working in parallel has improved the system's fault tolerance, and has reduced the time required for executing the tasks. Some of these applications are autonomous cleaning, tour guiding in museums, art-galleries, or exhibitions, surveillance and patrolling, rescue in disasters such as fires, floods, earthquakes, landmine detection and autonomous exploration and mapping.

In more of these applications it is necessary the use of vision in order to implement or acomplish the tasks. Human and animal vision are the most powerful perception systems. Vision is a sense that consists of the ability to detect the light and interpret it, that is "see". Vision gets help from multiple information sources to interpret the world. The visual system allows to assimilate information from the environment to help guide the actions.

One of the most important task in computer vision is recognition, which consists of determining whether or not the image data contains some specific object, feature, or activity. One of the most characteristic task in recognition is "pose estimation", that estimates the position or orientation of a specific object relative to the camera.

In this paper, a similar task is implemented. In this case, one robot with a camera tries to estimate the pose of the rest of robot of the team, which don't have a camera available. In that way, the robot with the camera can help the rest of robots in case of lost of their possitions due to inaccurate odometric estimation pose.

To determine the relative location of other robots, the leader uses the visual information obtained from the pan-tilt-zoom on-board camera. Camera images are used to detect other robots and to determine the relative position of the detected robot and its orientation with respect to the leader. Each robot carries a colored target that helps the leader to recognize it and calculate their position and orientation. Moreover, the zoom is used to enhance the perception and get a higher accuracy and a larger field of view.

Robot localization has been recognized as one of the fundamental problems in mobile robotics (Fox et al., 2000). Using sensory information to localize the robot in its environment is the most fundamental problem that has to be solved in order to provide a mobile robot with autonomous capabilities (Cox and Wilfong, 1990). Most of the existing work in localization is addressed to the localization of a single robot by itself. However, if robots can detect each other, there is the opportunity to do better. When a robot determines the location of another robot relative to its own, both robots can improve the accuracy with which they localize each other.

Vision has been widely used to get exteroceptive information in order to detect and localize robots. Although omnidirectional cameras have been used in the detection and localization of robots (Das et al., 2002), directional cameras suppose a better option due to their much lower cost (Sarcinelli-Filho et al., 2003) and because they have complementary performances despite the visibility constraints (Michaud et al., 2002). Regarding the image processing, color has been widely used to achieve robot detection (Fredslund and Mataric, 2002; Michaud et al., 2002). However, the robustness of color detection with respect to light conditions can be a major source of failures (Cubber et al., 2003).

Also, the use of the zoom has been used in the context active vision (Atienza and Zelinsky, 2001) or visual servoing (Hosoda et al., 1995). For using the zoom, it is necessary the explicit knowledge of intrinsic parameters from the calibration of the camera (Clady et al., 2001).

The rest of the paper is organized as follows. Section 2 provides a description of the experimental setup. Section 3 explains the process to lozalize the different robots of the team. Finally, section 4 provides some general conclusions and lines of future work.

2 EXPERIMENTAL SETUP

- **Hardware Setup**
  The team for this application consists of a group of four Pioneer-2 mobile robots. These robots, though sharing the same platform, have different features, such as different accessories mounted on them, constituting therefore a heterogeneous group. In particular, only one robot is equipped with a camera and the rest of robots do not have any type of exteroceptive sensors. The robot with the camera is in charge of detecting the rest of the robots in the environment and indicates to them which is their current position in the environment.
- **Software Setup**
  The formation control is developed in Acromovi (Nebot and Cervera, 2005), a framework specially designed for the development of distributed applications for a team of heterogeneous mobile robots. The software architecture gives us the ease of development of cooperative tasks among robots, using an agent-based platform. In particular, communication between robots can be easily integrated to the control scheme.

3 LOCALIZATION OF THE ROBOTS IN THE ENVIRONMENT

This section describes how the robots can be localized. Having the robots distributed in an environment, the robot with the camera, from this point the *leader*, uses the camera that it carries to detect and localize the rest of the robots of the team and indicates to them which is their current. This process is explained below in more detail.

The first step is the detection of the robots in the environment, in order for this decision to be taken, the leader uses its camera to detect a series of color patterns which identify each one of the robots in an unequivocal way. To detect the colors, the Mezzanine program is used.

Mezzanine (Howard, 2002) is an 2D visual tracking package intended primarily for use as a mobile robot metrology system. It uses a camera to track objects marked with color-coded fiducials and infers the pose of these objects in world coordinates. Mezzanine works with most color cameras and can correct for the barrel distortion produced by wide-angle lenses.

Mezzanine is used only for the detection of the colors of the patterns that are used to recognize the robots. And with the information that is collected from the mezzanine system, it is possible to localize the robots in the environment and calculate their pose (position and orientation) with respect to the leader.

The color pattern that the leader has to search for is created with very common object, a beer can of half a liter covered with colored cards in a specific layout, because each ID is unique. In figure 1, it is possible to see the sizes and dimensions of the target and the color layout at the front and at the back.

As it can be seen in the ID, there are two different parts separated by a black zone. These two parts are formed with the same colors and the same cards, but there is a 90^o difference in orientation between the ID cards which means that thay are read as different cards. That is in this way to get two different readings of the orientation of the can and thus getting more accurate estimations.

Since each robot carries a different color target, the leader is able to recognise each one. Also, with this pattern it is possible to calculate the pose of the robot in relation to the leader. It is easy to recognize which robot the camera is seeing, it is simply neces-

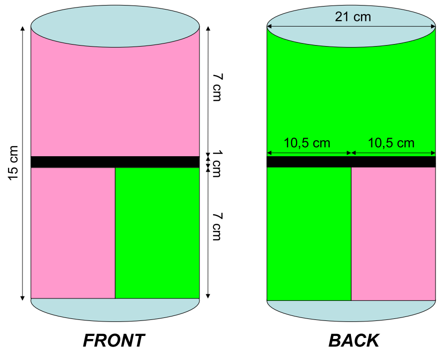


Figure 1: Dimensions and color layout of the IDs.

sary to pay attention to the layout of the colors. Mezzanine can detect several colors at the same time and group the different areas of the same color in blobs. With the information associated with these blobs it is possible to know which ID the camera is seeing at that moment.

The movement that the camera performs to find these IDs is firstly horizontal movement of 180^o from left to right. If nothing is found in this movement, the camera increases its vertical position in 5^o up to a maximum of 30^o. If when this process has finished, still any robot has not been identified, the leader executes a 180^o turn and repeats the same process until all the robots are found. In this way, the leader searches all the space around it for the other robots.

Throughout the searching process, mezzanine is monitoring all their channels where it has assigned a predefined color, and in the moment that it finds anyone of them, the camera is stopped. From this moment, a centering process begins. This new process tries to center the pattern found in the middle of the image. To this end, because the robots can be in movement, it is important to center the target in a minimum number of movements, and at the same time it is important to maximize the zoom of the camera to make a better identification in the following phases.

In order to center the target, when mezzanine detects one blob of any color bigger than a certain size, the robot stops the searching process previously described. The size has to be big enough to rule out possible errors of the program or reflections of the target. From this blob it is possible to know the position of its mass center in the image system, so the space between this and the center of the image can be calculated.

To translate this distance into a movement of the camera, it is necessary to know some intrinsic parameters of the camera, such as the focal length. These parameters can be obtained with a previous calibration of the camera.

As the camera includes a zoom, the focal length must be calculated for each of these values of the zoom. After several tests performed in the laboratory by an student, Vincent Robin, during a stay there, he managed to model the behaviour of the focal length depending on the zoom. The function that models these behaviours can be defined as:

$$f_v = (0.0368323 - 0.0000128323 * z)^{-2} * 1/2 \quad (1)$$

being z the value of the zoom that is desired. This function can be visualized in figure 2, and as it can be seen, the focal length does not follow a linear progression with the progression of the zoom.

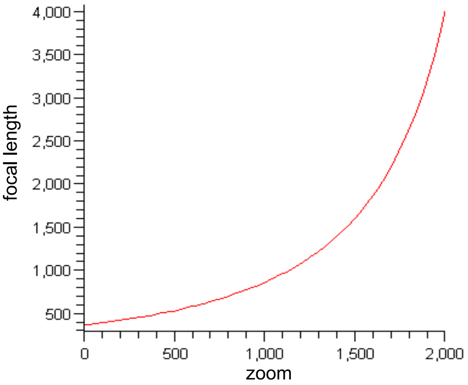


Figure 2: Relation among the necessary parameters to calculate the distance to the ID.

Knowing the value of the focal length for the actual zoom of the camera, it is simple to calculate the movements that the camera must perform in order to center the blob in the image. There are two movements that have to be made, in the pan (∇p), that is, in horizontal, and in the tilt or vertical (∇t). These values can be calculated as:

$$\begin{aligned} \nabla p &= \arctan\left(\frac{(x - x_i)}{f}\right) \\ \nabla t &= -\arctan\left(\frac{(y - y_i)}{f}\right) \end{aligned} \quad (2)$$

where x and y correspond to the coordinates of the mass center of the blob in the image, and x_i and y_i correspond to the center of the image. With these values the first blob that the robot finds can be centered.

Once the first blob of the target is centered, it is possible to calculate the optimal value of the zoom in order to reduce the detection failures of the targets of the robots and making sure of a better approximation in the calculation of the position of the robot. These calculations are based on the previous work of Pierre Renaud (Renaud et al., 2004) during a stay in this laboratory.

Prior to calculating the optimal zoom, it is necessary to calculate the optimal focal length, and with this value is possible to calculate the optimal zoom. In order to calculate the optimal focal length, it is necessary to know the actual distance (Z) to the target.

$$Z = f_v * \frac{h}{\Delta v} \tag{3}$$

With this distance, and knowing which is the desired height (h_{des}) of the blob that the program needs to get the optimal configuration, and knowing the height of the blob in the image, the optical focal length can be calculated.

$$f_{op} = h_{des} * \frac{Z}{h} \tag{4}$$

Finally, as deduced by Renaud in his work, the optimal zoom can be calculated merely by knowing the optimal focal length.

$$z_{op} = 77928.35 * \left(3.6833e^{-2} - (2 * f_{op})^{-0.5}\right) \tag{5}$$

Once this calculation is made, the zoom is applied to the camera, and as the rest of the blobs or colors of the target are now visible, it is possible to identify the robot. This process is simple and merely perceiving the distribution of the different colors in the target, the different robots can be identified.

Next, in order to make the calculation of the position more precise, a new centering process is carried out, but this time taking into account the blobs of the other colors present in the image. The biggest blob of the other color in the image provides the system with enough information to center the target in the image. The new equations to calculate the movements of the camera to center the target are:

$$\begin{aligned} \nabla p &= \arctan\left(\frac{((x_1+x_2/2)-x_i)}{f}\right) \\ \nabla t &= \arctan\left(\frac{((y_1+y_2/2)-y_i)}{f}\right) \end{aligned} \tag{6}$$

Once the ID is centered on the image and with the maximum size possible, its position and orientation with respect to the camera, or the leader, can be calculated. To know its pose $(x, y, angle)$, it is necessary to perform some calculations with the image.

In order to calculate (x,y), the system needs to know the distance and the angle of the ID with respect to the leader, and from these values, calculates the position.

To calculate the distance from the ID to the image, it is necessary firstly to know some parameters. These parameters are the real height of the ID (h), the height in pixels of the ID in the image (∇h), and the focal length of the camera (f_v). With these values, the distance to the ID (Z) can be calculated as:

$$Z = f_v \frac{h}{\Delta v} \tag{7}$$

The height of the ID is fixed, and the height in pixels of the ID in the image can be obtained from the blob information from mezzanine. The focal length of the camera can be obtained as explained before, merely by knowing which is the actual value for the zoom of the camera.

The precision of the approximated distance depends on the capacity of the system to recognize the specific colors of the cylinders, which is influenced by the prevailing lighting conditions. When the cylinder is lit from the side, their colors are preceived no longer uniform, making only part of the width of the cylinders visible to the leader. For an optimal approximation, good uniform lighting is necessary.

The calculation of the orientation (α) at which the robot is depending on the leader is easier, since the camera indicates the orientation that it has (α) at that moment. In figure 3, the relation between orientation and distance for the calculation of the ID position (x,y) can be observed.

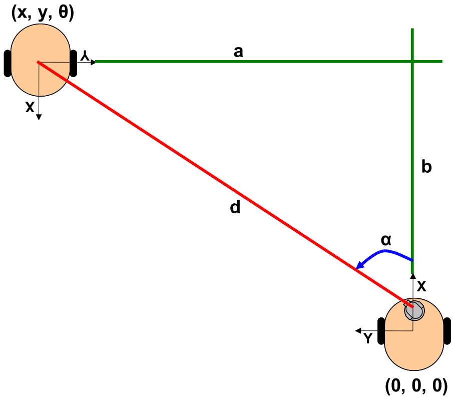


Figure 3: Relation among orientation and distance with the calculation of the ID position.

When the distance and the orientation have been calculated, it is possible to calculate the position (x,y) as:

$$\begin{array}{lcl} d = \sqrt{a^2+b^2} & & x = b = d \cdot \sin\alpha \\ & \Longrightarrow & \\ \alpha = \arctan b/a & & y = a = d \cdot \cos\alpha \end{array} \tag{8}$$

Once the position has been calculated, it only remains to calculate the orientation of the robot with respect to the leader. For this calculation, the two horizontal parts or colors of the ID are used, or rather, the relation between these two parts.

Based on the existing relation between the horizontal sizes of the two colored parts, the orientation of the robot can be calculated. It can be observed that the upper layer and the lower layer have the same colors, but the lower layer has them with a specific turn in relation with the configuration of the upper layer. This is done to have two differentiated parts and to calculate the orientation of the upper and lower layer separately, and thus, making the calculation more precise. Moreover, as it will be seen below, in that way it is possible to avoid some positions that are not accurate enough in the calculation of the orientation.

In the calculation of the orientation using the target selected, it is very important to take into account the order of the colors. Regarding this, from the 0^o position of the can to 180^o position, the pink color is in the upper left position and grows until covers the complete side of the can. If the green color is in the upper left, the orientation will be from -180^o to 0^o, depending on the portion of the can occupied by this color. In the two cases, it corresponds the 90^o or -90^o when the two colors occupy the same portion of the can, but depending on the color in the upper left side, the orientation will be positive or negative.

From the relation of the left part (X) and right part (Y), the orientation of the robot can be calculated. The relation among the left and right parts and the entire width of the cylinder can be seen in figure 4.

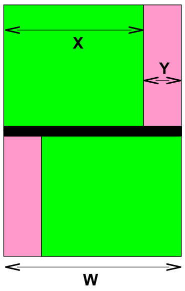


Figure 4: Relation among the parts to calculate the orientation of the robot.

From the figure, it can be deducted when the left and right parts are equal that

$$\frac{2X}{W} = \frac{2X}{X+Y} \tag{9}$$

The behaviour of the cylinder when it is turning can be modeled as $\sin(\alpha - 90) + 1$. Joining this with the previous equation,

$$\frac{2X}{W} = \sin(\alpha - 90) + 1 \tag{10}$$

from which it can be infered the value of α as,

$$\alpha = 90 + \arcsin(\frac{2X}{W} - 1) \tag{11}$$

In the case of the negative orientation, the equation is similar,

$$\alpha = -90 + \arcsin(\frac{2X}{W} - 1) \tag{12}$$

In figure 5, it can be seen the graphics in radians that model these to functions.

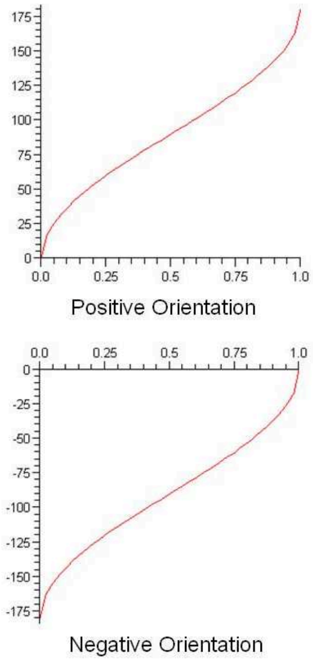


Figure 5: Functions that model the orientation of the robot.

Moreover, it is necessary to take into account four exceptions to the general rule. If there is only one color in one of the parts of the ID (upper or lower), then there is one of these special cases. Simply, distinguishing the order of the colors in the remaining layer, it is possible to recognize the spacial case in question.

Also, as it can be seen in the graphs in figure 5, there are two zones in each graph in which the value of the orientation changes too fast and it may cause the estimation to be less accurate than desired. These zones are at the extremes with a width of 30^o. When the calculation in one of the layers returns a value within one of these zones, the values that are taken for the orientation of the robot is the value that returns the calculation in the other layer. Due to the fact that the layers have different turns, the values that appeared in one layer when the other is in one of these situations are correct.

This calculation of the orientation of the robot can be easily extended to any of the IDs. It is only necessary to take into account which is the pattern of the colors. Furthermore, this identification system can be extended to any number of different IDs, the only limitation is the number of different colors that mezzanine is able to detect.

Once the robot's orientation is calculated, all the values necessary for determining its pose $(x, y, angle)$ in the environment with respect to the leader are available. This pose then is translated to the environment system. This process is obvliuos. Then, this pose is then sent to the corresponding robot so that it knows its position.

4 CONCLUSIONS

A new method for the visual localization of robots has been implemented. Using a very common and simple target it is possible to localize one robot and determine its position and orientation with regard to the robot with the camera and of course in the environment.

The main advantage consists on having a very simple object, by means of the corresponding geometric constrints, it is possible to stablish not only the distance to the target robot, but also the orientation. Regarding to the orientacion, by means of a two simultaneous readings process, it is possible to eliminate the accuracy errors produced by the specific features of the object used as target.

The localization of the robots by means of the colored targets has been a hazardous work due to the sensitivity of the vision system to the lighting conditions.

ACKNOWLEDGEMENTS

Support for this research is provided by the Fundació Caixa Castelló - Bancaixa under project P1-1A2008-12.

REFERENCES

Atienza, R. and Zelinsky, A. (2001). A practical zoom camera calibration technique: an application of active vision for human-robot interaction. In *Proceedings of the Australian Conference on Robotics and Automation*, pages 85–90.

Clady, X., Collange, F., Jurie, F., and Martinet, P. (2001). Objet tracking with a pan tilt zoom camera, application to car driving assistance. In *Proceedings of the International Conference on Advanced Robotics*, pages 1653–1658.

Cox, I. and Wilfong, G. (1990). *Autonomous Robot Vehicles*. Springer Verlag.

Cubber, G., Berrabah, S., and Sahli, H. (2003). A bayesian approach for color consistency based visual servoing. In *Proceedings of the International Conference on Advanced Robotics*, pages 983–990.

Das, K., Fierro, R., Kumar, V., Ostrowski, J. P., Spletzer, J., and Taylor, C. (2002). A vision-based formation control framework. *IEEE Transactions on Robotics and Automation*, 18(5):813–825.

Fox, D., Burgard, W., Kruppa, H., and Thrun, S. (2000). A probabilistic approach to collaborative multi-robot localization. *Autonomous Robots*, 8(3).

Fredslund, J. and Mataric, M. (2002). A general, local algorithm for robot formationss. *IEEE Transactions on Robotics and Automation (Special Issue on Advances in Multi-Robot Systems)*, 18(5):837–846.

Hosoda, K., Moriyama, H., and Asada, M. (1995). Visual servoing utilizing zoom mechanism. In *Proceedings of the International Conference on Advanced Robotics*, pages 178–183.

Howard, A. (2002). Mezzanine user manual; version 0.00. Technical Report IRIS-01-416, USC Robotics Laboratory, University of Sourthern California.

Michaud, F., Letourneau, D., Guilbert, M., and Valin, J. (2002). Dynamic robot formations using directional visual perception. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2740–2745.

Nebot, P. and Cervera, E. (2005). A framework for the development of cooperative robotic applications. In *Proceedings of the 12th International Conference on Advanced Robotics*, pages 901–906.

Renaud, P., Cervera, E., and Martinet, P. (2004). Towards a reliable vision-based mobile robot formation control. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3176–3181.

Sarcinelli-Filho, M., Bastos-Filho, T., and Freitas, R. (2003). Mobile robot navigation via reference recognition based on ultrasonic sensing and monocular vision. In *Proceedings of the International Conference on Advanced Robotics*, pages 204–209.

ANALYTICAL KINEMATICS FRAMEWORK FOR THE CONTROL OF A PARALLEL MANIPULATOR

A Generalized Kinematics Framework for Parallel Manipulators

Muhammad Saad Saleem, Ibrahim A. Sultan
School of Science and Engineering, University of Ballarat, Mount Helen, VIC, Australia
{m.saleem, i.sultan}@ballarat.edu.au

Asim A. Khan
School of Electrical and Electronic Engineering, University of Manchester, Manchester, U.K.
asim.khan@postgrad.manchester.ac.uk

Keywords: Analytical kinematic function, Jacobian, Forward kinematics, Inverse kinematics, Closed-loop manipulators, Parallel manipulators, Operational space control.

Abstract: Forward and inverse kinematics operations are important in the operational space control of mechanical manipulators. In case of a parallel manipulator, the forward kinematics function relates the joint variables of the active joints to the position of end-effector. This paper finds analytically forward kinematics function by exploiting the position-closure property. Using the proposed function along with the analytical Jacobian presented in the literature, the forward and the inverse kinematics blocks are formulated for a prospective operational space control scheme. Finally, an example is presented for a 3-R$\underline{P}$R robot.

1 INTRODUCTION

The end-effector of a parallel manipulator is connected to its *base* via a number of serial manipulators in parallel. In these manipulators, there are always more joints than the number of degrees of freedom (DOF) of the end-effector. This places constraints on the structure such that all the joints cannot be actuated at the same time. If the end-effector has l DOF, then there are l active joints where $l \leq 6$. All the other joints are passive and their motion is dependant on the motion of the active joints. The most famous family of such manipulators are called *Stewart-Gough* platforms (Bhattacharya et al., 1997). These platforms are widely used in simulators (Yamane et al., 2005), low impact docking systems for space vehicles (Timmons and Ringelberg, 2008), and in form of a hexapod for precise machining (Warnecke et al., 1998).

Figure 1 shows a 3-R$\underline{P}$R robot, which has three joints in each serial link. *R* stands for a rotatory joint and *P* stands for a prismatic joint whereby the underline signifies the joint which is actuated (Siciliano and Khatib, 2007).

The forward kinematics function of a parallel has been studied in detail in the literature, especially for a 3-R$\underline{P}$R robot. (Kong, 2008) derived algebraic expressions for the forward kinematics of a 3-R$\underline{P}$R robot and analyzed its singularities. (Collins, 2002) used planar quaternions to formulate kinematic constraints in equations for a 3-R$\underline{P}$R robot. (Murray et al., 1997) used coefficients of a *constraint manifold*, which are functions of the locations of the base and platform joints and the distance between them, for the kine-

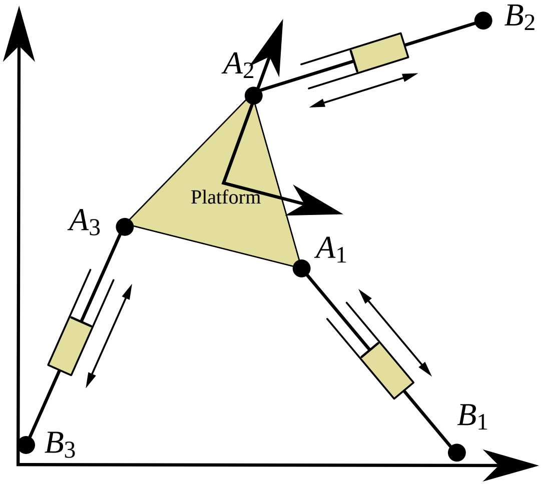


Figure 1: A 3-R$\underline{P}$R planar parallel manipulator. B_1, B_2, and B_3 are connected to a stationary base.

matics synthesis of a 3-R$\underline{P}$R robot. (Wenger et al., 2007) studied the *degeneracy* in the forward kinematics of a 3-R$\underline{P}$R robot. (Kim et al., 2000; Dutre et al., 1997) found the analytical Jacobian for a parallel manipulator. However, there is no attempt in literature to formulate analytically the forward kinematics function for non-redundant parallel manipulators. The forward kinematics function relates the joint variables of the active joints to the position of the end-effector.

In the following section , the structure of the forward and inverse kinematics blocks is layed out. Then forward kinematics function of a parallel manipulator is derived using the position-closure property. The analytical Jacobian of a parallel manipulator is also obtained as described in the literature. Finally, a framework to control a parallel manipulator is proposed, followed by an example for the forward kinematics function of a 3-R$\underline{P}$R robot.

2 KINEMATICS FRAMEWORK

If the task is given in operational space then it becomes inevitable to cater for the non-linearities introduced by the forward and inverse kinematics functions. First, the joint variables are translated into operational space. The resultant is compared to the reference trajectory and the error is then converted back to joint space, as shown in Figure 2.

Suppose there are n serial manipulators in a parallel manipulator that has n_a active joints and n_p passive joints such that the total number of joints is $n_c = n_a + n_p$. If x is the end-effector position and F is the forward kinematics function then the following definitions can be introduced;

$$\begin{aligned} x &= F &(1)\\ \dot{x} &= \frac{F}{\partial q_a}\frac{\partial q_a}{\partial t} = J_c\dot{q}_a &(2)\\ \ddot{x} &= J_c\ddot{q}_a + \dot{J}_c\dot{q}_a &(3)\end{aligned}$$

where $\dot{\bullet}$ signifies differentiation with respect to time, J_c is the systems Jacobian, $\dot{J}_c$ is its time-derivate, and q_a is a vector of active joint variables. These variables are in radians if the joint is revolute or in meters if the joint is prismatic.

Equations (1), (2), and (3) can be combined as follows;

$$\begin{bmatrix} x\\ \dot{x}\\ \ddot{x}\end{bmatrix} = \begin{bmatrix} F_c & 0 & 0\\ 0 & J_c & 0\\ 0 & \dot{J}_c & J_c\end{bmatrix}\begin{bmatrix} q_a\\ \dot{q}_a\\ \ddot{q}_a\end{bmatrix} \quad (4)$$

or

$$\begin{bmatrix} x\\ \dot{x}\\ \ddot{x}\end{bmatrix} = N_1 y \quad (5)$$

where $F_c \triangleq \frac{F}{q_a}$ is the forward kinematics function that relates the active joints to the end-effector position and

$$N_1 \triangleq \begin{bmatrix} F_c & 0 & 0\\ 0 & J_c & 0\\ 0 & \dot{J}_c & J_c\end{bmatrix} \in \Re^{3n_c \times 3l} \quad (6)$$

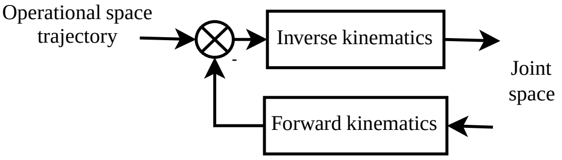


Figure 2: Operational space control of a parallel manipulator.

The above matrix produces large values for small values of q_a. To avoid this situation, a limit is imposed here on the value of each component of q_a so that there is always a valid solution available.

The difference between the reference operational space trajectory and the output of the forward kinematics block is referred to in this paper as system error. Let $\left[\Delta x, \Delta\dot{x}, \Delta\ddot{x}\right]^T$ be this error in operational space. If this error is small, then (2) can be approximated to

$$\Delta x \approx J_c \Delta q_a \quad (7)$$

However, it can be stated, without any approximation, that

$$\begin{aligned} \Delta\dot{x} &= J_c\Delta\dot{q}_a &(8)\\ \Delta\ddot{x} &= \dot{J}_c\Delta\dot{q}_a + J_c\Delta\ddot{q}_a &(9)\end{aligned}$$

It is a common practice that when end-effector trajectory is formulated in operational space, Δx is chosen in (7) such that the approximate movement of the end-effector partially matches the target velocities in (8) (Whitney, 1969). Equation (7) is only valid for a small value of Δx. If the target position is too distant, it is important to bring the target closer. This way, the manipulator reaches its final target in smaller steps. For this reason, Δx needs to be clamped such that

$$clamp(\Delta x, D_{max}) = \begin{cases} \Delta x \text{ if } ||\Delta x|| < D_{max}\\ D_{max}\frac{\Delta x}{||\Delta x||} \text{ otherwise}\end{cases} \quad (10)$$

where $||\bullet||$ is the Euclidean norm. The value of the scalar D_{max} should be at least several times larger than what end-effector moves in a single step and less than half the length of a typical link. This heuristic approach has also been reported to reduce oscillations in the system, which allows the designer to use a smaller value for damping constant. This usually results in a quicker response (Buss and Kim, 2005). To calculate the error in joint space, (7), (8), and (9) can be written as

$$\begin{aligned} \Delta q_a &= J_c^\dagger \Delta x & (11)\\ \Delta \dot{q}_a &= J_c^\dagger \Delta \dot{x} & (12)\\ \Delta \ddot{q}_a &= J_c^\dagger (\Delta \ddot{x} - \dot{J}_c \Delta \dot{q}_a) & \\ &= J_c^\dagger \Delta \ddot{x} - J_c^\dagger \dot{J}_c J_c^\dagger \Delta \dot{x} & (13) \end{aligned}$$

In matrix form, these equations can be written as

$$\begin{bmatrix} \Delta q_a \\ \Delta \dot{q}_a \\ \Delta \ddot{q}_a \end{bmatrix} = \begin{bmatrix} J_c^\dagger & 0 & 0 \\ 0 & J_c^\dagger & 0 \\ 0 & -J_c^\dagger \dot{J}_c J_c^\dagger & J_c^\dagger \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \dot{x} \\ \Delta \ddot{x} \end{bmatrix} \tag{14}$$

or alternatively

$$\Delta y = N_2 \begin{bmatrix} \Delta x \\ \Delta \dot{x} \\ \Delta \ddot{x} \end{bmatrix} \tag{15}$$

where $J_c^\dagger$ is the pseudoinverse of J_c. Pseudoinverse is defined for all matrices including the ones which are not square or are not full rank. It also gives the best solution in terms of least squares. Except near singularities, the pseudoinverse gives a stable solution even in those cases when the target end-effector position doesn't lie in the work volume of the mechanical manipulator. The resulting solution is the closest location to its target which minimizes $||J_c \Delta q - \Delta x||^2$. In the vicinity of singularity, the pseudoinverse creates large changes in joint variables, even for very small changes in the end-effector position, resulting in an unstable system. One important feature of pseudoinverse is that the term $(I - J_c^\dagger J_c)$ projects on the null space of J_c. This feature can be exploited for redundant manipulators. It is possible to generate internal motions in a redundant manipulator, i.e., $\dot{q}_0$, without changing its end-effector position (Sciavicco and Siciliano, 2000). For redundant manipulators, (2) can be written as

$$\dot{x} = J_c \dot{q} + (I - J_c^\dagger J_c)\dot{q}_0 \tag{16}$$

However, in this paper it is assumed that the parallel manipulator is not redundant, i.e., number of active joints is equal to the DOF of the end-effector.

The damped least-squares (DLS) method, which is also referred to the Levenberg-Marquardt method, solves many problems related to pseudoinverse. The method gives a numerically stable solution near singularities, and was first used in inverse kinematics by (Wampler, 1986) and (Nakamura and Hanafusa, 1986). It was also used for theodolite calibration by (Sultan and Wager, 2002).

Not only does DLS minimize the term $||J_c \dot{q}_a - \dot{x}||^2$ but it also minimizes the joint velocities with a damping factor, i.e., $\lambda^2 ||\dot{q}_a||^2$ where $\lambda \in \Re$ and $\lambda \neq 0$. The function to be minimized can be written as

$$\min_{\dot{q}_a} \left\{ ||J_c \dot{q}_a - \dot{x}||^2 + \lambda^2 ||\dot{q}_a||^2 \right\} \tag{17}$$

The DLS solution is equal to (Buss and Kim, 2005)

$$\dot{q}_a = (J_c^T J_c + \lambda^2 I)^{-1} J_c^T \dot{x} \tag{18}$$

or alternatively

$$\dot{q}_c = J_c^T (J_c J_c^T + \lambda^2 I)^{-1} \dot{x} \tag{19}$$

Equation (18) requires an inversion of an $n \times n$ matrix, while (19) requires an inversion of only an $l \times l$ matrix, which is computationally more efficient. In terms of SVD, the singular values change from $\frac{1}{\sigma^i}$ for $J_c^\dagger$ to $\frac{\sigma^2}{\sigma^2+\lambda^2}$ for $(J_c J_c^T + \lambda^2 I)^{-1}$ (Buss and Kim, 2005). If $\sigma_i \to 0, \frac{1}{\sigma_i} \to \infty$, while in the other case, $\frac{\sigma^2}{\sigma^2+\lambda^2} \to \frac{1}{\lambda^2}$ when $\sigma_i \to 0$. Therefore, a stable solution is observed even near singularities for $\forall \lambda : \lambda \neq 0$. Using (19), N_2 can be redefined as

$$N_2 \triangleq \begin{bmatrix} J_c^* & 0 & 0 \\ 0 & J_c^* & 0 \\ 0 & -J_c^* \dot{J}_c J_c^* & J_c^* \end{bmatrix} \in \Re^{3l \times 3n_c} \tag{20}$$

where $J_c^* = J_c^T (J_c J_c^T + \lambda^2 I)^{-1}$. The value of λ is set by the designer. Large values can result in a slower convergence rate and very small values can reduce the effectiveness of the method. In literature, there are many methods proposed to select the value of λ dynamically (Mayorga et al., 1990; Nakamura and Hanafusa, 1986; Chiaverini et al., 1994).

3 FORWARD KINEMATICS FUNCTION

In order to formulate the forward and inverse kinematics matrices, N_1 and N_2, it is important to formulate *analytically* the forward kinematics function of a parallel manipulator. The derivation is somewhat similar to the derivation of the analytic Jacobian of a parallel manipulator by (Dutre et al., 1997), which was derived using the velocity-closure property. The derivation is given as follows;

As all the manipulators are connected to the same end-effector, it can be stated, using the position-closure property, that

$$F_c q_a = F_1 q_1 = F_2 q_2 = \cdots = F_n q_n \tag{21}$$

where q_j is the vector of joint variables of j^{th} manipulator and $F_j(q) \triangleq \frac{F_j(q)}{q_j}$ is the forward kinematics function of the j^{th} manipulator.

Each column of the function F_c corresponds to rotational angle or displacement of an active joint, depending on whether the joint is rotatory or prismatic. Hence

$$F_c^i = F_j q_j^i \tag{22}$$

where $F_c^i \in \Re^{n_a}$ is the i^{th} column of F_c and q_j^i is a vector of joint variables of the j^{th} manipulator when the i^{th} active joint is moved one unit while all the other active joints are locked. If q_c is the vector of all the joint variables, i.e.,

$$q_c = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_{n_c} \end{bmatrix} \in \Re^{n_c} \qquad (23)$$

then (22) can be written as

$$F_c^i = F_j S_j q_c^i \qquad (24)$$

where q_c^i is a vector of all the joints when the i^{th} active joint is moved one unit while all the other active joints are locked and S_j is a selection matrix to select the variables of the j^{th} manipulator, i.e.,

$$S_j = \begin{bmatrix} 0 & \dots & 1 & 0 & \dots & 0 & \dots & 0 \\ 0 & \dots & 0 & 1 & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & 0 & \dots & 1 & \dots & 0 \end{bmatrix} \in \Re^{n_j \times n_c}$$

where n_j is the number of joints in the j^{th} manipulator. Let q_p be the vector of passive joint variables and q_a be the vector of active joint variables such that

$$\begin{aligned} q_p &= S_p q_c & (25) \\ q_a &= S_a q_c & (26) \end{aligned}$$

where $q_p \in \Re^{n_p}$ and $q_a \in \Re^{n_a}$ and S_p and S_a are selection matrices for passive and active joints, respectively. Typical values of S_p and S_a can be written as

$$S_p = \begin{bmatrix} \dots & 0 & 1 & 0 & \dots & 0 & 0 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \dots & 0 & 0 & 0 & \dots & 0 & 1 & 0 & \dots \end{bmatrix} \in \Re^{n_p \times n_c}$$

and

$$S_a = \begin{bmatrix} \dots & 0 & 0 & 0 & \dots & 0 & 1 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \dots & 0 & 1 & 0 & \dots & 0 & 0 & 0 & \dots \end{bmatrix} \in \Re^{n_a \times n_c}$$

Both of these matrices are sparse and orthogonal, i.e., $S_p S_p^T = I$ and $S_a S_a^T = I$, which implies

$$\begin{aligned} q_{c_p} &= S_p^T q_p & (27) \\ q_{c_a} &= S_a^T q_a & (28) \end{aligned}$$

where q_{c_p} is equivalent to q_c except that the active joints are set to zero and similarly, q_{c_a} is equivalent to q_c except that the passive joints are set to zero such that

$$q_c = q_{c_p} + q_{c_a} \qquad (29)$$

Substituting (27) and (28) in (29) yields

$$q_c = S_p^T q_p + S_a^T q_a \qquad (30)$$

In reference to the position-closure property (21), let

$$A q_c = 0 \qquad (31)$$

where

$$A = \begin{bmatrix} \frac{F_1}{q_1} & -\frac{F_2}{q_2} & 0 & \dots & 0 \\ \frac{F_1}{q_1} & 0 & -\frac{F_3}{q_3} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{F_1}{q_1} & 0 & 0 & \dots & -\frac{F_n}{q_n} \end{bmatrix} \in \Re^{n_a(n-1) \times n_c} \qquad (32)$$

Substituting the value of q_c from (30) gives

$$\begin{aligned} A q_c &= A(q_{c_p} + q_{c_a}) \\ &= A S_p^T q_p + A S_a^T q_a \\ &= A_p q_p + A_a q_a & (33) \end{aligned}$$

Applying (31)

$$q_p = -A_p^{\dagger} A_a q_a \qquad (34)$$

Substituting this expression in (30) yields

$$q_c = S_a^T q_a - S_p^T A_p^{\dagger} A_a q_a \qquad (35)$$

As q_c^i is defined for a unit displacement of the i^{th} active joint, hence, q_a can be replaced with a column of S_a which corresponds to the i^{th} active joint, denoted by $(S_a)^i$, to evaluate q_c^i, i.e.,

$$q_c^i = S_a^T (S_a)^i - S_p^T A_p^{\dagger} A_a (S_a)^i \qquad (36)$$

Substituting the above value in (24) gives

$$F_c^i = F_j S_j q_c^i \qquad (37)$$

or

$$F_c = F_j S_j \begin{bmatrix} q_c^1 & q_c^2 & \dots & q_c^{n_a} \end{bmatrix} \qquad (38)$$

4 ANALYTICAL JACOBIAN AND ITS DERIVATIVE

(Dutre et al., 1997) evaluated the analytical Jacobian for a parallel manipulator using the velocity-closure property. The Jacobian can also be derived by replacing F_c in (38) by J_c and q_c^i by $\dot{q}_c^i$, i.e.,

$$J_c = J_j S_j \begin{bmatrix} \dot{q}_c^1 & \dot{q}_c^2 & \dots & \dot{q}_c^{n_a} \end{bmatrix} \qquad (39)$$

where J_c is the analytical Jacobian that relates the velocities of the active joints to the end-effector velocity. J_j and $\dot{q}_j$ are the Jacobian and the vector of joint velocities of the j^{th} manipulator, respectively. $\dot{q}_c^i$ can be stated using (36) as follows;

$$\dot{q}_c^i = S_a^T (S_a)^i - S_p^T B_p^{\dagger} B_a (S_a)^i \qquad (40)$$

where $B_p = BS_p^T$, $B_a = BS_a^T$, and

$$B = \begin{bmatrix} J_1 & -J_2 & 0 & \dots & 0 \\ J_1 & 0 & -J_3 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ J_1 & 0 & 0 & 0 & -J_n \end{bmatrix} \in \Re^{n_a(n-1)\times n_c} \tag{41}$$

Using B, the velocity-closure property of a parallel manipulator can be written as

$$B\dot{q}_c = 0$$

The derivative of the closed-loop Jacobian (J_c) given in (39) is

$$\dot{J}_c = \dot{J}_j S_j \begin{bmatrix} \ddot{q}_c^1 & \ddot{q}_c^2 & \dots & \ddot{q}_c^{n_a} \end{bmatrix} \tag{42}$$

where $\ddot{q}_c^i$ can be calculated by differentiating (40), i.e.,

$$\ddot{q}_c^i = -S_p^T \left(\frac{\partial B_p^\dagger}{\partial q_i} B_a + B_p^\dagger \frac{\partial B_a}{\partial q_i} \right) (S_a)^i \tag{43}$$

where q_i is the i^{th} driving joint.

The time derivative of a Jacobian column for a serial manipulator is the sum of the partial derivatives of this column with respect to joint variables, multiplied by the time-derivates of these variables (Bruyninckx and De Schutter, 1996). As such, time-derivative of the i^{th} column of the Jacobian is given by

$$\dot{J}^i = \sum_{j=1}^{n} \frac{\partial J^i(q)}{\partial q^j} \frac{\partial q^j}{\partial t} = \sum_{j=1}^{n} \frac{\partial J^i(q)}{\partial q^j} \dot{q}^j \tag{44}$$

Similarly, the derivative of the Jacobian of each manipulator of a parallel manipulator can be expressed using (44), i.e.,

$$\dot{J}_j = \sum_{i=1}^{n_a} \frac{\partial J_j}{\partial q_i} \dot{q}_i = \sum_{i=1}^{n_a} \left(\sum_{k=1}^{n_j} \frac{\partial J_j}{\partial q_{j,k}} \frac{\partial q_{j,k}}{\partial q_i} \right) \dot{q}_i \tag{45}$$

where k is a joint of the j^{th} manipulator and $\frac{\partial J_j}{\partial q_{j,k}}$ represents Jacobian derivative of the j^{th} serial manipulator. The factor, $\frac{\partial q_{j,k}}{\partial q_i}$ in (45), is the k^{th} component in $S_j \dot{q}_c^i$.

5 PROPOSED CONTROL FRAMEWORK

Figure 3 shows the structure of the proposed control framework for parallel manipulators. As only active joints are actuated, it is important to incorporate the contribution of the passive joints onto the active joints. If joint friction is ignored, the relationship between the torque of active joints and passive joints is given by the following equation (Cheng et al., 2003);

$$\tau_c = \tau_a + \left(\frac{\partial q_p}{\partial q_a} \right)^T \tau_p \tag{46}$$

where $\tau_p \in \Re^{n_p}$ is the torque measured from strain gauges on passive joints, $\tau_a \in \Re^{n_a}$ is the torque produced by the actuators on active joints, and $\tau_c \in \Re^{n_a}$ is the torque measured by strain gauges mounted on active joints. From (Dutre et al., 1997), it can be inferred that

$$\frac{\partial q_p}{\partial q_a} = B_p^\dagger B_a$$

Using the above value in (46) yields

$$\tau_c = \tau_a - (B_p^\dagger B_a)^T \tau_p$$

or

$$\tau_c = \tau_a - B_a^T (B_p^\dagger)^T \tau_p \tag{47}$$

The passive joints project torque onto the active joints with a factor of $-B_a^T (B_p^\dagger)^T$. This will be used as the exogenous force disturbance signal in the hybrid controller, as shown in Figure 3.

To ease the implementation of the *clamp* block, it can be taken out of the closed loop. This can be done by redefining r_k using

$$r_k = clamp(r_{original} - N_1 y_k) + N_1 y_k \tag{48}$$

6 EXAMPLE

As the proposed kinematics framework is evaluated analytically, it can be applied on any non-redundant parallel manipulator. However, in this section, for the sake of demonstration, a simple case of a 3-R$\underline{\text{P}}$R robot is presented, shown in Figure 1.

The forward kinematics function for the first manipulator can be stated as

$$F_1 = \begin{bmatrix} (x_{1,1} + q_{1,2} + x_{1,2})\cos(q_{1,1}) + x_{1,3}\cos(q_{1,1} + q_{1,3}) \\ (x_{1,1} + q_{1,2} + x_{1,2})\sin(q_{1,1}) + x_{1,3}\sin(q_{1,1} + q_{1,3}) \\ q_{1,1} + q_{1,3} \end{bmatrix} \tag{49}$$

where $x_{1,2}$ and $q_{1,2}$ denote the length of the second link and the second joint variable, respectively. The expressions for other links can be written in the same way. Using this kinematic model for the values given in Table 1, the end-effector position was found to be at

$$x_{end_c} = \begin{bmatrix} 1.5 \\ 2.5981 \\ 1.0472 \end{bmatrix}$$

where the first two elements represent the position in $x - y$ plane and the third element represents the angular rotation of the end-effector.

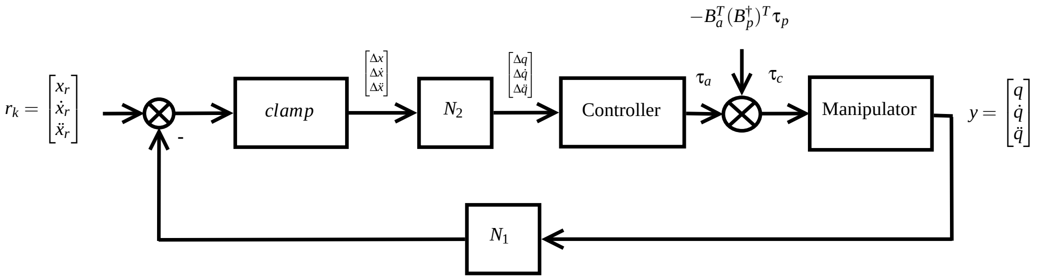

$-B_a^T(B_p^\dagger)^T\tau_p$
$r_k = \begin{bmatrix} x_r \\ \dot{x}_r \\ \ddot{x}_r \end{bmatrix}$
-
clamp
$\begin{bmatrix} \Delta x \\ \Delta \dot{x} \\ \Delta \ddot{x} \end{bmatrix}$
N_2
$\begin{bmatrix} \Delta q \\ \Delta \dot{q} \\ \Delta \ddot{q} \end{bmatrix}$
Controller
τ_a
τ_c
Manipulator
$y = \begin{bmatrix} q \\ \dot{q} \\ \ddot{q} \end{bmatrix}$
N_1

Figure 3: Operational space control of a parallel manipulator.

Table 1: Assumed values for a 3-R$\underline{P}$R robot.

Manipulator 1	Manipulator 2	Manipulator 3
$q_{1,1} = \pi/3$	$q_{2,1} = 2\pi/3$	$q_{3,1} = 4\pi/3$
$q_{1,2} = 1$	$q_{2,2} = 1$	$q_{3,2} = 1$
$q_{1,3} = 0$	$q_{2,3} = -\pi/3$	$q_{3,3} = -\pi$
$x_{1,1} = 0.5$	$x_{2,1} = 0.5$	$x_{3,1} = 0.5$
$x_{1,2} = 0.5$	$x_{2,2} = 0.5$	$x_{3,2} = 0.5$
$x_{1,3} = 1$	$x_{2,3} = 1$	$x_{3,3} = 1$

The forward kinematics function, F_c, gives the following end-effector position for the active joints $[1,1,1]^T$;

$$x_{end} = \begin{bmatrix} 1.498 \\ 2.597 \\ 1.048 \end{bmatrix}$$

7 CONCLUSIONS

Similar to the analytical Jacobian for a parallel manipulator, which is a function of joint variables and relates the velocity of the active joints to the velocity of the end-effector, the analytical forward kinematics function is also a function of the joint variables that relates the position of the active joints to the position of the end-effector. The generality of the proposed technique allows the forward kinematics function to be used in a variety of applications. A control configuration is also described in this paper as a prospective application of the proposed technique.

REFERENCES

Bhattacharya, S., Hatwal, H., and Ghosh, A. (1997). An on-line parameter estimation scheme for generalized Stewart platform type parallel manipulators. *Mechanism and Machine Theory*, 32(1):79–89.

Bruyninckx, H. and De Schutter, J. (1996). Symbolic differentiation of the velocity mapping for a serial kinematic chain. *Mechanism and Machine Theory*, 31(2):135–148.

Buss, S. and Kim, J. (2005). Selectively Damped Least Squares for Inverse Kinematics. *Journal of Graphics Tools*, 10(3):37–49.

Cheng, H., Yiu, Y.-K., and Li, Z. (2003). Dynamics and control of redundantly actuated parallel manipulators. *IEEE/ASME Transactions on Mechatronics*, 8(4):483–491.

Chiaverini, S., Siciliano, B., and Egeland, O. (1994). Review of the damped least-squares inverse kinematics with experiments on an industrial robot manipulator. *IEEE Transactions on Control Systems Technology*, 2(2):123–134.

Collins, C. (2002). Forward kinematics of planar parallel manipulators in the Clifford algebra of P2. *Mechanism and Machine Theory*, 37(8):799–813.

Dutre, S., Bruyninckx, H., and De Schutter, J. (1997). The analytical jacobian and its derivative for a parallel manipulator. In *IEEE International Conference on Robotics and Automation*, volume 4, pages 2961–2966.

Kim, D., Chung, W., and Youm, Y. (2000). Analytic jacobian of in-parallel manipulators. In *Proceedings of the IEEE International Conference on Robotics and Automation ICRA '00*, volume 3, pages 2376–2381.

Kong, X. (2008). *Advances in Robot Kinematics: Analysis and Design*, chapter Forward Kinematics and Singularity Analysis of a 3-R$\underline{P}$R Planar Parallel Manipulator, pages 29–38. Springer Netherlands.

Mayorga, R., Milano, N., and Wong, A. (1990). A simple bound for the appropriate pseudoinverse perturbation of robot manipulators. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, pages 1485–1488.

Murray, A., Pierrot, F., Dauchez, P., and McCarthy, J. (1997). A planar quaternion approach to the kinematic synthesis of a parallel manipulator. *Robotica*, 15(04):361–365.

Nakamura, Y. and Hanafusa, H. (1986). Inverse kinematic solutions with singularity robustness for robot manipulator control. *Journal of dynamic systems, measurement, and control*, 108(3):163–171.

Sciavicco, L. and Siciliano, B. (2000). *Modelling and Control of Robot Manipulators*. Springer, second edition.

Siciliano, B. and Khatib, O. (2007). *Springer Handbook of Robotics*. Springer-Verlag, Secaucus, NJ, USA.

Sultan, I. and Wager, J. (2002). Simplified theodolite calibration for robot metrology. *Advanced Robotics*, 16(7):653–671.

Timmons, K. and Ringelberg, J. (2008). Approach and Capture for Autonomous Rendezvous and Docking. In *2008 IEEE Aerospace Conference*, pages 1–6.

Wampler, C. W. (1986). Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods. *IEEE Transactions on Systems, Man, and Cybernetics*, 16(1):93–101.

Warnecke, H., Neugebauer, R., and Wieland, F. (1998). Development of Hexapod Based Machine Tool. *CIRP Annals-Manufacturing Technology*, 47(1):337–340.

Wenger, P., Chablat, D., and Zein, M. (2007). Degeneracy Study of the Forward Kinematics of Planar 3-R$\underline{\text{P}}$R Parallel Manipulators. *Journal of Mechanical Design*, 129:1265.

Whitney, D. E. (1969). Resolved motion rate control of manipulators and human prostheses. *IEEE Transactions on Man Machine Systems*, 10(2):47–53.

Yamane, K., Nakamura, Y., Okada, M., Komine, N., and Yoshimoto, K. (2005). Parallel Dynamics Computation and $\mathscr{H}_\infty$ Acceleration Control of Parallel Manipulators for Acceleration Display. *Journal of Dynamic Systems, Measurement, and Control*, 127:185.

PARAMETER IDENTIFICATION OF A HYBRID REDUNDANT ROBOT BY USING DIFFERENTIAL EVOLUTION ALGORITHM

Yongbo Wang, Huapeng Wu and Heikki Handroos
Department of Mechanical Engineering, IMVE, Lappeenranta University of Technology, Lappeenranta, Finland
{Yongbo.wang, huapeng.wu, heikki.handroos}@lut.fi

Keywords: Calibration, Parameter Identification, Parallel Robot, Differential Evolution.

Abstract: In this paper, a hybrid redundant robot IWR (Intersector Welding Robot) which possesses ten degrees of freedom (DOF) where 6-DOF in parallel and additional redundant 4-DOF in serial is proposed. To improve the accuracy of the robot, the kinematic errors caused by the manufacturing and assembly processes have to be compensated or limited to a minimum value. However, currently, there is no effective instrument which capable of measuring the symmetrical errors of the corresponding joints and link lengths after the structure has been assembled. Therefore, calibration and identification of these unknown parameters is utmost important and necessary to the systematic accuracy. This paper presents a calibration method for identifying the unknown parameters by using differential evolution (DE) algorithm, which has proven to be an efficient, effective and robust optimization method to solve the global optimization problems. The DE algorithm will guarantee the fast convergence and accurate solutions regardless of the initial conditions of the parameters. Based on the inverse kinematic error model of the robot, the simulation of the actual robot is achieved by introducing random geometric errors and measurement poses which representing their relative physical behavior. Moreover, through computer simulation, the validity and effectiveness of the DE algorithm for the parameter identification of the proposed application has also been examined.

1 INTRODUCTION

It is widely believed that parallel robot has high stiffness, low inertia, high speed and accuracy but small workspace compared to its counterpart serial robot. To take advantage of the benefits (bigger workspace and higher stiffness) of both types of robotic structures, a compromised hybrid redundant robot which can be used to perform the welding, machining and remote handling is developed in Lappeenranta University of Technology (Wu, 2005). In order to satisfy the required accuracy of the robot, the calibration and identification of the real structure parameters is essential and necessary. Generally, calibration can be classified into two types: static and dynamic. The static or kinematic calibration is an identification of those parameters which influence primarily the static positioning characteristics of a robot, such as the errors caused by length of the links and joints. Whereas the dynamic calibration is used to identify parameters influencing primarily motion characteristics, such as the deflection of mechanisms caused by temperature, and the compliances of joints and links. This paper will be concentrated on the static calibration to identify the geometric parameters of the proposed hybrid redundant robot. At present, there exist two kinds of static or kinematic calibration methods, one is self or autonomous calibration method based on inner information or restrictions of the kinematic parameters of joints (Ryu, 2001; Zhuang, 1996; Khalil, 1999; Zhuang, 2000; Ecorchard, 2005), and another is exterior or classical calibration method by using accurate instruments to measure the pose of the moving platform directly (Gao, 2003; Besnard, 1999; Prenaud, 2003). For these calibration methods, most of them are focused on the kinematic calibration and parameter identification of the pure-serial or pure-parallel mechanisms. Moreover, many calibration models are based on the identification Jacobian matrix which formulates a linear relationship between measurement residuals and kinematic parameter errors, then the parameter errors are evaluated by using least square algorithm. However, this kind of method is subject to break down in the vicinity of singular robot configurations due to the iterative inversion of the robot Jacobian (Zhong, 1996). Instead of the Jacobian matrix based calibration approaches, the non-parametric calibration method was introduced by Shamma and Whitney (Shamma, 1987), in which the actual kinematic parameters which drive the robot to minimize the end-effector deviations can be found by using non-linear least-square optimization without explicit evaluation of the Jacobian. Based on the non-parametric calibration method, some evolutionary computing algorithms, such as genetic algorithm (GA) (Liu, 2007; Zhuang, 1996), artificial neural networks (NN) (Zhong,1996) and genetic programming (GP) (Dolinsky, 2007), have been successfully employed to calibrate serial or parallel robot. Differential evolution (DE) is a simple but effective evolutionary algorithm for solving non-linear, global optimization problems. It has demonstrated superior performance in both widely used benchmark functions (Vesterstrom, 2004) and practical applications (Wu, 2000). In this work, based on the static and non-parametric calibration method, DE will be adopted to identify the real kinematic parameters of the proposed hybrid redundant robot.

The paper is organized into five main sections. The first section serves as an introduction. The second section reviews the kinematic model of the proposed robot, which includes the inverse kinematic equations and the error models of the robot. Section 3 presents the calibration equations and the implement of DE optimization method. Simulation results are presented in section 4, and conclusions are drawn in section 5.

2 IDENTIFICATION MODELS

The kinematics of the proposed hybrid robot as shown in Fig.1 is a combination of a multi-link serial mechanism (here named as Carriage) and a standard Stewart parallel manipulator (here named as Hexa-WH). To simplify its analysis, the two parts will be first carried out separately, and then combined them together to obtain the final solutions. According to Shamma and Whitney (Shamma, 1987), the calibration also can be classified into forward calibration and inverse calibration. Forward calibration involves finding the actual location in the world space for a given joint configuration, while inverse calibration involves finding exact joint values for given locations in the world space. As we all know that the inverse kinematics of the parallel robot is simple than forward kinematics and vice versa for the serial robot, so we decide to identify the kinematic parameters of the parallel part based on inverse calibration method and the serial part based on forward solutions

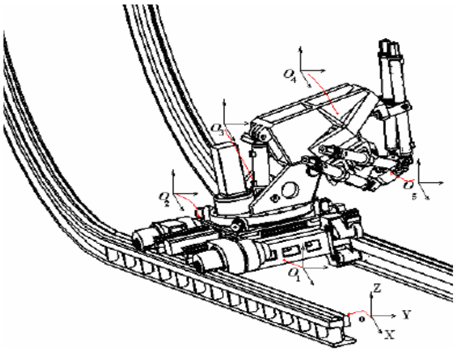


Figure 1: 3D model of IWR.

2.1 Forward Kinematics

To study the kinematics of the serial multi-link mechanisms, the convention of Denavit-Hartenberg (Craig, 1986) is commonly adopted. Based on this convention, the principle of the 4-DOF Carriage mechanism can be established as shown in Fig.2, which provides four degrees of freedom to the transient end-effector (O_4), including two translational movements and two rotational movements.

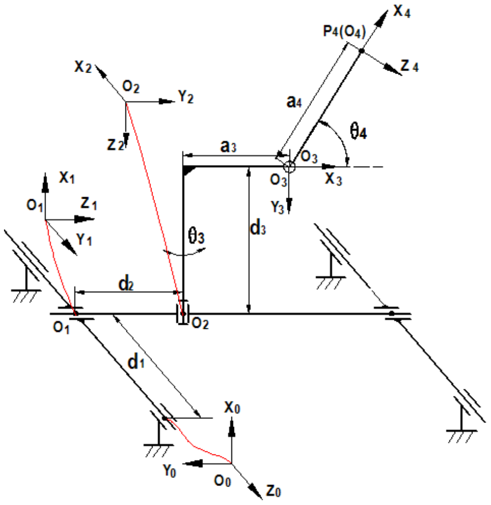


Figure 2: Coordinate system of Carriage.

Using the coordinate systems established in Fig. 2, the corresponding link parameters are given in Table1. Substituting the D-H link parameters into (1), we can obtain the D-H homogeneous transformation matrices ${}^0\mathbf{A}_1, {}^1\mathbf{A}_2, {}^2\mathbf{A}_3, {}^3\mathbf{A}_4$.

Table 1: Nominal DH parameters of Carriage.

Joint	α_i	a_i	d_i	θ_i
1	$\pi/2$	0	d_1	0
2	$\pi/2$	0	d_2	$\pi/2$
3	$\pi/2$	a_3	d_3	θ_3
4	$-\pi/2$	a_4	0	θ_4

$$^{i-1}\mathbf{A}_i = \begin{bmatrix} c\theta_i & -c\alpha_i s\theta_i & s\alpha_i s\theta_i & a_i c\theta_i \\ s\theta_i & c\alpha_i c\theta_i & -s\alpha_i c\theta_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

where $c\theta_i$ denotes $\cos\theta_i$, and $s\theta_i$ denotes $\sin\theta_i$.

The resulting homogeneous transformation matrix, i.e. the forward kinematics of the Carriage, can be obtained by multiplying the matrices of $^0\mathbf{A}_1, {}^1\mathbf{A}_2, {}^2\mathbf{A}_3$ and $^3\mathbf{A}_4$

$$^0\mathbf{A}_4 = {}^0\mathbf{A}_1 {}^1\mathbf{A}_2 {}^2\mathbf{A}_3 {}^3\mathbf{A}_4 \\ = \begin{bmatrix} s\theta_4 & 0 & c\theta_4 & a_1 + d_3 + a_4 s\theta_4 \\ -s\theta_3 c\theta_4 & -c\theta_3 & s\theta_3 s\theta_4 & -d_2 - a_3 s\theta_3 - a_4 s\theta_3 c\theta_4 \\ c\theta_3 c\theta_4 & -s\theta_3 & -c\theta_3 s\theta_4 & d_1 + a_3 c\theta_3 + a_4 c\theta_3 c\theta_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

From (2) we can get the rotation matrix and position vector of the frame {4} with respect to frame {0} as follows:

$$^0\mathbf{R}_4 = \begin{bmatrix} s\theta_4 & 0 & c\theta_4 \\ -s\theta_3 c\theta_4 & -c\theta_3 & s\theta_3 s\theta_4 \\ c\theta_3 c\theta_4 & -s\theta_3 & -c\theta_3 s\theta_4 \end{bmatrix} \quad (3)$$

$$^0\mathbf{P}_4 = \begin{bmatrix} a_1 + d_3 + a_4 s\theta_4 \\ -d_2 - a_3 s\theta_3 - a_4 s\theta_3 c\theta_4 \\ d_1 + a_3 c\theta_3 + a_4 c\theta_3 c\theta_4 \end{bmatrix} \quad (4)$$

In reality, the above D-H parameters will deviate from their nominal values because of the manufacturing and assembly errors. Since each joint provides four parameters, therefore, the four links will produce 16 identified parameters for the robot.

2.2 Inverse Kinematics of Hexa-WH

Fig. 3 shows a schematic diagram of hexapod parallel mechanism, for the purpose of analysis, two Cartesian coordinate systems, frames O$_4$(X$_4$, Y$_4$, Z$_4$) and O$_5$(X$_5$, Y$_5$, Z$_5$) are attached to the base plate and the end-effector, respectively. Six variable limbs are connected with the base plate by Universal joints and the task platform by Spherical joints.

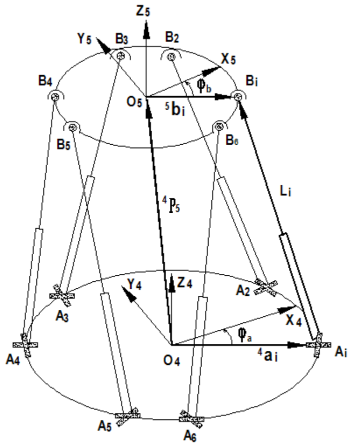


Figure 3: Norminal model of the Hexapod parallel mechanism.

For the designed kinematics parameters, let $\mathbf{l}_i$ be the unit vector in the direction of $\mathbf{A}_i\mathbf{B}_i$, and l_i denote the magnitude of the leg vector $\mathbf{A}_i\mathbf{B}_i$, then the following vector-loop equation will represent the inverse kinematics of the *ith* limb of the manipulator.

$$l_i\mathbf{l}_i = {}^4\mathbf{P}_5 + {}^4\mathbf{R}_5 {}^5\mathbf{b}_i - {}^4\mathbf{a}_i \qquad (i = 1,2,\cdots 6) \quad (5)$$

where $^4\mathbf{P}_5$ denotes the position vector of the task frame {5} with respect to the base frame {4}, and $^4\mathbf{R}_5$ is the Z-Y-X Euler transformation matrix expressing the orientation of the frame {5} relative to the frame {4},

$$^4\mathbf{R}_5 = \begin{bmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma \\ s\alpha s\beta & s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma \\ -s\beta & c\beta s\gamma & c\beta c\lambda \end{bmatrix} \quad (6)$$

and the $^4\mathbf{a}_i$, $^5\mathbf{b}_i$ represent the position vectors of U-joints $\mathbf{A}_i$ $\mathbf{A}_i$ and S-joints $\mathbf{B}_i$ in the coordinate frames {4} and {5} respectively. In practice, due to the manufacturing and assembly errors, the coordinate $^4\mathbf{a}_i$ and $^5\mathbf{b}_i$ will deviate from their nominal values and l_i will also have an initial offset, altogether there will be 42 identified parameters provided by Hexa-WH.

2.3 Kinematics and Identified Error Model of the Hybrid Manipulator

The schematic diagram of the redundant hybrid manipulator is shown in Fig. 4, which consists of

Carriage and Hexapod manipulator as mentioned above. The base plate frame {4} of Hexa-WH is coincided with the end task frame of Carriage. The global base frame {0} is located at the left rail of Carriage.

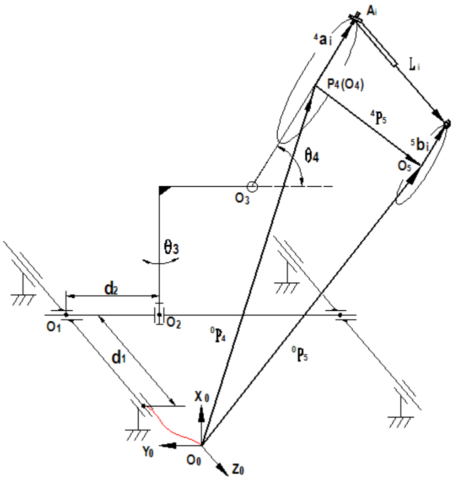


Figure 4: Schematic diagram of IWR.

According to the geometry, a vector-loop equation can be derived:

$$\begin{aligned} {}^0\mathbf{P}_5 &= {}^0\mathbf{P}_4 + {}^0\mathbf{R}_4\,{}^4\mathbf{P}_5 = {}^0\mathbf{P}_4 + {}^0\mathbf{R}_4\left(l_i\mathbf{l}_i + {}^4\mathbf{a}_i - {}^4\mathbf{R}_5\,{}^5\mathbf{b}_i\right) \\ &= {}^0\mathbf{P}_4 + {}^0\mathbf{R}_4 l_i\mathbf{l}_i + {}^0\mathbf{R}_4\,{}^4\mathbf{a}_i - {}^0\mathbf{R}_5\,{}^5\mathbf{b}_i \end{aligned} \quad (7)$$

From (7), we can obtain the nominal leg length, i.e. the inverse solution of the robot as:

$$l_i\mathbf{l}_i = \left({}^0\mathbf{R}_4\right)^{-1}\left({}^0\mathbf{P}_5 - {}^0\mathbf{P}_4 - {}^0\mathbf{R}_4\,{}^4\mathbf{a}_i + {}^0\mathbf{R}_5\,{}^5\mathbf{b}_i\right) \quad (8)$$

where ${}^0\mathbf{P}_5$ and ${}^0\mathbf{R}_5$ is the position vector and rotation matrix of the task frame {5} (or end-effector) with respect to the fixed base frame {0}.

Let l_i^m represent the whole leg length which made up of the measured leg length with the inner sensor and the fixed initial leg length offset. Therefore, if parameter errors are not be taken into account, there is the following relation.

$$l_i = l_i^m \quad (9)$$

As a matter of fact, since geometrical errors and other error sources exist, two sides of (9) will never be equal, even if their geometrical parameters are properly corrected. Consequently, if we get enough measurement point data from the inner sensors of the parallel Hexa-WH legs and the Carriage actuators, then our identified kinematic error model can be expressed as an optimization function given as follows:

$$\min \quad Fun\left(\delta\boldsymbol{\alpha}_c, \delta\mathbf{d}_c, \delta\mathbf{a}_c, \delta\boldsymbol{\theta}_c, \delta\mathbf{a}_h, \delta\mathbf{b}_h, \delta\mathbf{l}_h\right) = \sum_{i=1}^{N}\sum_{j=1}^{6}(l_{i,j} - l_{i,j}^m)^2 \quad (10)$$

where $\delta\boldsymbol{\alpha}_c, \delta\mathbf{d}_c, \delta\mathbf{a}_c, \delta\boldsymbol{\theta}_c, \delta\mathbf{a}_h, \delta\mathbf{b}_h, \delta\mathbf{l}_h$ denote the 58 identified parameter vectors, among which 16 parameters are from Carriage and 42 parameters from Hexa-WH. N is measurement number, $l_{i,j}$ and $l_{i,j}^m$ respectively represent the calculated value and measured value of the jth leg in the ith measurement point.

3 DIFFERENTIAL EVOLUTION

Differential Evolution (DE), which introduced by Price and Storn (Storn, 2005), has been proven to be a promising candidate for minimizing real-valued, non-linear and multi-modal objective functions. It belongs to the class of evolutionary algorithms and utilizes the same steps as Genetic Algorithm, i.e. mutation, crossover and selection. Individuals in DE are represented by D-dimensional vectors $\mathbf{x}_{i,G}$, $\forall i \in \{1, 2, \cdots, NP\}$, where D is the number of optimization parameters and NP is the population size. There are several variants or strategies of DE, but the DE scheme which classified by notation DE/rand/1/bin is the most commonly used one. The optimization process of this classical DE can be summarized as follows:

3.1 Initialization

To establish a starting point for the optimization process, an initial population must be created. Typically, each decision parameter in every vector of the initial population is assigned by a randomly chosen value from its feasible bounds:

$$x_{j,i,G=0} = x_{j,i}^L + rand_j[0,1) \cdot (x_{j,i}^U - x_{j,i}^L) \quad (11)$$

where $j = 1, 2, \cdots, D$ is parameter index, and $i = 1, 2, \cdots, NP$ is population index, $x_{j,i}^L$ and $x_{j,i}^U$ are the lower and upper bound of the jth decision parameter, respectively. After the initial population has been created, it evolves through the following operations of mutation, crossover and selection until the terminal condition satisfied.

3.2 Mutation

For each vector $\mathbf{x}_{i,G}$, a mutant vector $\mathbf{m}_{i,G}$ is generated according to

$$\mathbf{m}_{i,G} = \mathbf{x}_{r1,G} + F \cdot (\mathbf{x}_{r2,G} - \mathbf{x}_{r3,G}) \qquad (12)$$

where randomly selected integers $r_1, r_2, r_3 \in \{1,2,\cdots NP\}, r_1 \neq r_2 \neq r_3 \neq i$, and mutation scale factor $F > 0$.

3.3 Crossover

The trial vector is generated as follows:

$$\mathbf{u}_{i,G+1} = (u_{1,i,G+1}, u_{2,i,G+1}, \cdots, u_{D,i,G+1})$$

$$u_{j,i,G+1} = \begin{cases} m_{j,i,G+1} & if\ (rand_j[0,1) < CR \vee j = j_r) \\ x_{j,i,G+1} & otherwise, \end{cases} \qquad (13)$$

where $G = 1,2,\cdots,G_{\max}$ denotes generation index, the index j_r is chosen randomly from the set $\{1,2,\cdots,D\}$, which is used to ensure that vector $\mathbf{u}_{j,i,G+1}$ gets at least one parameter from $\mathbf{m}_{i,G}$, and CR is known as a crossover rate constant which is a user-defined parameter within the range $[0,1]$.

3.4 Selection

To decide whether or not the trail vector should become a member of the next generation, the trail vector $\mathbf{u}_{i,G+1}$ is compared to the target vector $\mathbf{x}_{i,G}$ by evaluating the cost or objective function. A vector with a minimum value of cost function will be allowed to advance to the next generation. That is,

$$\mathbf{x}_{i,G+1} = \begin{cases} \mathbf{u}_{i,G+1} & if \quad fun(\mathbf{u}_{i,G+1}) \leq fun(\mathbf{x}_{i,G}), \\ \mathbf{x}_{i,G} & otherwise, \end{cases} \qquad (14)$$

Using this selection procedure, all individuals of the next generation are as good as or better than the individuals of the current population.

4 SIMULATION RESULTS

To simulate the above process, we randomly generate 100 measurement poses within the robot workspace to form the measured input values. As stated above, we can take (10) as our fitness function, among which, we assume a set of fixed geometric errors for the identified parameter to represent the actual measurement values of the robot, and at the same time suppose these error parameters to be our simulation variables. Through enough evolution generations, the simulated identification parameter will finally approximate to the assumed parameter errors. Table 2 shows the constant parameters we have chosen and the best objective function values of each generation are plotted in Fig. 5.

Table 2: Parameters of DE.

Symbol	Parameter	Value
D	Number of parameters (Variables)	58
NP	Number of population	600
F	Scale or difference factor	0.9
CR	Crossover control constant	1.0
N	Measurement number	100
$G_{\max}$	The maximum generations	60000
$x_{j,i}^L$	Lower bound of identified error parameters	-0.5
$x_{j,i}^U$	Upper bound of identified error parameters	0.5

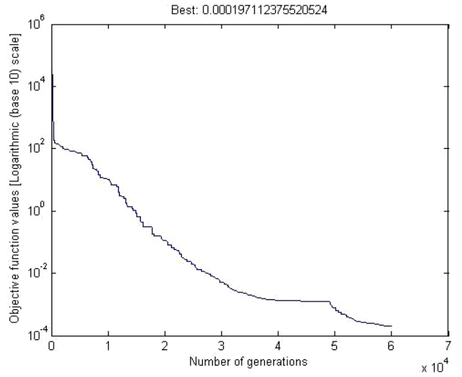


Figure 5: Best objective function values of 60000 generations.

From the above tables and the figure of evolutionary process, we can see that the objective function values decrease dramatically at the beginning, but with the advance of evolution process, they tend to be calm and the convergence speed also become slow. After 60000 generations, most of the identified errors are approximated to the assumed errors, and the final best object function value reach to the accuracy of 10^{-4}. Of course, if we increase the maximum generation number and add more measured poses, then the identification accuracy will be improved and the identified parameters will infinitely approach to the actual values.

5 CONCLUSIONS

In this paper, a hybrid redundant robot used for both machining and assembling of Vacuum Vessel of ITER is introduced. Furthermore, a parameter identification model which has the ability to account for the static error sources is derived. Due to the redundant freedom of the robot, we first divide the robot into two parts according to its mechanism, then formulate the parameter identification model respectively, and finally combine them together to get the final optimization identification model. Based on the DE algorithm and the derived identification model, the 58 kinematic error parameters of the robot were identified by computer simulations. According to the simulation results, we can see that DE has a very strong stochastic searching ability, which is reliable and can be easily used to identify the high non-linear kinematic error parameter models.

REFERENCES

Ecorchard, G.; Maurine, P, 2005. Self- calibratrion of delta parallel robots with elastic deformation compensation, *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2-6 Aug. 2005, pp, 1283 – 1288.*

Gao, Meng, Li Tiemin, Yin Wensheng, 2003. Calibration method and expriment of stewart platform using a laser tracker, *IEEE International Conference on Systems, Man and Cybernetics, Vol. 3, 5-8 Oct. 2003 pp. 2797 – 2802.*

Hanqi Zhuang, Jie Wu et al., 1996. Optimal Planning of Robot Calibration Experiments by Genetic Algorithms, *Proceedings of the 1996 IEEE International Conference on Robotics and Automation, Minneapolis, Minnesota, April 1996, pp. 981-986.*

Hanqi Zhuang, Lixin Liu, 1996. Self-calibration of a class of parallel manipulators, *Proceedings of the 1996 IEEE international conference on robotics and automation, Minneapolis, Minnesota, April 1996,pp.994-999.*

Hanqi Zhuang, Lixin Liu, Oren Masory, 2000. Autonomous calibration of hexapod machine tools, *ASME transactions on Journal of Manufacturing Science and Engineering, February 2000, Vol. 122, pp. 140-148.*

Huapeng Wu, Heikki Handroos, 2000. Utilization of differential evolution in inverse kinematics solution of a parallel redundant manipulator, *IEEE conference on knowledge-based intelligent engineering systems & allied technologies, 30 Aug. – 1 Sept. 2000, Brighton, UK, pp.812-815.*

Huapeng Wu, Heikki Handroos et al. 2005. Development and control towards a parallel water hydraulic weld/cut for machining processes in ITER vacuum vesse. *Int. J. fusion Engineering and Design, Vol. 75-79, pp. 625-631.*

J. Craig, 1986. *Introduction to Robotics: Mechanics and Control* , Addison-Wesley Publishing Co., Reading, MA.

J.Vesterstrom, R.Thomsen, 2004. A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems, *Evolutionary computation, Vol. 2. pp. 1980-1987.*

J.U. Dolinsky, I.D.Jenkinson et al., 2007. Application of Genetic Programming to the Calibration of Industrial Robots, *Journal of Computers in Industry, 58 (2007), pp. 255-264.*

Jeha Ryu, Abdul Rauf, 2001. A new method for fully autonomous calibration of parallel manipulators using a constraint link. *2001 IEEE/ASME international Conference on advanced intelligent mechatronics proceedings 8-12 July, 2001 Como, Italy, pp. 141-146.*

Prenaud and N. Andreff et al, 2003. Vision-based kinematic calibration of a H4 parallel mechanism, Proc. *IEEE Int. Conf. on Robotics and Automation, Taipei, Taiwan, pp.1191-1196.*

R. Storn and K. Price, 2005. *Differentila Evolution – a Practical Approach to Global Optimization,* Springer, Germany.

S.Besnard,W.Khalil, 1999. Calibration of parallel robots using two inclinometers, *Proceedings of the 1999 IEEE int. Conference on Robotics & Automation, Detroit, Michigan, pp.1758-1763.*

Shamma J.S. and Whitney D.E., 1987. A mehod for inverse robot calibration, *J. of Dynamic Systems, Measurement and Control. Vol, 109,pp.36-43.*

W. Khalil and S. Besnard, 1999. Self calibration of Stewart-Gough parallel robots without extra sensors, *IEEE transactions on robotics and automation, Vol.15, No.6, December 1999. pp.1116-1121.*

Xiaolin Zhong, John Lewis etc., 1996. Inverse Robot Calibration Using Artificial neural Networks, *Engng Applic. Artif. Intell. Vol. 9, No. 1, pp. 83-93.*

Yu Liu, Bin Liang et al., 2007. Calibration of a Steward Parallel Robot Using Genetic Algorithm, *Proceedings of the 2007 IEEE International Conference on Mechatronics and Automation, August 5-8, 2007, Harbin,China. pp. 2495-2500.*

SURVEILLANCE SYSTEM USING A MOBILE ROBOT EMBEDDED IN A WIRELESS SENSOR NETWORK

Syed Irtiza Ali and Baerbel Mertsching
GET Lab, Institute of Computer Science, Electrical Engineering and Mathematics, University of Paderborn
Pohlweg 47-49, 33098, Paderborn, Germany
{*irtiza, mertsching@get.upb.de*}

Keywords: Mobile robot, Localization, Robot mapping, Path planning, Wireless sensor network.

Abstract: In this paper, we proposed a surveillance system for guard robots to perform indoor navigation using a wireless sensor network. The aim is to provide a generic surveillance solution for multiple indoor scenarios. The multi-sensor based localization method for the robot has been employed to overcome the shortcomings of the standard AMCL based localization technique. It is also helpful in dealing with sensor limitations. The proposed strategy has been implemented and tested within lab environments. The results show a fair reduction in processing time required by convergence of localization process.

1 INTRODUCTION

The motivation of this paper is to present a generic surveillance strategy for different indoor environments (offices or museums) by making use of a mobile robot and a static wireless sensor network (WSN). The proposed system consists of surveying an entire floor. The floor consists of different rooms and every room is equipped with at least one motion detector which is combined with a radio transmitter. A mobile robot carries a receiver node and a map of the environment and it resides in one of the rooms. In case of an intrusion the activated motion detector generates an alarm via the network and brings the robot into action. The robot being in an autonomous mode, navigates to the location of the detected intruder where it is switched to the teleoperated mode. A human safeguard (e. g. in another building) can make use of the robot's camera to identify the intruder who may be an employee or a burglar. In future such robots may be equipped to tackle the intrusion as well.

In the scenario the robot utilizes the principle of simultaneous localization and mapping (SLAM). The localization process is a prime issue in the field of autonomous mobile robots. Its accuracy depends on the preciseness of sensors and the exactness of the map. There exist different methods to estimate the position of a mobile robot for indoor and outdoor scenes. Global positioning system (GPS) is a standard method to localize a mobile robot in outdoor scenes but this turns out to be inefficient for indoor scenarios due to a poor reception of GPS signals. Therefore we have to look for an alternate technique for localization in our problem domain. In this regard, the term SLAM addresses a dependency of the mapping on the localization and vice versa. It has been comprehended in sufficient detail in various robotics literature (see (Durrant-Whyte and Bailey, 2004) and (Fresse, 2006)). The SLAM solutions based on particle filter effectively deal with non-linearities existing in environments (see (Arulampalam et al., 2002) and (Montemerlo et al., 2003)). Therefore, we have selected a particle filter based SLAM solution to implement our strategy. Normally, most of these techniques make use of sensor readings and then apply Bayesian classifier based calculations to perform localization. It is safe to presume here that SLAM methods based on particle filter yield the acceptable results in specific scenarios provided a precise *a priori* map of an environment is available. However, sensors are subjected to noise in the environment resulting in an imprecise perception. Therefore it is appropriate here to use a multi-sensor based localization solution as presented in (Castellanos and Tardos, 1999), (Wu and Johnson, 2008) and (Wold et al., 2002) instead of relying on a single sensor. This approach performs localization by fusing data received from different heterogeneous sensors. Navigation is another important aspect in the field of an autonomous mobile robots and it mainly aims at reaching a particular location while avoiding both dynamic and static obstacles.

The collaborative use of a mobile robot and WSN had been previously presented with different aspects. (Batalin et al., 2004) have discussed a mobile robot navigation using a sensor network. A mobile robot receive signals from different sensor nodes and decides which sensor node is nearest. It then performs localization. This method clearly obviates the use of a known map but the accuracy of estimation is poor. An interesting application scenario to perform the flying robot navigation using sensor network has been discussed in (Corke et al., 2005). The next section discusses the surveillance strategy.

2 SURVEILLANCE STRATEGY

The generic surveillance strategy is shown in the figure 1 which illustrates a multiple room scenarios. This work focuses mainly on building a robust and the reliable surveillance system for an indoor environments. Therefore, we have utilized standard techniques for path planning and navigation (local and global) whereas the process of localization is improved using a multi-sensor localization process.

As shown in figure 1 every room is deployed with radio transmitters R_n. The radio transmitters are connected to motion detectors and generate an alarm via the network upon the detection of any physical intrusion. The alarm message contains the transmitter node identification number and its position. The robot receives a wake-up call upon detection of intrusion and start to estimate its location using a multi-sensor localization process. The shortest path from the robot's current position to the target position of sensor node is generated using a wavefront based path planning technique. The robot then navigates toward the goal position while avoiding obstacles and performs visual sensing upon reaching its goal. The next sections provide a detailed discussion on different modules.

2.1 Establishing a Wireless Sensor Network

The first step involves the establishment of wireless sensor network. The *j=n-2* of total *n* radio nodes R_n are deployed in an area which the robot monitors. They are static and connected to motion detectors. The rest of the two nodes are attached to a mobile robot and a console PC. A mobile robot receiver picks up status messages generated by static transmitter nodes. The network is established among radio transmitters using a table based routing scheme. Each transmitter sends a beacon message to its neighboring nodes which acknowledges the beacon message with *routing table* messages. This is necessary to provide information about the next possible hop. After an exchange of acknowledgement messages, each node then broadcasts *node detected* messages to update the routing table. Each motion detector has its defined vicinity in which it can detect the movement of different physical objects. Whenever a movement is sensed by a transmitter node, it generates an alarm which is received both by a mobile robot and the console PC. The alarm contains an identification number of the transmitter node. The position of the transmitter node on *a priori* map is identified by its number. Figure 2 displays a graphical user interface (GUI) running on the console PC. The placement and status of each radio transmitter node in an established sensor network can be monitored through this designed GUI. It is dynamic in a sense that it is capable to load a map of a new operational environment and can adjust the placement of the sensor nodes in the new environment.

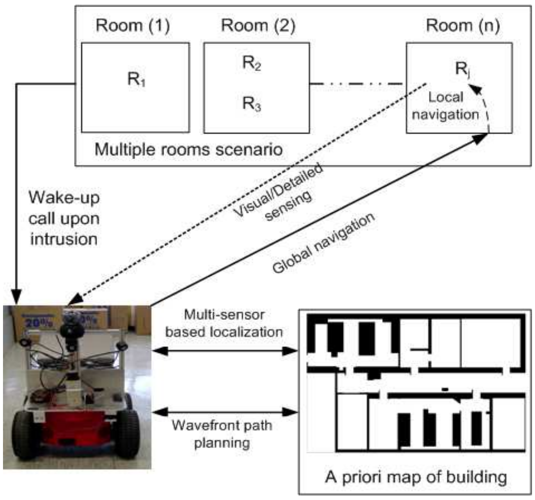
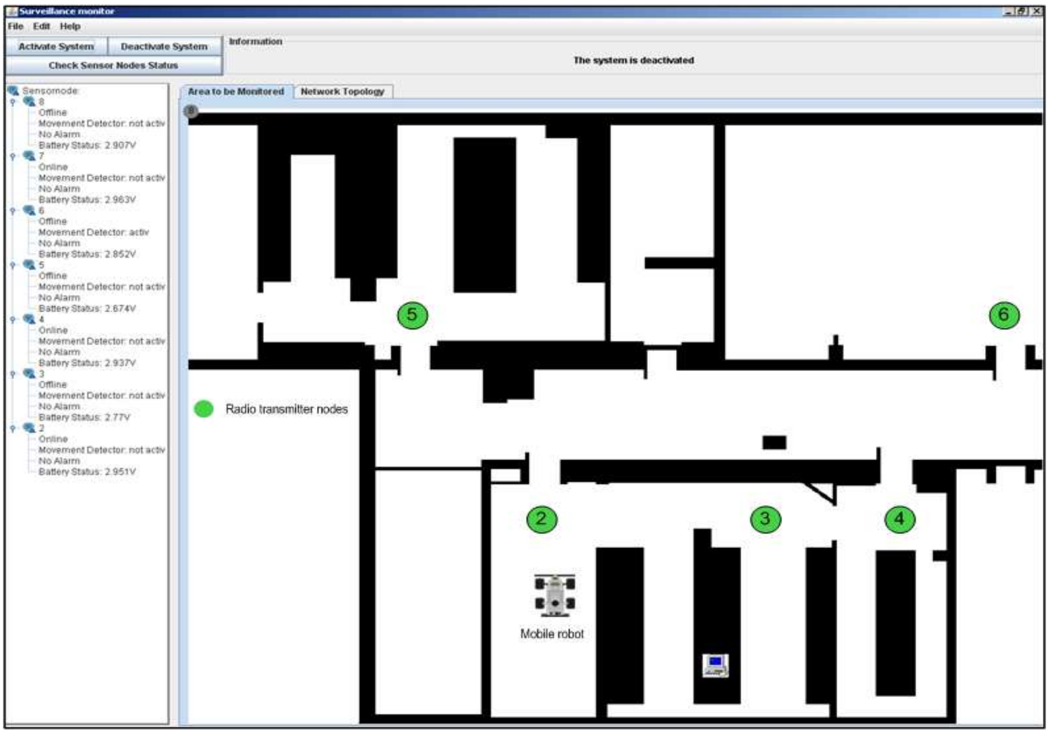


Figure 1: Generic surveillance strategy of a mobile robot.

2.2 Localization: Pose Estimation of a Mobile Robot

Once an alarm is generated, the next step is to estimate the location of a mobile robot which is assumed to be unknown in the begining. In the current application, the presence of a noise in the sensor models and the imprecise actuator control of mobile robots makes it difficult to accurately estimate the robot position even for the known environments. A fast convergence of the estimated position is another challenge in the field of localization. In this regard, we propose a multi-sensor based localization scheme to deal with different issues faced during the position estimation



The system is deactivated

Sensornode:
8
Offline
Movement Detector: not activ
No Alarm
Battery Status: 2.907V
7
Online
Movement Detector: not activ
No Alarm
Battery Status: 2.963V
6
Offline
Movement Detector: activ
No Alarm
Battery Status: 2.852V
5
Offline
Movement Detector: not activ
No Alarm
Battery Status: 2.674V
4
Online
Movement Detector: not activ
No Alarm
Battery Status: 2.937V
3
Offline
Movement Detector: not activ
No Alarm
Battery Status: 2.77V
2
Online
Movement Detector: not activ
No Alarm
Battery Status: 2.951V

Figure 2: Graphical user interface on the console PC.

of a mobile robot in an indoor environments. A mobile robot performs the localization using an adaptive Monte Carlo localization (AMCL) technique (Pfaff et al., 2006). It is applicable to both local and global localization problems. It is simply a variant of particle filtering (Arulampalam et al., 2002). It makes use of the recursive Bayesian filtering scheme in order to estimate a mobile robot location in an environment. It also requires a decent sensor model and the motion model of a mobile robot. It is worth mentioning here that AMCL is very much capable to handle complex, multimodal (non-Gaussian) posterior distributions of a mobile robot locations. However, it has difficulties when the pose of a mobile robot is high dimensional because the number of particles increases exponentially with the dimensionality of state space and hence increases the computational complexity of the overall process. This is yet an open research issue in the field of SLAM and the human body tracking applications.

AMCL estimates the pose from an input data of an odometry sensor and a laser range finder. AMCL requires an accurate sensor model but it is difficult to design a perfect sensor for the varied environments. For example there are cases where a laser range finder fails to provide the reliable range data, especially in the presence of glass windows or doors. The presence of the bright light and the vibrations produced by a mobile robot also affect the performance of a laser range finder. In order to deal with these problems, we have implemented a multi-sensor based localization strategy as shown in figure 3. The scheme is inspired by the multi-sensor data fusion techniques presented in (Castellanos and Tardos, 1999) and (Wu and Johnson, 2008) with an aim to deal with drawbacks of a laser range finder with an added advantage of lowering the convergence time during pose estimation process. The main steps involve are:

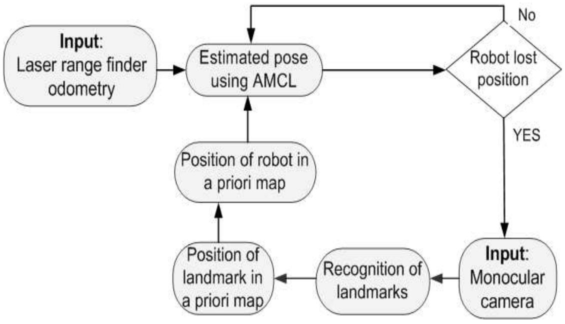


Figure 3: A Multi-sensor localization process.

1. **Exploration of the Environment.** A mobile robot explores an area using a monocular web camera mounted on a mobile robot.
2. **Recognition of Landmarks.** We have selected a segmentation method provided by (Aziz and Mertsching, 2006) in order to recognize the different landmarks. The segmentation is reliable for

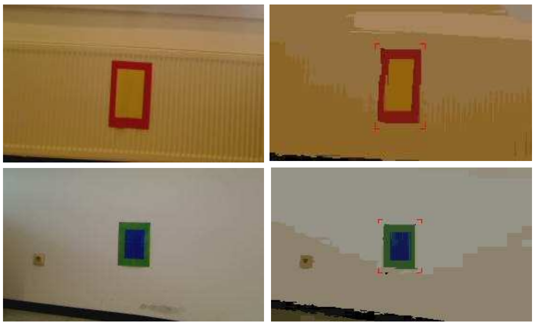


Figure 4: Left: original images and right: segmented output.

indoor scenarios. Figure 4 displays different landmarks and the segmented output. The landmarks are placed at known positions.

3. **Position of the Landmark in a Map.** The detected landmark is matched with known landmarks to determine its global position coordinates on the map.
4. **Position of a Mobile Robot in a Map.** The pose of the robot is calculated using the landmark's position and distance information obtained through a laser range finder. The procedure is explained in the subsequent section.
5. **AMCL.** The robot pose is provided to the AMCL localization module as an initial position of a mobile robot to estimate its actual pose in the map.
6. **Relocalization.** Steps 1-5 are repeated whenever the robot position is lost during navigation applications.

2.2.1 Orientation of Robot

Figure 5 illustrates the method to find out the orientation of the robot θ_r relative to a priori map in the world coordinates with following details:

1. The pose information of the wall (x_{wall}, y_{wall}, θ_{wall}) relative to the world coordinates is known.
2. The orientation of the camera θ_{cam} relative to the robot coordinates is also known.
3. The landmark is detected such that it should be in the middle of an acquired visual input. A laser range finder and a camera both are identically oriented with respect to robot coordinates.
4. The angles θ_{s1} and θ_{s1} are computed by adding and subtracting 10 degrees to θ_{cam}.
5. The distances d_1 and d_2 are then calculated using a laser range finder for θ_{s1} and θ_{s2} respectively.
6. The angle θ_{w-r} is calculated to find out the wall to a mobile robot orientation.

$$\theta_{w-r} = \arctan \frac{d_2 \cos\theta_{s2} - d_1 \cos\theta_{s1}}{d_2 \sin\theta_{s2} - d_1 \sin\theta_{s1}} \qquad (1)$$

7. Finally, the orientation of a mobile robot relative to the wall θ_r is computed using:

$$\theta_r = \theta_{wall} - \theta_{w-r} \qquad (2)$$

Finally a mobile robot location relative to a priori map is found out using θ_r and range information of landmark.

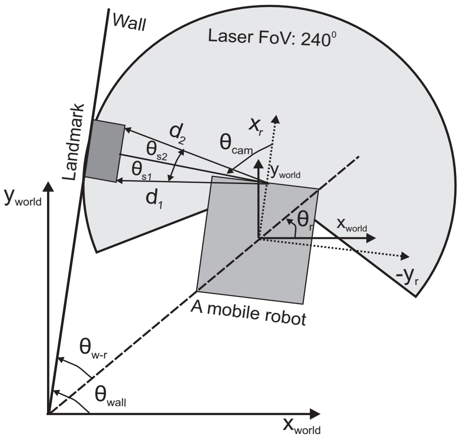


Figure 5: Pose estimation of a mobile robot in *a priori* map using multi-sensor localization.

2.3 Navigation

After estimating the global position of a mobile robot in a known map, the next step is to navigate a mobile robot to a goal location. The whole procedure of navigation is shown in figure 6. When any of the radio transmitters detects a motion in its vicinity, it generates an intruder alarm signal. It is then broadcasted to other wireless sensor nodes. This alarm signal provides the identification number (IN) of a particular transmitter node which generated an alarm. The *IN* is then found out using a table which contains a list of coordinate position of different transmitter nodes and this position is considered as a goal location for the navigator module. The goal location is needed to find a desired path from the actual position of the robot. The desired path is calculated using a path planning algorithm. Path planning is a well discussed topic in the field of a mobile robot navigation. We have selected the *wavefront* path planning algorithm (Behring et al., 2000) due to its suitability with grid based maps. The wavefront planning algorithm calculates a list of waypoints between a mobile robot position and a goal location. This list is necessary to generate the shortest possible path among them after taking into account the size of the robot and a safe distance from different obstacles. Once the successful path is discovered, the next step is to navigate a mobile robot between its present location and a goal

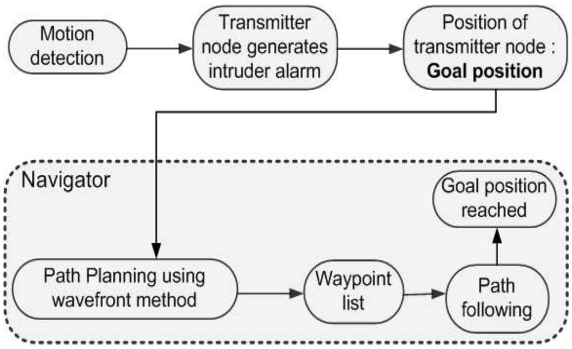

Motion detection
Transmitter node generates intruder alarm
Position of transmitter node : Goal position
Navigator
Goal position reached
Path Planning using wavefront method
Waypoint list
Path following

Figure 6: Navigation process.

by making use of the robust path following method. This also requires an integrated obstacle avoidance behavior for both the static and the dynamic obstacles. Normally, the list of static obstacles is known through an *a priori map* while a well defined local navigation strategy is required to deal with dynamic obstacles. There are different methods to perform this kind of navigation. We have utilized the *nearness diagram* (Minguez and Montano, 2004) approach. This approach provides a good methodology to avoid static as well as dynamic obstacles for indoor environments. The robot then follows a generated path and navigates safely toward its goal. Once it reaches the goal location, it generates a *goal reached* message. There is a provision to perform an exploration of the goal location surroundings with the help of an available visual sensor. This also enables a console PC to monitor a suspected area. The experiments and their outcomes are discussed in the next section.

3 EXPERIMENTS AND DISCUSSION

The experiments are conducted at GET lab, University of Paderborn. We used the customized Pioneer 3AT robot *GETbot* equipped with the two dimensional and a 240 degree field of view laser range finder (Hokuyo, 2009) and the pan-tilt based webcam for exploring an environment and avoiding the obstacles. Tmote sky sensor nodes (Tmote, 2009) are used as wireless nodes in a static WSN. They are attached to the motion detector. The map of the experimental setup and a pre-deployed static WSN of 6 nodes are shown in figure 2. It is a regular office environment with narrow door openings of about 90 cm. The nodes are deployed on the ceiling. Figure 1 shows the experimental mobile robot being employed. The surveillance strategy is implemented using a player-stage (Collet et al., 2005) robot control toolkit.

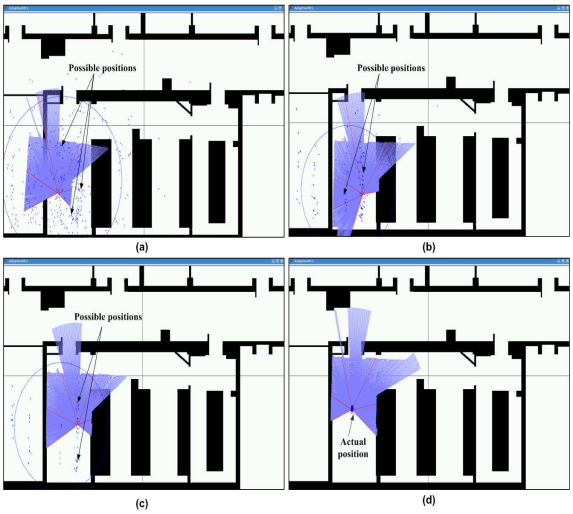


Figure 7: Robot localization using standard AMCL (a) T=0 s and particles=10,000. (b) T=10 s and particles=5000. (c) T=15 s and Particles=1000. (d) T=20 s and particles=100.

The experiments are performed on the basis of *event handling* and the generation of an alarm is considered as an event. The two different navigation behaviors are generated according to following conditions:

1. The first alarm received by a mobile robot is considered as the priority alarm which is then locked to reach the goal location. The alarms received afterwards are considered as the false alarms.
2. The last alarm received by a mobile robot is considered as a priority alarm.

A mobile robot reaches its goal location area with an accuracy of 0.25 meters. It is set as a parameter during the path computation step. The availability of an exact map of an environment increases the robust behavior. The results of localization performed using standard AMCL is shown in figure 7. The initial position of a mobile robot is unknown. Once a mobile robot starts its localization process, it begins matching the scans obtained through a laser range finder with a known map of an environment and awards high weightage to most probable matched places. In this way, a mobile robot tends to localize itself to most probable position over the time. The exact match sometimes takes up to several seconds to find an accurate estimation of a mobile robot pose as shown in figure 7. Symmetric environments are prone to false position estimations. In order to avoid these problems we have placed landmarks as shown in figure 4 at different places in the experimental scenario. It is important to highlight that the size and placement of landmarks is an important factor. The landmarks must be placed at the height of a laser range finder. This is advantageous in avoiding collisions with glass doors which are otherwise not detected by the employed laser range finder. The positions of landmarks on the map are known. The visual sensor is then utilized to perform a multi-sensor based localization process. This is quite useful in estimating a mobile robot pose on the map. It also speeds up the convergence process of the localization from around 20 seconds to 10 seconds. However, it depends on a good landmark detection scheme. It has been tested that this methodology works quite efficiently whenever the robot needs to perform self re-localization in wake of position loss. Overall this results into a more reliable and the efficient navigation behavior.

4 CONCLUSIONS AND FUTURE DIRECTIONS

We have presented a generic surveillance strategy for a guard robot using a wireless sensor network. The scheme has been implemented and worked out for different indoor scenarios. Our approach presents an improved localization process by employing a multi-sensor localization technique. It also allows the integration of different sensors to deal with different kinds of environment. The results show that a fast convergence of the localization process is achieved while effectively reducing the effects of a sensor noise. In the future work, we intend to see how the system performs reliably providing a relaxation in assumed conditions and parameters. The detailed comparison of a proposed localization strategy with other standard techniques is also an immediate step.

ACKNOWLEDGEMENTS

The authors would like to thank the higher education commission of Pakistan and the DAAD of Germany for supporting the research studies at University of Paderborn, Germany. The authors would also like to thank Mr. D. Fischer, Mr. M. Z. Aziz and Mr. S. Shafik for providing useful comments.

REFERENCES

Arulampalam, S., Maskell, S., Gordon, N., and Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Trans. on Signal Processing*, 50(2):174–188.

Aziz, M. Z. and Mertsching, B. (2006). Color segmentation for a region-based attention model. In *Proceedings of 12th Workshop Farbbildverarbeitung*, pages 74–83.

Batalin, M. A., Sukhatme, G. S., and Hatting, M. (2004). Mobile robot navigation using a sensor network. In *Proceedings of IEEE Intl. Conf. on Robotics and Automation*, pages 636–641.

Behring, C., Brancho, M., Castro, M., and Moreno, J. (2000). An algorithm for robot path planning with cellular automata. In *Proceedings of Intl. Conf. on Cellular Automata for Research and Industry*, pages 11–19.

Castellanos, J. A. and Tardos, J. D. (1999). *Mobile robot localization and map building - A multi-sensor fusion approach*. Kluwer Academic Publishers.

Collet, T. H. J., MacDonald, B. A., and Gerkey, B. P. (2005). Player 2.0 : Toward a practical robot programming framework. In *Proceedings of Australian Conf. on Robotics and Automation*.

Corke, P., Peterson, R., and Rus, D. (2005). Networked robotics: Flying robot navigation with a sensor network. *Springer Tracts in Advanced Robotics*, 15:234–243.

Durrant-Whyte, H. and Bailey, T. (2004). Simultaneous localization and mapping (slam): Part i the essential algorithms. *Robotics and Automation Magazine*, 13:99–110.

Fresse, U. (2006). A discussion of simultaneous localization and mapping. *Autonomous Robots*, 20(1):25–42.

Hokuyo (2009). *Laser range finder URG-LX04*. www.hokuyo-aut.jp/02sensor/07scanner/urg.html, site accessed: 22.02.2009.

Minguez, J. and Montano, L. (2004). Nearness diagram navigation (nd) : Collision avoidance in troublesome scenarios. *IEEE Trans. on Robotics and Automations*, 20(1):45–59.

Montemerlo, M., Thrun, S., Koller, D., and Wegbreit, B. (2003). Faststlam 2.0 : An improved particle filtering algorithm for simultaneous localization and mapping that provable converges. In *Proceedings of Intl. Joint Conf. on Artificial Intelligence*, pages 1151–1156.

Pfaff, P., Burgard, W., and Fox, D. (2006). Robust monte-carlo localization using adaptive likelihood models. In *European Robotics Symposium*, pages 181–194. Springer-Verlag Germany.

Tmote (2009). *Tmote sky nodes IEEE 802.15.4*. www.sentilla.com/pdf/eol/tmote-sky-datasheet.pdf, site accessed: 22.02.2009.

Wold, J., Burgard, W., and Burkhardt, H. (2002). Robust vision-based localization for mobile robots using an image retrieval system based on invariant features. In *Proceedings of IEEE Intl. Conf. on Robotics and Automation*, pages 359–365.

Wu, A. D. and Johnson, E. N. (2008). Methods for localization and mapping using vision and inertial sensors. In *Proceedings of AIDA Guidance, Navigation and Control Conf. and Exhibit*. AIAA 2008-7441.

OBSTACLE AVOIDANCE FOR AUTONOMOUS MOBILE ROBOTS BASED ON POSITION PREDICTION USING FUZZY INFERENCE

Takafumi Suzuki
Graduate School of Science and Technology, Keio University, 3-14-1, Hiyoshi, Kohoku-ku, Yokohama 223-8522, Japan
suzuki@yoshida.sd.keio.ac.jp

Masaki Takahashi
Department of System Design Engineering, Faculuty of Science and Technolog
Keio University, 3-14-1, Hiyoshi, Kohoku-ku, Yokohama 223-8522, Japan
takahashi@sd.keio.ac.jp

Keywords: Autonomous Mobile Robot, Obstacle Avoidance, Fuzzy Potential Method, Omni-directional Mobile Robot.

Abstract: This study presents an obstacle avoidance method for Autonomous Mobile Robot by Fuzzy Potential Method (FPM) considering velocities of obstacles relative to the robot. The FPM, which is presented by Tsuzaki, is action control method for autonomous mobile robot. In the proposed method, to decide a velocity vector command of the robot to avoid moving obstacles safely, Potential Membership Function (PMF) considering time until colliding and relative velocity is designed. By means of considering predicted positions of the robot and the obstacle calculated from the time and the relative velocity, the robot can start avoiding behaviour at an appropriate time according to the velocity of the obstacle and the robot. To verify the effectiveness of the proposed method, numerical simulations and simplified experiment intended for an omni-directional autonomous mobile robot are carried out.

1 INTRODUCTION

In the future, it's not difficult to image that we will often come across many autonomous mobile robots traversing densely populated place we live in. In such situation, because the autonomous mobile robots need to carry out their tasks in a place with unknown obstacles, the obstacle avoidance is one of the important functions of the robots. With a view to implementation of autonomous mobile robot working in doors, we employ an omni-directional platform as shown in Figure 1(a). For experimental verification, an omni-directional mobile robot shown in Figure 1(b) is developed. The robot has an omni-directional camera for environmental recognition, and can move to all directions by four omni wheels. While there are many studies about obstacle avoidance method focusing attention on possibility of avoidance, this paper presents the method focusing on not only possibility but also safer trajectory of avoidance. Even if there are the same situations that the robot needs to avoid a static obstacle, timing of beginning avoidance behaviour should vary according to the robot speed. If the obstacles are moving also, the timing should vary according to the velocities of the obstacles. To cite a case, in a situation that a robot and an obstacle go by each other as shown in Figure 2, the robot should avoid along the curved line like (iii) according to the speeds of the obstacle and own speed. To get to the goal with efficient and safe avoidance behaviour in the unknown environment for the robots, predicting the future obstacles' position by their current

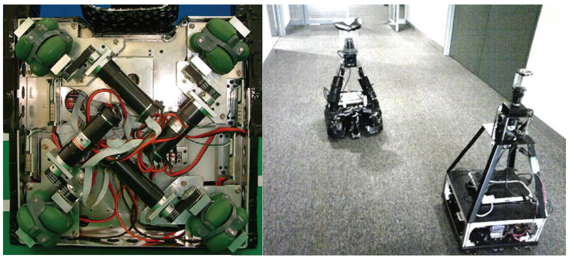

(a) (b)

Figure 1: An omni-directional platform of a prototype robot (a) and an example of a situation that the robot needs to avoid the other robot (b).

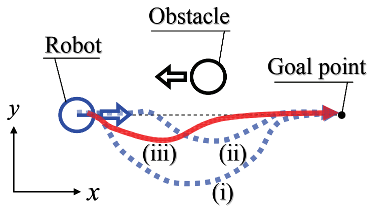


Figure 2: Example of a situation of obstacle avoidance.

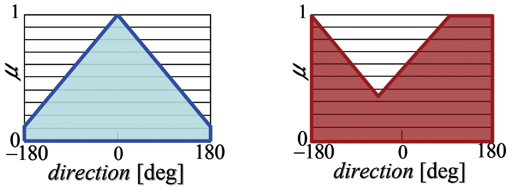


Figure 3: Example of PMF.

movements is needed. This paper introduces a real-time obstacle avoidance method introducing the velocity of obstacle relative to the robot. By means of considering predicted positions of the robot and the obstacle calculated from the time and the relative velocity, the robot can start avoiding behaviour at an appropriate time according to the velocity of the obstacle and the robot. Some researches focus attention on the velocity of obstacle (Ko et al., 1996) to avoid moving obstacles efficiently. In this research, virtual distance function is defined based on distance from the obstacle and speed of obstacle, however, only projection of the obstacle velocity on the unit vector from the obstacle to the robot is considered. In other words, the velocity of the robot is not considered. On the other hand, in (Ge et al., 2002), the velocity of the obstacle relative to the robot is considered. Our approach also employs the relative velocity. In addition to this approach, a position vector of the obstacle relative to the robot in the future is calculated by the relative position and the velocity. To solve the real-time motion planning problem, fuzzy potential method (FPM) is proposed by Tsuzaki (Tsuzaki et al., 2003). In this research, the method is applied to autonomous mobile robot which plays soccer. By adequate designing of potential membership function (PMF), it is realized that wheeled robots can get to the goal with conveying a soccer ball and avoiding obstacles. This method is easy to understand at a glance. However, in dynamic environment, to avoid moving obstacles efficiently, more specific guideline of designing is desired. In this paper, we introduce design method of PMF considering the predicted positions and discuss the availability by comparing the design of PMF considering the relative velocity and that not considering.

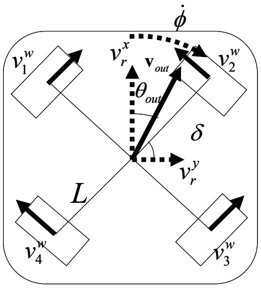


Figure 4: An omni-directional platform.

2 FUZZY POTENTIAL METHOD (FPM) FOR OMNI-DIRECTIONAL PLATFORM

In the Fuzzy Potential Method (FPM), a recent command velocity vector considering element actions is decided. Element actions are represented as Potential Membership Functions (PMFs), and then they are integrated by means of fuzzy inference. Furthermore, by using a state evaluator, the PMFs are modified adaptively according to the situation. The directions on the horizontal axis in Figure 3 correspond to the directions which are from -180 to 180 degrees and measured clockwise from the front direction of the robot. The priority for the direction is represented on the vertical axis. By use of the priority, direction and configured maximum and minimum speed, the current command velocity vector $\mathbf{v}_{out}$ is calculated. The command velocity vector is realized by four DC motors and omni wheels using following equations:

$$v_r^x = \|\mathbf{v}_{out}\| \cos\theta_{out} \tag{1}$$

$$v_r^y = \|\mathbf{v}_{out}\| \sin\theta_{out} \tag{2}$$

$$\begin{pmatrix} v_1^w \\ v_2^w \\ v_3^w \\ v_4^w \end{pmatrix} = \begin{pmatrix} \cos\delta & \sin\delta & L \\ \cos\delta & -\sin\delta & -L \\ -\cos\delta & -\sin\delta & L \\ -\cos\delta & \sin\delta & -L \end{pmatrix} \begin{pmatrix} v_r^x \\ v_r^y \\ \dot{\phi} \end{pmatrix} \tag{3}$$

where $\mathbf{v}_{out}$ and $\dot{\phi}$ are respectively current command velocity vector and rotational speed. δ is an angle of gradient for each wheel. L is a half of a distance between two catawampus wheels. v_i^w is a command movement speed of each i-th wheel.

PMF idea allows us to represent our knowledge and experiences easily, and furthermore it gives us easy understanding. The priority can be seen as a desire for each direction of the robot. In this paper, to discuss an obstacle avoidance problem, methods for generating of PMF to head to the goal and to avoid moving obstacles are introduced. This method has two steps. First step is generating PMFs. Second step is deciding the command velocity vector by use of fuzzy inference to integrate the PMFs. Hereinafter, design method of PMF considering the obstacle velocity relative to the robot and way to decide the command velocity vector by fuzzy inference are described.

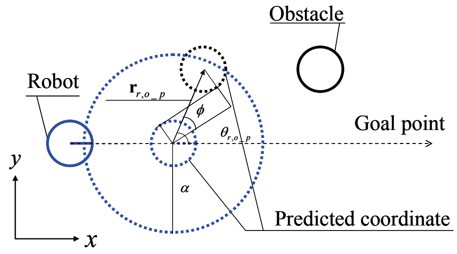


Figure 5: Predicted coordinate.

3 FPM CONSIDERING THE RELATIVE VELOCITY

To realize the obstacle avoidance in dynamic environment, the proposed method employs two different PMFs, one is considering the velocity of obstacle relative to the robot, the other is to head to the goal. PMF is denoted by μ which is function of θ. Note θ is the direction from -180 to 180 degrees measured clockwise from front direction of the robot. To simplify the analysis, it is assumed that the autonomous mobile robots detect obstacles by equipped external sensors and are capable of calculating the positions and velocities of obstacles relative to the robot. The shapes of the robot and the obstacles are treated as circles on 2D surface.

3.1 Design of PMFs

3.1.1 PMF for an Obstacle

To avoid moving obstacles safely and efficiently, an inverted triangular PMF by specifying a vertex, height and base width is generated. Because this PMF considers future positions of the robot and the obstacle, the robot can start avoiding the obstacle early and be prompted not to go on to the future collision position. For the purpose of safe avoidance, the PMF μ_o is generated.

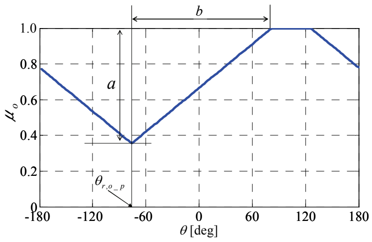


Figure 6: PMF for obstacle considering relative velocity.

First, to predict the future state of both the obstacle and the robot with the aim of efficient avoidance, a predicted relative position vector, in γT seconds, $\mathbf{r}_{r,o\_p} = (\mathrm{r}_{x\_p}, \mathrm{r}_{y\_p})$ is calculated as following equation:

$$\mathbf{r}_{r,o\_p} = \mathbf{r}_{r,o} + \gamma T \mathbf{v}_{r,o} \qquad (4)$$

where $\mathbf{r}_{r,o} = (\mathrm{r}_x, \mathrm{r}_y)$ is current position vector of the obstacle relative to the robot, and $\mathbf{v}_{r,o} = (\mathrm{v}_x, \mathrm{v}_y)$ is the current velocity vector of obstacle relative to the robot. γ is an arbitrary parameter from 0 to 1. T ,which is the time until the distance between the obstacle and the robot is minimum, is defined as following equation:

$$T = \frac{\left\| \mathbf{r}_{r,o} - \mathbf{p} \right\|}{\left\| \mathbf{v}_{r,o} \right\|} \qquad (5)$$

where $\mathbf{p} = (\mathrm{p}_x, \mathrm{p}_y)$ is a position vector of the obstacle relative to the robot when a distance in the future between the obstacles and the robot is minimum. $\mathbf{p}$ is calculated by means of relative position and velocity vector as following equation:

$$\begin{pmatrix} \mathrm{p}_x \\ \mathrm{p}_y \end{pmatrix} = \begin{pmatrix} \left\{ (\mathrm{v}_y / \mathrm{v}_x) \mathrm{r}_y - \mathrm{r}_x \right\} \Big/ (\mathrm{v}_y / \mathrm{v}_x + \mathrm{v}_x / \mathrm{v}_y) \\ -(\mathrm{v}_y / \mathrm{v}_x) \mathrm{p}_x \end{pmatrix} \qquad (6)$$

As described above, the predicted relative position vector, at the time γT seconds from now, $\mathbf{r}_{r,o\_p}$ is calculated as Figure 3 shows. By use of this position vector, a predicted obstacle direction relative to the robot θ_{r,o_p} is calculated as following:

$$\theta_{r,o\_p} = \arctan\left(\frac{\mathrm{r}_{y\_p}}{\mathrm{r}_{x\_p}} \right) \qquad (7)$$

where, θ_{r,o_p} is decided to be the vertex of the inverted triangle.

Next, as a measure to decide how far the robot should depart from the obstacle, a is defined as the height of the inverted triangular PMF. a is described as following equation:

$$a = \frac{\alpha - \left\| \mathbf{r}_{r,o\_p} \right\|}{\alpha - R_{r,o}} \qquad if \ \left\| \mathbf{r}_{r,o\_p} \right\| < \alpha \qquad (8)$$

$$R_{r,o} = R_r + R_o \qquad (9)$$

where R_r and R_o denote respectively the radius of the robot and that of the obstacle treated as circles. If the calculated obstacle position at γT seconds later is inside of a circle with radius α from the robot position at γT seconds later, the PMF for obstacle avoidance considering the relative velocity is generated. In other words, if a predicted relative distance $\left\|\mathbf{r}_{r,o\_p}\right\|$ is below α, a is defined and the inverted triangular PMF corresponding to the obstacle is generated. Smaller the predicted relative distance is, larger the value of a is.

In addition, a base width of inverted triangular PMF is decided by following equation:

$$b = \eta\left\|\mathbf{v}_{r,o}\right\| + \phi \tag{10}$$

where ϕ is decided based on the sum of radiuses of the robot and the obstacle, and predicted relative position vector as Figure 3 shows. ϕ is calculated by following equation:

$$\phi = \arcsin\left(\frac{R_{r,o}}{\left\|\mathbf{r}_{r,o\_p}\right\|}\right) \tag{11}$$

b increases up to π [rad] in proportion to an absolute value of the relative velocity and predicted relative distance. If the obstacle comes at rapidly, for instance, the value of b increases. Hence, the base width grows shown in Figure 4, and the value of priority for the direction of the obstacle relative to the robot comes about to be reduced. η is a gain.

As mentioned above, by deciding the vertex, the height and the base width of inverted triangle considering the predicted relative position, PMF μ_o, which aims to early starting of avoidance behavior and prompt the direction of the velocity vector to be far from obstacle direction in response to the fast-moving obstacle, is generated.

3.1.2 PMF for a Goal

To head to the goal, a PMF μ_d shaped like triangle as shown in Figure 5. As a measure to decide how much the robot want to head to the goal, c is defined as the height of the triangular PMF. c gets the maximum value at an angle of the goal direction relative to the front direction of the robot, θ_d, and is described as following equation:

$$c = \begin{cases} \dfrac{\left\|\mathbf{r}_{r,d}\right\|}{\varepsilon} & if \quad \left\|\mathbf{r}_{r,d}\right\| \le \varepsilon \\ 1.0 & if \quad \left\|\mathbf{r}_{r,d}\right\| > \varepsilon \end{cases} \tag{12}$$

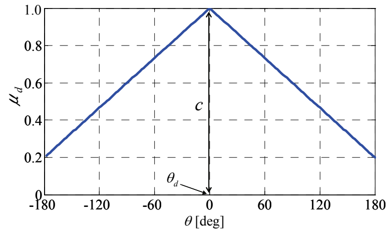


Figure 7: PMF for a goal point.

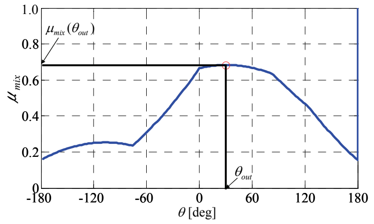


Figure 8: Mixed PMF.

where $\left\|\mathbf{r}_{r,d}\right\|$ is an absolute value of the position vector of the goal relative to the robot. ε is constant. If $\left\|\mathbf{r}_{r,d}\right\|$ is below ε, c is defined. The shorter the distance between the obstacle and the robot is, the smaller c becomes. Therefore the robot can decelerate and stop stably.

3.2 Calculation of Command Velocity Vector by Fuzzy Inference

The proposed method employs fuzzy inference to calculate the current command velocity vector. Specifically, The PMF μ_o, which considers the velocity of obstacle relative to the robot, and the PMF μ_d, which is to head to the goal, are integrated by fuzzy operation into a mixed PMF μ_{mix} as shown in Figure 6. μ_{mix} is an algebraic product of μ_o and μ_d as following equation:

$$\mu_{mix} = \mu_d \cdot \mu_o \tag{13}$$

Finally, by defuzzifier, the command velocity vector is calculated as a traveling direction θ_{out} and an absolute value of the reference speed of the robot base on the mixed PMF μ_{mix}. θ_{out} is decided as the direction θ_i which makes a following function $f(\theta)$ maximum.

$$f(\theta) = \sum_{i=j-n}^{j+n} \mu_{mix}(\theta_i) \tag{14}$$

where n is the parameter to avoid choosing undesirable θ_i caused by such as noises on the

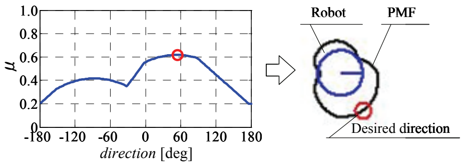


Figure 9: Visualization of PMF.

sensor data. Based on θ_{out} , v_{out} is calculated as following equation:

$$v_{out} = \mu_{mix}(\theta_{out})(v_{max} - v_{min}) + v_{min} \qquad (15)$$

where $\mu_{mix}(\theta_{out})$ is the mixed PMF μ_{mix} corresponding to the θ_{out} , v_{max} and v_{min} are configured in advance respectively as higher and lower limit of the robot speed.

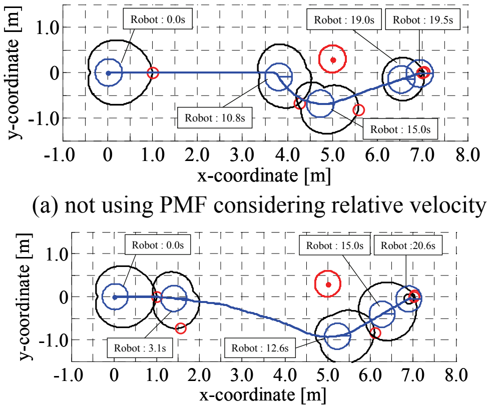
3.3 Visualization for PMF on Two-dimension Surface

It would be convenient to have a visualizer that show us why the robot will go on to the direction. In the proposed method, we can see aspects of the PMF on two dimension surface and understand easily the reason for choice of the direction. For example, a PMF described on polar coordinate shown in Figure 9(a) is comparable to the PMF described on x-y coordinate shown in Figure 9(b).

4 SIMULATION RESULTS

The radius of robot and obstacle are supposed to be both 0.3m, therefore, $R_{r,o} = 0.6\text{m}$. α in equation (8) is 1.6m. γ in equation (4) is 0.7. ε in equation (12) is 1.0m.

Figure 10, 11 and 12 show the simulation results when the robot passes the obstacle. Initial positions of the robot and the obstacle are respectively $(0\text{m}, 0\text{m})$ and $(5.0\text{m}, 0.3\text{m})$. The goal position of the robot is $(7.0\text{m}, 0\text{m})$. In the situation in Figure 10, the higher limit of robot speed is $v_{\max} = 0.5\text{m/s}$, the lower one is $v_{\min} = 0.0\text{m/s}$. The higher limit of acceleration of the robot is $a_r = 1.0\text{m/s}^2$. The simulations have done with three different obstacle speed $v_o = 0.0, 0.5\text{m/s}$, that the direction is negative on x -axis. Figure 10(a) and (b) show respectively the trajectory of the robot that the PMF for obstacle avoidance is generated without considering the relative velocity and that with considering the relative velocity, when $v_o = 0.0\text{m/s}$. In Figure 10(a), the robot gets close to the obstacle because the relative velocity is not considered. On the other hand, in Figure 10(b), the early starting of avoidance behaviour due to generating PMF by use of predicted information based on the relative velocity. In addition to the situation as in Figure 10(b), in Figure 11, the higher limit of the robot speed has been changed: $v_{\max} = 0.8\text{m/s}$. Even if the robot speed becomes more rapid, the robot succeed in efficient avoidance. In Figure 12(a) and (b), the trajectories of the robot, with PMF considering the relative velocity and not considering that, when the obstacle speed $v_o = 0.5\text{m/s}$. In (a), due to delay of starting avoidance behaviour, the robot collided with the obstacle. On the other hand, in (b), due to the early starting of the avoidance behaviour, the robot succeeded at the obstacle avoidance.



(a) not using PMF considering relative velocity

(b) using PMF considering relative velocity

Figure 10: Simulation results of an obstacle avoidance going by each other when speed of obstacle (v_o) is 0.0m/s and of a robot (v_r) is 0.5m/s .

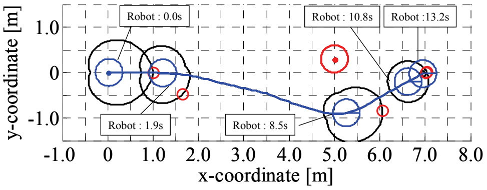
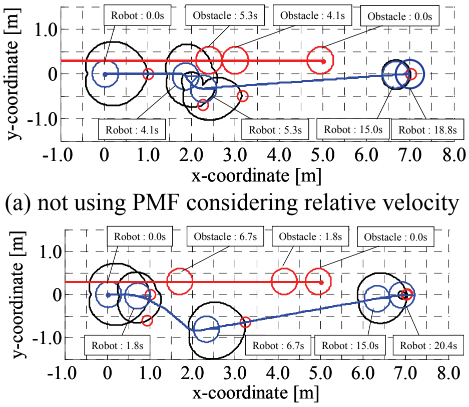


Figure 11: Simulation results of an obstacle avoidance going by each other when speed of obstacle (v_o) is 0.0m/s and of a robot (v_r) is 0.8m/s.

From these simulation results, it is confirmed that by an associating the PMF for avoidance with the relative velocity, faster the obstacle speed is, earlier the timing of the avoidance behaviour of the robot is, therefore the ability of avoiding obstacle can be enhanced.



(a) not using PMF considering relative velocity

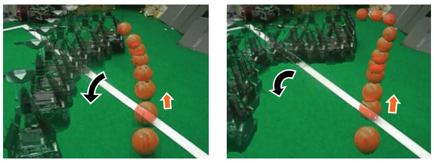
(b) using PMF considering relative velocity

Figure 12: Simulation results of obstacle avoidance going by each other when speed of an obstacle (v_o) is 0.5m/s and of a robot (v_r) is 0.5m/s.

5 EXPERIMENTAL RESULTS

To verify the effectiveness of the proposed method that employs PMF considering the velocity of the obstacle of the robot, a ball is supposed to be a moving obstacle and is rolled toward the robot. The robot recognizes the environment by the omni-directional camera. A position of a goal and that of an obstacle relative to the robot are calculated by extracting features based on objects' colours. The robot size is L $0.4\times$ W $0.4\times$ H 0.8m and the ball diameter is 0.2m. The radius of robot and obstacle are supposed to be 0.3m and 0.1m respectively, therefore, $R_{r,o}=0.4\text{m}$. α is set to 1.4m when the robot uses the proposed PMF which is considering relative velocity. When the robot doesn't use the proposed PMF, α is set to 2.4m. γ is 0.7. ε is 1.0m. $v_{\max}$ is 0.5m/s, $v_{\min}$ is 0.0m/s. a_r is 1.0m/s^2 .

When the robot used the proposed PMF, which was considering relative velocity, as shown in Figure 13 (a), it succeeded in avoiding the moving ball with smooth trajectory. On the other hand, in the situation Figure 13 (b), the robot with the PMF, which was not considering relative velocity, diverged once.



(a) (b)

Figure 13: Trajectories of the obstacle (ball) and the robot with the PMF considering relative velocity (a) and not considering relative velocity (b).

6 CONCLUSIONS

In this paper, design method of the potential membership function (PMF), which is considering the velocity of the obstacle relative to the robot for the purpose of avoiding the moving obstacle safely and smoothly, has been presented. In the proposed method, the proposed PMF for an obstacle and PMF for a goal are unified by fuzzy inference. By defuzzification, the command velocity vector of the robot is calcu lated and the obstacle avoidance has realized. A numerical simulation, which assumes an obstacle avoidance of autonomous omni-directional mobile robot, has done. As the result of the comparison between the design method of PMF using relative velocity and not using, it is confirmed that the ability of avoiding the moving obstacle can be enhanced. In addition, thorough simplified experiments, the real robot can avoid an obstacle using proposed method.

REFERENCES

Khatib, O., 1986. Real-time Obstacle Avoidance for Manipulators and Mobile Robots, In *Int. J. of Robotics Research*, vol.5, no.1, pp.90-98.

Ge, S., S., and Cui, J., Y., 2002. Dynamic Motion Planning for Mobile Robots Using Potential Field Method, In *Autonomous Robots,* vol.13, pp.207-222.

Arkin, R., C., 1989. Motor schema-based mobile robot navigation, In *Int. J. Robotics Research*, vol. 8, no. 4, pp. 92-96.

Ko, Y., N., and Lee, H., B., 1996. Avoidability Measure in Moving Obstacle Avoidance Problem and Its Use for Robot Motion Planning, In *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 96)*, pp.1296-1303.

Borenstein, J., and Koren, Y., 1991. The Vector Field Histogram Fast Obstacle Avoidance For Mobile Robots, In *IEEE Trans. on Robotics and Automation*, Vol.7, No.3, pp.278-288.

Fox, D., Burgard,W. and Thrun,S., 1997. The Dynamic Window Approach to Collision Avoidance, In *IEEE Robotics and Automation*, Vol. 4, No. 1, pp.1-23.

Tsuzaki, R., Yoshida , K., 2003. Motion Control Based on Fuzzy Potential Method for Autonomous Mobile Robot with Omnidirectional Vision. In *Journal of the Robotics Society of Japan* . vol.21, no.6, pp.656-662.

Otsuka, F., Fujii, H., Yoshida, K.,,2005. Action Control Based on Extended FPM for an Autonomous Mobile Robot. *The 26th Annual Conference of the Robotics Society of Japan*.

AUTONOMOUS CAMERA CONTROL BY NEURAL MODELS IN ROBOTIC VISION SYSTEMS

Tyler W. Garaas, Frank Marino and Marc Pomplun
Department of Computer Science, University of Massachusetts Boston
100 Morrissey Boulevard, Boston, MA 02125-3393, U.S.A.
tgaraas@gmail.com, frankmarinojr@gmail.com, marc@cs.umb.edu

Keywords: Robotic Vision, Neural Modeling, Camera Control, Auto White Balance, Auto Exposure.

Abstract: Recently there has been growing interest in creating large-scale simulations of certain areas in the brain. The areas that are receiving the overwhelming focus are visual in nature, which may provide a means to compute some of the complex visual functions that have plagued AI researchers for many decades; robust object recognition, for example. Additionally, with the recent introduction of cheap computational hardware capable of computing at several teraflops, real-time robotic vision systems will likely be implemented using simplified neural models based on their slower, more realistic counterparts. This paper presents a series of small neural networks that can be integrated into a neural model of the human retina to automatically control the white-balance and exposure parameters of a standard video camera to optimize the computational processing performed by the neural model. Results of a sample implementation including a comparison with proprietary methods are presented. One strong advantage that these integrated sub-networks possess over proprietary mechanisms is that 'attention' signals could be used to selectively optimize areas of the image that are most relevant to the task at hand.

1 INTRODUCTION

Recent advances in neuroscience have allowed us unprecedented insight into how assemblies of neurons can integrate together to establish complex functions such as the deployment of visual attention (Moore & Fallah, 2004), the conscious visual perception of objects (Pascual-Leone & Walsh, 2001), or the remapping of visual items between eye movements (Melcher, 2007). This accumulation of detailed knowledge regarding the structure and function of the individual neurons and neural areas that are responsible for such functions has led a number of neuroscientists to prepare large-scale neural models in order to simulate these areas. Some of the modeled areas include the primary visual cortex (McLaughlin, Shapley, & Shelly, 2003), the middle temporal area (Simoncelli & Heeger, 1998), or an amalgamation of areas (Walther & Koch, 2006). Although the usual motivation for creating such models is to ultimately make predictions about their possible mechanisms or functional roles in biological organisms, recent advances in parallel computing – in particular, the introduction of the Cell processor as well as the graphics processing unit (GPU) – will likely direct the attention of robotics researchers toward developing comprehensive neural models for use in robotic applications.

Robotic vision systems that are based on biologically inspired neural models represent an initially promising path to finally achieving intelligent vision systems that have the power to perform the complex visual tasks that we take for granted on a daily basis. A classic example is that of object recognition, at which computer vision systems are notoriously poor performers. Humans, on the other hand, can quickly – on the order of hundreds of milliseconds – and effortlessly recognize complex objects under a variety of situations – e.g., various lightning conditions, rotations, or levels of occlusion. Various models of how this processing may occur in humans have been proposed, which have resulted in increased object recognition abilities by artificial systems (e.g., Riesenhuber & Poggio, 1999; Walther & Koch, 2006). Consequently, it is likely that subsequent iterations of these models will make their way into future robotic vision systems.

Most neural models of visual areas operate in an idealized space (e.g., Lanyon & Denham, 2004); receiving pre-captured and manipulated images, whereas in robotic vision systems certain constraints are necessarily imposed. These constraints largely revolve around the need to process information in realistic time-frames – optimally real-time – as well as to interact directly with the physical world; likely through some form of video camera. Since these models will operate on input that cannot be known ahead of time, the system should be designed to handle a wide range of situations that may arise.

Most video cameras suitable for a robotic vision system include some ability to automatically monitor and adjust white-balance, exposure, and focus. However, in a robotic vision system that employs large-scale neural models, these automatic functions may lead to suboptimal processing conditions or even conflicts with the neural mechanisms. This paper presents a proof-of-concept method for manually controlling certain parameters in the camera to optimize the processing of a neural model of the retina, which will likely form the initial processing stage of future biologically inspired vision systems. In particular, the control of white-balance (WB) and exposure parameters are considered. Implementation details and a comparison to proprietary methods are given in the following sections.

2 SYSTEM OVERVIEW

The vision system presented here consists of a number of simple neural layers (2D layout of neurons that process image signals from nearby neruons) interconnected to form the basis for a robotic vision system; Figure 1 gives a simplified illustration of how the neurons within each layer are connected. The layers are modeled after a subset of the neurons present in the human retina. Figure 2 illustrates the individual neuron types and connections that constitute the artificial retina, which are briefly described below in order to establish the motivation for the WB and exposure sub-networks (subnets) presented hereafter. The neural model (excluding WB and exposure subnets) consists of 17 layers which total to approximately 225,000 individual neurons and 1.5 million connections. The network is designed to be executed on the GPU of a standard video card.

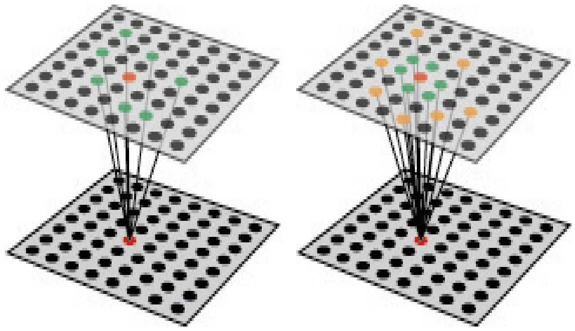


Figure 1: Simplified illustration of the connections used in the neural model and camera control subnets: (left) random connections to cells in previous layers and (right) random connections demonstrating a center-surround organization. Information in both cases flows from the upper layer to the bottom layer.

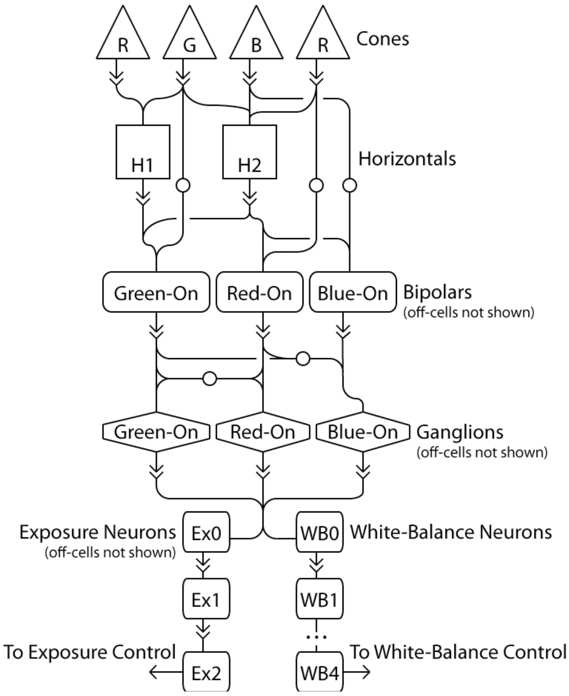


Figure 2: Connections between the various neuron types in the retina and camera control subnets. Solid arrows indicate excitatory connections while arrows with a white circle indicate inhibitory connections; ellipses between WB1 and WB4 indicate a continuation of the connection pattern directly above.

2.1 Apparatus

The network presented hereafter was simulated on two computer graphics cards (Nvidia 380 gtx) using an SLI setup. The OpenGL shading language (GLSL) was used to implement the computations for individual neurons. Video input was retrieved using

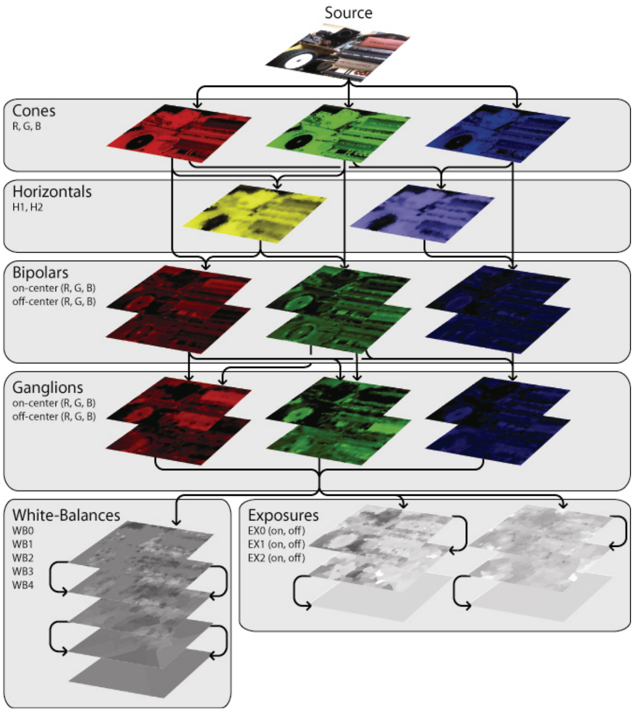


Figure 3: Activation maps and connection structure of the neural model and camera control subnets. Lighter areas represent higher activations while the colors indicate the spectral contributions to the activations.

a Cannon VCC4 video camera, and images were captured from the camera using the Belkin Hi-Speed USB 2.0 DVD Creator. Activations of the entire network can be computed very quickly: 100 iterations of computing activations for every single neuron take approximately 0.5 seconds.

2.2 Simulated Retina

The human retina is often considered a simple means for sensing light that enters the eye. On the contrary, the retina is actually a complex extension of the brain that is responsible for both reducing the amount of information transmitted to the various visual centers of the brain and converting the incoming signal into a form that is suited for higher-level processing by cortex. In the neural model presented here, we simulate the cones (R, G, B; referred to as long-, medium-, and short-wavelength cones in biological organisms, which are responsible for extracting the contributions of three primary color-components of the image), the horizontal cells (H1 & H2 cells, which essentially compute a 'blurred' version of the incoming image), on- and off-center bipolar cells (R, G, B cells, which compute an initial contrast-sensitive activation due to the antagonistic center-surround arrangement), and on- and off-center ganglion cells (R, G, B cells, which also compute a center-surround, contrast-sensitive signal that is also spectrally opposed, due to the inhibitory connections from bipolar cells). For the sake of brevity, we do not describe the specifics of individual neuron activations and connections. However, the essentials of the retinal neurons simulated here follow very closely those laid out by Dacey (2000) and Dowling (1987).

3 WHITE-BALANCE CONTROL

WB control in cameras was included so that changes in illumination could be countered to keep white areas within an image looking white. For instance, lighting that is stronger across the red spectrum of visible light will cause white areas to take on a reddish hue. Many different algorithms, such as white point estimation (Cardei, Funt & Barnard, 1999), chromaticity estimation using neural networks (Funt & Cardei, 1999), and gray world (Buchsbaum, 1980), have been proposed to control for changes in color due to the infinite spectrum of light sources. Although humans do not have the ability to directly control the color of objects as they are being received by the various early visual areas, neural mechanisms do exist to counter the effect of illuminants on the actual *perception* of color (Brainard, 2004). This ability is aptly referred to as color constancy.

The automatic white-balance mechanism described here is a subnet of the neural model portrayed above. The basic goal is largely the same as that of previously proposed mechanisms; that is, to make white objects project white color onto the incoming image regardless of the illumination color. As such, the proprietary automatic white-balance mechanism would provide an adequate means to achieve this; however, there are a few caveats that may make a specifically designed WB control mechanism desirable. First, ganglion cells, from which the WB function will be computed, do not directly encode the primary image colors (i.e., RGB); instead, they encode a spatially and spectrally opponent signal that encodes the differences between red/green and blue/yellow signals. This property may introduce differences between an optimal white-balance parameter set by proprietary mechanisms and the optimal white-balance parameter for network computation. Second, certain biologically inspired mechanisms may take advantage of having the computation of such things implemented directly inside the network. This will be discussed in detail later.

The WB subnet introduced here is conceptually very simple. It begins by including a layer into the network (WB0) that 'extracts' areas of the image that represent candidates for white or light gray regions (technically, B/Y – R/G neutral). The candidate areas are exactly those areas in which the on-center ganglion cells have nearly the same level of activation and where the sum of the activations is greater than some threshold. A small amount of programming code is given below which gives a basic idea of how neurons' activations in layer WB0 are computed; red, green, and blue variables store the average activations of incoming red, green, and blue on-center ganglion cells, respectively; on-center activations range from 0.0 (no activation) to 1.0 (full activation).

```
float intensity = red + green + blue;
float R = red / intensity;
float G = green / intensity;
float B = blue / intensity;

if(R > 0.25 && G > 0.25 && B > 0.25 &&
 total > 1.0)
   activation = (B – R)*4.0;
else
   activation = 0.0;
```

After layer WB0 has extracted the areas that are potentially white or light gray, neurons in WB1 then compute a local maximum of the WB0 neurons to which it is connected; Figure 1 (left) illustrates the basic connection structure. Finally, layers WB2 through WB4 perform a simple averaging of the neuron activations from incoming layers; however, only neurons with non-zero activation (i.e., those representing a candidate area) will contribute to the average. The end-product of the WB subnet is a value that can be used to step the white-balance parameter either towards a more blue hue or a more red hue depending on the situation. If, for instance, the activations of red and blue ganglion cells are close to equal across the image, the step functions will be zero and the white-balance parameter will not change. However, if red ganglion cells have larger activations, in general, then the stepping function will be negative, which will cause the camera to introduce a slightly bluer hue to the image.

The WB subnet was designed to balance the activations across red and blue on-center ganglion cells. Consequently, the subjective view of the image cannot be used to assess the performance of the subnet, which is contrary to proprietary mechanisms. With that said, the WB subnet adjusts

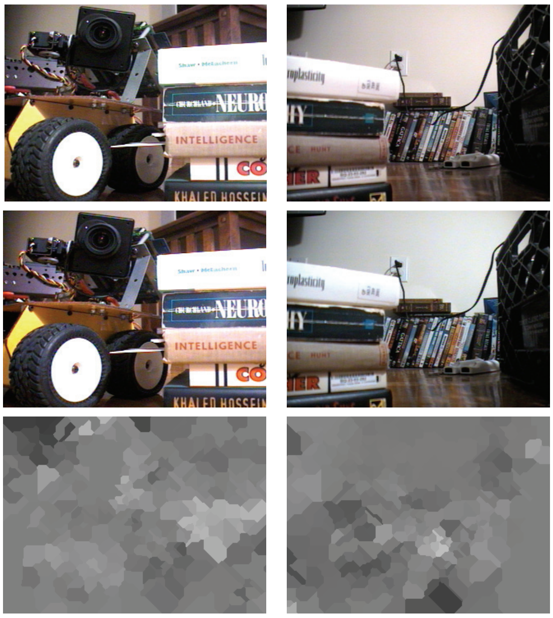


Figure 4: White-balance results: (top) adjusted image using proprietary auto-WB mechanism, (middle) adjusted image using the WB subnet, and (bottom) activation map of WB2. Lighter portions in WB2 represent candidate areas that contain greater activations of blue ganglion cells, while darker portions represent candidate areas greater activations of red ganglion cells.

the WB of the camera in much the same way as the proprietary mechanism in certain situations; see Figure 4 (left), for example. In contrast, other situations can produce deviations in WB settings between the subnet and proprietary mechanisms; see Figure 4 (right), for example. The size of WB steps should also be considered, as too large a step size will introduce over-correction and, ultimately, a ping-ponging of the WB parameter as the subnet slowly narrows in on the correct value; on the other hand, too small a step size will lead to a very slowly adjusting WB. Finally, in the current network, following a change in the white-balance parameter, it was necessary to insert a short delay before another step could be made; this was needed to allow the changes in image color due to the WB parameter change to spread through the various neural layers.

4 EXPOSURE CONTROL

One of the most remarkable properties of the human vision system is its ability to function over a strikingly large range of luminance conditions, a span of approximately 10 billion to 1 (Dowling, 1987). The human eye has essentially two ways of dealing with the variation it experiences in day-to-day luminance levels. (1) The pupil can reduce its area by a factor of approximately 16 due to changes in ambient illumination. (2) The circuitry in the retina is specially designed to handle two general lighting conditions: dim light, primarily handled by the rod-pathway in the retina; and bright light, handled by the cone-pathway in the retina.

Video cameras, on the other hand, do not have the luxury of such robust input mechanisms. Nevertheless, various methods have been developed to allow cameras to function under a rather impressive span of luminance levels – at least when all things are considered. The camera used for the present study employs two primary parameters that can be adjusted to compensate for luminance levels: iris size and gain control.

The network control of exposure is similar to that of WB in that a conceptually simple subnet progressively computes various properties of the incoming image which allows it to ‘step’ the relevant parameter towards optimizing some computation. The computation that is optimized in exposure control is contrast; too much light entering and the image gets ‘washed-out’; too little light creates an underexposed image. This mechanism in particular will likely be very important to robotic vision systems using biologically neural models, as contrast has shown to play a particularly critical role in the neural computations that take place in the primate visual cortex (Sceniak et al., 1999).

As with the WB subnet, the functioning of the exposure subnet is conceptually very simple. Essentially, the subnet attempts to maximize the contrast of two spatially adjacent areas using the on- and off-center ganglion cells. Recall that in the neural model of the retina (and the biological retina) contrast plays a specific role for two classes of neurons, bipolar cells and ganglion cells. That is, these cells compute an activation that highlights high contrast areas of the image. Consequently, much of the work required for computing our exposure control function is already implemented.

The remaining work is performed by two independent subnets, an off-subnet and an on-subnet. Each subnet first computes a local maximum of the incoming ganglion cell activations (on-center ganglion cells will have higher activations in bright areas, especially if it is adjacent to a dark area, and vice-versa for off-center ganglion cells). This maximum is then averaged across the image to produce an exposure step-value similar in nature to the WB step-value, with one difference, the step value for the exposure control must work to control both the iris and gain of the camera, which is handled by a simple scheme: changes to the iris take precedence over changes to gain, which instead serves to fine-tune the exposure using small step-values.

Sample results of the exposure subnet are shown in Figure 5 for both bright-light conditions (left) and dim-light conditions (right). Exposure values in dim-light conditions closely follow those computed by the proprietary control mechanisms. However, significant differences can be seen in bright-light conditions. This is likely due to the goal of the exposure subnet to extract a maximum contrast signal, which may only occur in a portion of the image, as opposed to enhancing a global contrast signal. This difference is most prominent around the wheel in Figure 5 (left) where the proprietary mechanism introduces spectral highlighting on the rubber (black) portion of the wheel; whereas when under the exposure subnet’s control, the gain is weaker to allow the natural blackness of the wheel to maximize the contrast between the plastic (white) and rubber (black) portions.

5 CONCLUSIONS

The current paper introduced small subnets that can be integrated into biologically inspired neural models of human visual areas to control the WB and exposure parameters on most standard video cameras. In contrast to their usual functions, WB and exposure parameters are used to optimize the actual processing that occurs in the neural model, as opposed to simply providing a clearer image. The subnets of this particular implementation are based on the activations of on-center and off-center ganglion cells from a neural model of an artificial retina.

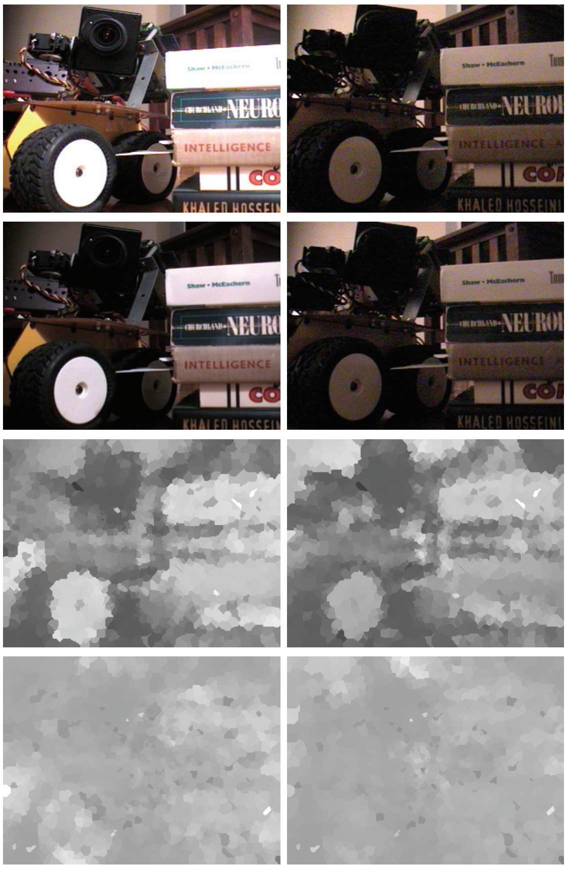


Figure 5: Exposure results: (top) adjusted image using the proprietary auto-exposure mechanism, (upper-middle) adjusted image using the exposure subnet, (bottom-middle) activation map of on-center EX0, and (bottom) activation map of off-center EX0.

Aside from customizing the parameters of the camera to optimize model computation, the subnets introduced here have other features that would make them a desirable replacement for proprietary mechanisms. One feature in particular could provide a substantial benefit, which is the ability to selectively optimize computation for areas in the image in which the neural model is ‘interested’. Indeed, one of the most studied neural signals in biological organisms is that of attention, which is often implemented in artificial neural models (Lanyon & Denham, 2004). Consequently, with very little modification, the subnets presented here could be modified to selectively provide emphasis to attended areas based on incoming attention signals. For instance, imagine a robotic vision system that is placed in a daylight setting receiving very bright light from the sun. If the robot wishes to examine a dark portion of the incoming image – say the lettering of a poster printed on a black background, proprietary mechanisms will be inadequate as they will selectively optimize the range of high pixel values – i.e., those representing bright areas. Instead, if the image is adjusted to optimize the range of low pixel values – i.e., those representing the poster, the robot may then successfully achieve its goal. Attention signals representing the robot’s desire to inspect the poster would provide a perfect indicator by which to optimize the correct portion of the incoming image. Future implementations will be directed toward realizing such models.

REFERENCES

Brainard, D. H. (2004). Color constancy. In L. Chalupa & J. Werner (Eds.), *The Visual Neurosciences* (pp. 948-961): MIT Press.

Buchsbaum, G. (1980). A spatial processor model for object colour perception. *Journal of Franklin Institute*, 310, 1-26.

Cardei, V., Funt, B. & Barnard, K. (1999). White point estimation for uncalibrated images. *Proceedings of the IS&T/SID seventh color imaging conference*. (pp. 97-100). Scottsdale, AZ, USA.

Dacey, M. (2000). Parallel pathways for spectral coding in primate retina. *Annual Review of Neuroscience*, 23, 743-775.

Dowling, J. E. (1987). *The Retina: An Approachable Part of the Brain*. Cambridge, MA, USA: Belknap Press.

Funt, B. & Cardei, V. (1999). Bootstrapping color constancy. *SPIE Electronic Imaging ’99*.

Lanyon, L. H. & Denham, S. L. (2004). A model of active visual search with object-based attention guiding scan paths. *Neural Networks*, 873-897.

McLaughlin, D., Shapley, R. & Shelly (2003). Large-scale modeling of the primary visual cortex: influence of cortical architecture upon neuronal response. *Journal of Physiology-Paris*, 97, 237-252.

Melcher, D. (2007). Predictive remapping of visual features precedes saccadic eye movements. *Nature Neuroscience*, 10, 903-907.

Moore, T. & Fallah, M. (2004). Microstimulation of the frontal eye field and its effects on covert spatial attention. *Journal of Neurophysiology*, 91, 152-162.

Pascual-Leone, A. & Walsh, V. (2001). Fast backprojections from the motion to the primary visual area necessary for visual awareness. *Science*, 292, 510-512.

Riesenhuber, M., & Poggio, T. (1999). Hierarchical models of object recognition in cortex. *Nature Neuroscience*, 2, 1019-1025.

Sceniak, M. P., Ringach, D. L., Hawken, M. J. & Shapley, R. (1999). Contrast's effect on spatial summation by macaque V1 neurons. *Nature Neuroscience*, 2, 733-739.

Simoncelli, E. P. & Heeger, D. J. (1998). A model of neuronal responses in area MT. *Vision Research*, 38, 743-761.

Walther, D. & Koch, C. (2006). Modeling attention to salient proto-objects. *Neural Networks*, 19, 1395-1407.

A SIMULATION SETUP FOR COMMUNICATION HARDWARE IN THE LOOP EXPERIMENTS

Markus Sauer and Florian Zeiger
Zentrum für Telematik e.V., Allesgrundweg 12, Gerbrunn, Germany
{*markus.sauer,florian.zeiger*}*@telematik-zentrum.de*

Klaus Schilling
Department of Computer Science VII, University of Würzburg, Am Hubland, Würzburg, Germany
schi@informatik.uni-wuerzburg.de

Keywords: Networked mobile robots, Communication, Simulation.

Abstract: Simulations are a very powerful tool in robotics to design and verify new algorithms before doing time-consuming tests with real hardware. Nowadays, a lot of very realistic simulation environments are available to simulate robot kinematics and dynamics and any type of multi-robot systems in a virtual physical environment. Unfortunately, the communication in these simulations is often only considered in a very simplified matter, although the characteristics of a real communication link are very complex and might have a strong influence on the performance of a multi-robot algorithm. This contribution proposes a setup to perform communication hardware in the loop tests with the 3D simulation environment USARSim. For this setup any communication device which can be connected to a PC architecture like WLAN, UMTS or Bluetooth can be used. A cooperative collision avoidance algorithm is presented as an example which is realized with this setup, while real hardware is used for the communication link between the robots. Finally, the limitations are presented.

1 INTRODUCTION

The progress in the area of telecommunication technology together with the demand of networked mobile robot systems to assist humans in many different areas (e.g. disaster management, security and surveillance, or search and rescue applications) forces the development of multi robot systems which incorporate several autonomy functions like formation driving and obstacle avoidance. Hereby, due to the required flexibility and dynamic communication topology, distributed control algorithms are very desirable. For the development of these mechanisms to control and coordinate swarms of mobile systems or multi robot systems capable simulation or emulation environments are a useful and necessary tool for efficient development and analysis. But the use of simulation environments for networked mobile robot systems also implies some consideration with respect to significance and validity of the simulation. On the one hand the complete dynamics and kinematics of each system must be modeled appropriately. On the other hand, also the available communication link inbetween the robots must be represented in a suitable manner. With respect to the simulation of the dynamics and kinematics of mobile robots in multi robot systems several simulation environment were developed in the recent years. Two well-known examples of the many available simulators are Player/Stage (Gerkey et al., 2003) and USARSim (Carpin et al., 2007). Player is a robot device server to realize multi-robot or sensor-network systems. Stage can be used together with Player and can simulate large populations of robots in a 2D environment. USARSim is based on the famous Unreal Tournament 2004 game engine. It is a general purpose 3D - multi-robot simulator which provides basic physical properties of the robot and the simulated environment which closely match the real implementation of the robots and the real environment. In addition, it is also possible to simulate camera images from cameras inside the simulation. Compared to Player/Stage it is only a simulation without a device server and controller concept like Player. Figure 1 shows a typical environment simulated with USARSim for the virtual RoboCup Rescue league.

With respect to the simulation of the communication link also many approaches and even products are available to be integrated. Of course, the importance of these simulations of communication link technologies is not only limited to the area of multi robot sys-

Figure 1: Typical environment simulated for the virtual RoboCup Rescue with USARSim.

tems. In the area of network testing and evaluation of wireless network systems (Doshi et al., 2007) often the QualNet network simulator is used for the setup of real-time emulations. This simulator is also used in (Xu et al., 2003) for simulations regarding quality of service provisioning in wireless ad-hoc networks, as well as in (Bagrodia et al., 2006), where a systems simulation environment for future combat systems is presented. In the area of networked haptic virtual environments (Sankaranarayanan et al., 2007) used NIST-Net to create realistic Internet-like characteristics in a laboratory setting. NIST-Net (cf. (Carson and Santay, 2003)) is a tool to facilitate testing and experimentation with network code through emulation which can model communication performance characteristics like packet delay, jitter, bandwidth limitations, congestion, and packet loss.

Of course, there exist other powerful simulation tools like NS2 or OPNET. All these simulation environments are very mighty tools which have focused on the simulation of the characteristics of the communication channel. Unfortunately, they are often very complex and time-consuming to operate and most of them cannot be easily integrated with the simulation environments for mobile robot dynamics and kinematics mentioned before. It is also known that the simulation tool itself influences the outcome of a simulation (Liu and Kaiser, 2005). In addition, you need to test the algorithm anyway later with real communication hardware. Currently, in the area of simulation of networked robot systems and robot swarms the simulation of the communication interface is often represented in a very abstract or simplified way. Nevertheless, several publications for networked control systems turned out the importance of the knowledge about the communication characteristics and its influence on the implemented control algorithms. In (Lopez et al., 2006), experiments of closed-loop networked control systems are evaluated focusing specifically on the performance and time delays effects for different compensation actions. In (Wei et al., 2001) stability of networked control systems is investigated for different network-scheduling protocols. Also methods for compensating network-induced delay are presented together with experimental results for networked control systems with packet loss on the communication link. (Walsh et al., 2002) provided an analytical proof of global exponential stability for a novel control network protocol and commonly used statically scheduled access methods. There, the focus is set on communication constraints which are imposed by the network and the performance of the proposed protocol and the statically scheduled protocols are examined in simulations. As above mentioned, the behavior of the communication channel is very important for the analysis and implementation of coordination and distributed control algorithms for networked robotic systems and may influence the behavior of the complete system. Thus, this work proposes an approach how real communication hardware can easily be included into hardware simulation environments - in this case USARSim. The communication hardware is used as in real world applications but nevertheless directly integrated to the algorithms to be analyzed. The environment consisting in a map and the dynamics and the kinematics of the physical entities (mobile robot clients) is provided by the USARSim server. This modular design allows flexible extensions in terms of replacing the simulated robot hardware by real mobile robot hardware which accelerates the development duration of multi robot systems. As the proposed system integrates real wireless communication hardware and standard protocol stacks directly into the simulation an intensive analysis of implemented coordination and control algorithms for robot teams under consideration of the effects of real wireless communication is possible.

The remainder of this work is structured as follows. First, the hardware in the loop setup which integrates the real communication stack in the simulation is introduced. Then an implementation of a cooperative collision avoidance algorithm as example application is presented. Afterwards, the areas and the constraints of the proposed setup are investigated.

2 HARDWARE IN-THE-LOOP SETUP

The objective of the simulation system design is the use of real communication hardware while simulating multiple robots with USARSim. Therefore, the presented system can be divided into three main parts: a local area network segment, the clients, and a wireless communication segment (cf. Figure 2).

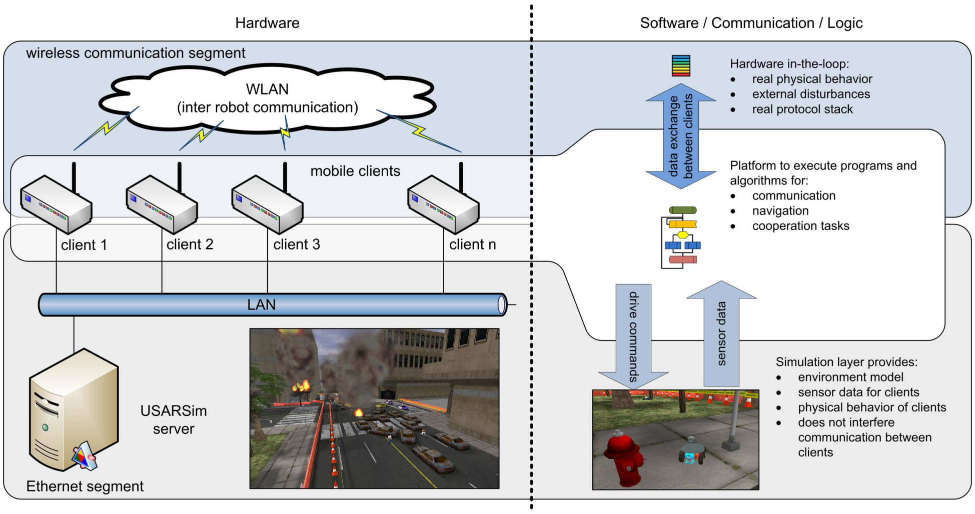


Figure 2: Setup of Hardware in-the-loop Simulation Components.

2.1 Hardware Setup

The local area network segment uses standard Ethernet communication to provide connectivity between the USARSim server and all clients which supports high bandwidth communication with low delays. This connection is used for the exchange of drive commands and sensor data between the USARSim simulation environment and the different clients. This segment represents the indirect communication between each client and its environment, and as it is realized via the Ethernet segment, the direct communication between each client over the wireless link is not interfered. Communication between the clients is not implemented via this link. The clients are equipped with a mini PC architecture with 1200MHz, 1GB RAM, a 8GB compact flash card as hard disk, and Debian Etch as operating system provides a platform to execute the programs and algorithms for navigation and cooperation tasks which should be investigated and analyzed. This mini PC represents the computing power of a single robot. The LAN segment is only used by each client to retrieve environment data from the USARSim server. Communication between the clients is only realized via the wireless communication segment. The wireless segment is based on IEEE 802.11 wireless LAN and represents the communication hardware which is directly integrated into the simulation setup. This communication link is exclusively used for the communication between all clients i.e. robots and human operators. As a standard operating system is used the corresponding protocol stacks are available and also routing mechanisms for wireless ad-hoc networks like OLSR, DSR, or AODV can easily be used.

2.2 Software Components

For each of the hardware components described in Section 2 also dedicated software components are existing. On the simulation layer a USARSim server is running which provides an environment model, the physical behavior of the clients, and sensor data for clients. As only the Ethernet segment is used for the communication between the USARSim server and the clients, the inter client communication via the wireless segment is not affected. On the clients no specific installation for the USARSim simulation and for maintaining the connection to the USARSim server is required. The communication to the USARSim server, and respectively the simulated robots are realized with simple string messages over TCP-socket (Carpin et al., 2007). Each client is running on one of the described mini PC. Basically here, the distributed control algorithms can be implemented. Furthermore, the operating system is also maintaining the communication link to the USARSim server for sensor data acquisition and sending commands. The client PCs are also equipped with WLAN PCMCIA cards supporting the IEEE 802.11 b/g standard. The wireless communication is exclusively used for inter-client-communication which represents one of the key issues of the proposed architecture. In the presented setup all standard protocol versions which are available for the client operating system (e.g. Debian Linux) can be used. As the wireless communication link is exclusively used for inter client communication, the real protocol stacks and real physical behavior of the link allows for meaningful hardware in-the-loop simulations. Thus, the navigation, coordination and cooperation algorithms which should be analyzed are exchanging data via communication links with a realistic behavior - including external disturbances.

2.3 The Simulation System Design

This system setup is designed as simulation environment for network control systems and scenarios of robots or robot formation driving with real IEEE 802.11 wireless LAN communication. In this work, one client represents the formation leading robot and the other three clients are robots which should keep a certain formation. The leading robot sends its position data with a frequency of 10Hz to the other team members via the wireless link. A communication from the team members back to the leader is not present. The communication between the USARSim server and each client uses the standard USARSim interface based on TCP connections. All robots run the same distributed cooperative collision avoidance algorithm while moving to their respective goal points.

3 EXAMPLE: COOPERATIVE COLLISION AVOIDANCE

Typically, distributed control algorithms for robotic networks (Bullo et al., 2008) often assume a certain simplified model of the communication channel. Here, a setup is proposed to test these control algorithms with a real communication stack. As application example for this contribution, a cooperative collision avoidance control algorithms based on the concepts of (Stipanović et al., 2007) is used.

In the example scenario, a group of n mobile robots should move through an environment without colliding with objects in the environment or with each other. There is no central instance coordinating the movement of the robots. In the shown simulation mobile robots with differential drive are used. They are equipped with a simulated laser range finder for obstacle detection. The laser range finder has a field of view of 180 degree and is mounted to the front of the robots. The leader robot $a_{i=1}$ drives a rhombus in this environment and continuously sends its pose to the other following $n-1$ robots. The robots $a_{i\in\{2..n\}}$ receive this pose over the wireless communication segment and set their own new goal pose relative to the pose of the received leader pose. Thus, the formation shown in Fig.3 is established in equilibrium of the controller. Any other logical communication topology can be realized with this kind of setup e.g. robot one can only communicate with robot two and robot three and four can only communicate with robot two. This is especially interesting for the investigation of the system behavior of distributed algorithms with communication constraints.

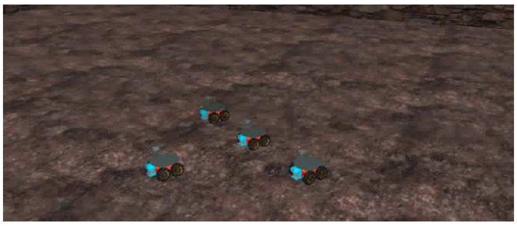


Figure 3: Relative positioning of the robots in formation.

For the presented example application, the mobile robots are modeled with the kinematics of a differential drive robot as first order system (cf. equation 1);

$$\begin{aligned} \dot{x}_i &= v_i \cdot \cos\Theta_i \\ \dot{y}_i &= v_i \cdot \sin\Theta_i \\ \dot{\Theta}_i &= \omega_i \end{aligned} \tag{1}$$

x_i, y_i, and Θ_i denote the pose of the robot i. v_i is the translational velocity and ω_i the turn rate. On each of the robots a combination of the following position controller and a controller for obstacle avoidance is implemented. The controller switches between different behaviors depending on the current conditions. Without obstacles in the defined obstacle avoidance zone and the robot's orientation is not towards the goal ($\Theta_i \neq \Theta_{gi}$) the following controller applies:

$$\begin{aligned} \dot{\Theta}_i = -(\Theta_i - \Theta_{gi}) &= \frac{r_i}{L_i}(u_{ri} - u_{li}) \\ \Rightarrow (u_{ri} - u_{li}) &= -\frac{L_i}{r_i}(\Theta_i - \Theta_{gi}) \end{aligned} \tag{2}$$

r_i denotes the radius of the i-th robots' wheels, L is the length between the wheels, u_{ri} is the left wheel speed, u_{ri} is the right wheel speed respectively and Θ_{gi} is the desired orientation towards the currently defined goal.

If the robot is oriented towards the goal $(\Theta_i - \Theta_{gi}) < t_o$ (t_o - threshold for accuracy of orientation of robot towards goal), v_i is aligned with the straight line between the robot's position and the goal position. Therefore, v_i only applies to $\dot{x}_i$ in the robot coordinate frame $\dot{x}_i = v$ and $\dot{y}_i = 0$. The following controller can be applied:

$$\begin{aligned} v_i = -x_i &= \frac{r_i}{2}(u_{li} + u_{ri}) \\ \Rightarrow u_{ri} = u_{li} &= -\frac{x_i}{r_i}\Theta \end{aligned} \tag{3}$$

In the robot coordinate frame x_i is under the above given conditions equal to the distance between robot and current goal and it becomes zero if the desired goal is reached. If an obstacle is in the defined sensing range, the controller is adapted according to the following rules: First a vector F_{oi} is calculated. F_{oi} points in opposite direction of the nearest obstacle to the robot and its length increases indirect proportional with the distance to the next obstacle. Then this vector is combined with the normalized vector in goal direction F_{gi} to a new goal direction vector incorporating an obstacle avoidance component and a new desired heading Θ_{oai} (cf. Figure 4):

$$F_{oai} = (F_{oi} + F_{gi}) \tag{4}$$

$$\Theta_{oai} = \arctan 2(y_{oai}, x_{oai}) \tag{5}$$

Finally, this value is inserted in the controller defined in Equation 2:

$$\dot{\Theta}_i = -(\Theta_i - \Theta_{oai}) = \frac{r_i}{L_i}(u_{ri} - u_{li})$$

$$\Rightarrow (u_{ri} - u_{li}) = -\frac{L_i}{r_i}(\Theta_i - \Theta_{oai}) \tag{6}$$

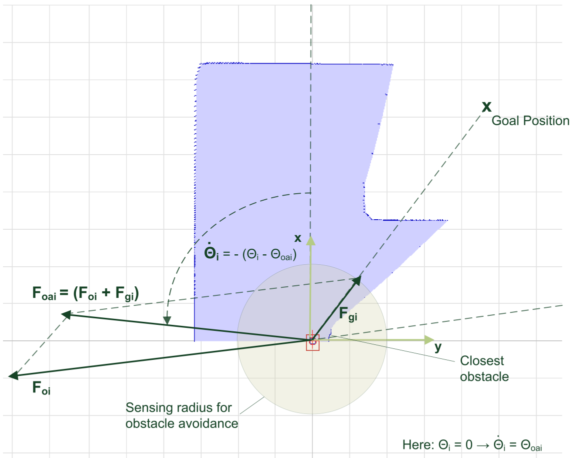

x
Goal Position
x
$\dot{\Theta}_i = -(\Theta_i - \Theta_{oai})$
$F_{oai} = (F_{oi} + F_{gi})$
F_{gi}
y
Closest obstacle
F_{oi}
Sensing radius for obstacle avoidance
Here: $\Theta_i = 0 \rightarrow \dot{\Theta}_i = \Theta_{oai}$

Figure 4: Overview of the different values for the obstacle avoidance controller.

After orientation towards Θ_{oai} the robot always moves for a small time period in this direction to avoid oscillations in reorienting due to the limitation of the obstacle sensing to 180 degrees. This translational movement is only done in cases where definitely no collision can occur.

The experiments with this cooperative collision avoidance algorithm were done with $n = 4$ robots. The results can be seen in Fig. 5 and Fig. 6. Fig. 5 shows how the four robots move with respect to each other over time while the three robots follow the leader robot driving a predefined rhombus trajectory for a certain experiment time. In each plane at a certain time the position of the robots at this times can be seen. The 3D plot of the trajectories shows the reorientation of the formation at the edges of the rhombus over time and it can be seen that there was now collision because none of the trajectories is touching or crossing each other. Fig. 6 shows the minimum distance inside the group of mobile robots. The relative position of the three following robots was designed to have a distance minimum of $1.7m$ between all robots when they are moving in perfect formation. In addition the robots should never get closer then $0.4m$. Fig. 6 shows that the algorithm satisfies these requirements. The peaks in the graph occur always when the formation is reorienting at the edges of the rhombus driven by the leader robot.

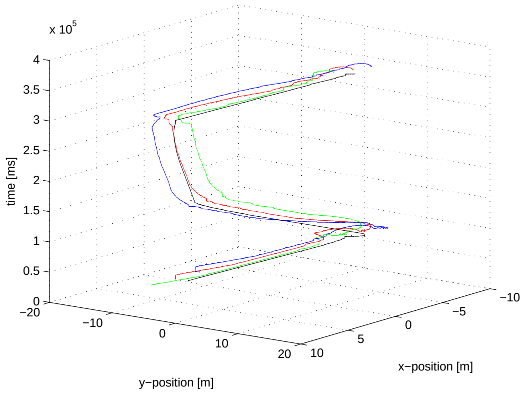


Figure 5: Position of each robot while driving in formation.

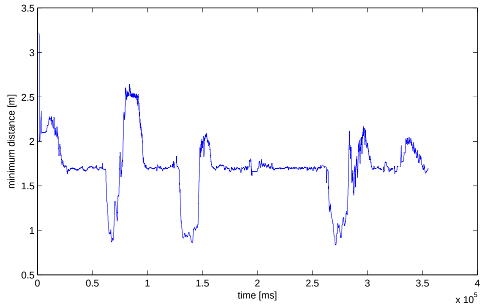


Figure 6: Minimum distance occurring between all robots inside the formation.

4 APPLICATION AREAS AND CONSTRAINTS

This simulation setup is designed for real communication hardware in-the-loop simulations of networked robotics scenarios. The advantage of this special setup is, that the real communication protocol stack is used which saves the very complex simulation of the protocol stack and the physical behavior of the link. A well suited application area of this system is in the simulation of swarms, multi-robot teams, and formation driving. In these scenarios, control algorithms can be tested and evaluated under the influence of real communication link behavior (limitations in medium access etc.) and also different communication protocols can be tested easily. Of course, also some limitations exist. As the hardware nodes are located quite close to each other, long distance communications and the consequential channel behavior cannot be simulated. Nevertheless, for detailed simulations of the interaction of communication protocols, control engineering and the underlying system in multi-robot teams and formation scenarios, the presented architecture is suites very well and allows an easy and fast setup of significant simulations.

5 CONCLUSIONS

In order to simulate the behavior of networked multi-robot systems in general a model for the communication channel has to be to implemented and verified. In most cases this is only possible with simplifications and limitations and the simulated channel is not representing a real communication channel anymore. Therefore, the conclusions drawn from the simulation of the tested algorithm might not be as meaningful as desired.

The presented approach offers a possibility to realize easily a meaningful simulation with real communication hardware for network robotic scenarios. It provides the exact behavior of the complete, complex communication stack without any approximation or simplifications. Thus, the behavior of multi-robot algorithms can be directly investigated with all the changes in the communication and data flow between the robots. In the combination with a simulator like USARSim is is possible to simulate network robotic systems with basic physics and real communication.

The setup of a hardware communication in the loop simulation is much easier than the setup of a meaningful communication channel simulation combined with a multi-robot simulation. There are even less uncertainties in the behavior of the system when you later go to real hardware.

Therefore, it is very easy to test the behavior of algorithms for typical applications of network control systems like teleoperation of robots or robot formation driving with a real communication channel before going to the real hardware. Due to the standardized interfaces which are used, such kind of setups also allow for an easy evaluation of different type of wireless communication systems like e.g. WLAN, UMTS, HSDPA/HSUPA, Bluetooth, WiMax. Especially testing of swarm behavior is very meaningful, because like in the real system naturally the nodes in the communication hardware in the loop simulation are very close to each other.

REFERENCES

Bagrodia, R., Tang, K., Goldman, S., and Kumar, D. (2006). An accurate, scalable communication effects server for the fcs system of systems simulation environment. In *Proceedings of the Winter Simulation Conference*.

Bullo, F., Cortés, J., and Martínez, S. (2008). *Distributed Control of Robotic Networks*. Princeton Series in Applied Mathematics. Princeton University Press, Princeton, NJ.

Carpin, S., Lewis, M., Wang, J., Balakirsky, S., and Scrapper, C. (2007). Usarsim: a robot simulator for research and education. In *2007 IEEE International Conference on Robotics and Automation (ICRA'07)*.

Carson, M. and Santay, D. (2003). Nist net: a linux-based network emulation tool. *SIGCOMM Computer Communication Review archive*, 33(3):111 – 126.

Doshi, S. R., Lee, U., and Bagrodia, R. L. (2007). Wireless network testing and evaluation using real-time emulation. *ITEA Journal of Test and Evaluation*, 28(2).

Gerkey, B., Vaughan, R. T., and Howard, A. (2003). The player/stage project: Tools for multi-robot and distributed sensor systems. In *11th International Conference on Advanced Robotics (ICAR)*, pages 317–323, Coimbra, Portugal.

Liu, C. and Kaiser, J. (2005). A survey of mobile ad hoc network routing protocols. Technical report, University of Magdeburg, TR-4: Middleware for Network Eccentric and Mobile Applications (MINEMA).

Lopez, I., Piovesan, J., Lee, C. A. D., Martinez, O., Spong, M., and Sandoval, R. (2006). Practical issues in networked control systems. In *Proceedings of the American Control Conference*.

Sankaranarayanan, G., Potter, L., and Hannaford, B. (2007). Measurement and emulation of time varying packet delay with applications to networked haptic virtual environments. In *RoboComm - First International Conference on Robot Communication and Coordination*.

Stipanović, D. M., Hokayem, P. F., Spong, M. W., and Šiljak, D. D. (2007). Cooperative avoidance control for multiagent systems. *Journal of Dynamic Systems, Measurement, and Control*, 129(5):699–707.

Walsh, G., Hong, Y., and Bushnell, L. (2002). Stability analysis of networked control systems. *IEEE Transactions on Control Systems Technology*, 10(3):438–446.

Wei, Z., Branicky, M., and Phillips, S. (2001). Stability of networked control systems. *Control Systems Magazine, IEEE*, 21(1):84 99.

Xu, K., Tang, K., Bagrodia, R., Gerla, M., and Bereschinsky, M. (2003). Adaptive bandwidth management and qos provisioning in large scale ad hoc networks. In *Proceedings of MILCOM'03*.

CO-DESIGN FOR WIRELESS NETWORKED CONTROL OF AN INTELLIGENT MOBILE ROBOT

Amine Mechraoui, Zeashan Hameed Khan, Jean-Marc Thiriet and Sylviane Gentil
Gipsa-Lab, Control departement INPG-UJF-CNRS, BP 46, 38402 Saint Martin d'Hères, France
amine.mechraoui, zeashan-hameed.khan, jean-marc.thiriet, sylviane.gentil@gipsa-lab.inpg.fr

Keywords: Autonomous mobile robots, Networked control systems, Wireless network, 802.15.4 protocol, Horizontal handoff, TrueTime simulation.

Abstract: This paper describes a wireless network based control of a Khepera mobile robot moving in a distributed infrastructure. Due to critical dependence on wireless communication, a procedure for reconfiguration of the network is proposed as a possibility to maintain communication between control station and the mobile robot in a successful manner. The network handoff is made under a criterion that takes into account key application dependent performance parameters. The controlled system and the communication network are simulated respectively with Matlab/Simulink and TrueTime.

1 INTRODUCTION

Networked control systems (NCS) in mobile robotics are getting very popular today. With the rapid progress in communication techniques, especially the wireless networks, distributed control and decision have become mandatory to reduce the onboard processing overhead. It includes control, decision, obstacle avoidance etc; as it effects the battery consumption in miniature robots with space and weight as key design constraints. However, introducing a wireless network in the control loops presents some disadvantages such as band limited channels, sampling delays and packet dropouts (Hespanha et al., 2007). Furthermore, the mobility of the robot also adds some problems, e.g. increasing the distance between the control station and the robot increases the number of lost packets due to decreased signal strength and increased bit error rate (BER) (Zhu et al., 2005).

The communication architecture in mobile robotics may be centralized, in which case there is a fixed or mobile node, which communicates with all the other nodes, or decentralized, where individual mobile nodes should ideally operate without any central control (Schwager et al., 2007). In the decentralized control scheme, each component solves a part of the problem and shares memory without having a global view of the mission. There is less emphasis on computation than communication. In distributed control systems, communication is an important parameter and individual components don't need to share memory (Martinez et al., 2007). In related research work, many approaches have been used for distributed control of mobile robots. In (Fierro and Lewis, 1996) the dynamic model of the mobile robot is controlled by means of neural networks. In (Aicardi et al., 1995) and (canudas de Wit and Sordalen, 1992), a nonlinear control approach has been introduced. Another research area, related to the hybrid architecture of control for autonomous navigation robots is studied in (Benzerrouk et al., 2008).

The objective of this paper is to define a communication architecture for a mobile robot moving in a 2D space with several fixed stations (wireless infrastructure communication). According to the position of the robot, communication is possible in a specific coverage area with one or several stations (see Fig. 1). When several stations are reachable, the robot will choose one of the stations (Horizontal Handoff (HHO) strategy (Wang et al., 2007)) that can allocate sufficient resources to ensure a good level of communication. This comprises of optimal choice of payload and delay based on distance between robot and station and thus maintaining the necessary Quality of Service (QoS) in order to ensure that the Quality of Control (QoC) is sufficient (distributed control). When there is no station in the reachable space, the robot will have to be absolutely autonomous (embedded control), maintaining a sufficient QoC despite a degradation of the communication QoS. The QoC in wireless

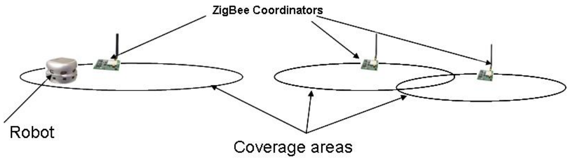


Figure 1: Problem description.

NCS is defined as the performance delivered by each closed-loop operation. Stability is the main property that must be guaranteed but control error and response time are also important to analyse.

In the literature, many researchers have proposed a HHO strategy. They proposed making a decision by taking into account the Received Signal Strength (RSS), the power consumption and the cost of communication (Chen et al., 2004). We propose to add the packet dropouts caused by the propagation delay or the distance and orientation between the robot and the station, which have a consequence on the QoC.

The paper is organized as follows. The second section presents a brief description of the model of the Khepera robot, notably the kinematic model and the dynamic one. In the third section, the controller design is described and simulations of the tracking trajectory are presented. Section 4 presents the control over ZigBee wireless network and the influence of the integration of this network on control performance. After that, the proposed HHO architecture is described with one, two and three stations. Finally, the conclusion and perspectives are presented.

2 MODELING AND CONTROL

This section presents the study of a unicycle Khepera robot (Lambercy and Caprari, 2007). Consider a unicycle robot (Khepera) as shown in Fig. 2. Let x, y and θ be the state variables where $x \in \Re$ and $y \in \Re$ are the Cartesian coordinates, $\theta \in [0, 2\pi[$ is the robot's orientation with respect to the X-axis. We consider 'v' and 'ω' respectively as the linear and the angular velocities of the robot. The kinematic of the robot can be modeled as

$$\dot{x} = v\cos\theta, \;\; \dot{y} = v\sin\theta, \;\; \dot{\theta} = \omega \tag{1}$$

The kinematics model of the mobile robot has two control inputs ω_{left} and ω_{right} i.e. the left and right wheels velocities. These are related to the linear velocity v and the angular velocity ω of the robot according to the following equations

$$V_{right} = v + R\omega, \;\; V_{left} = v - R\omega \tag{2}$$

where R is half the distance between the two robot's wheels. The dynamic model of the robot wheels is characterized by the equations of the DC motors driving the wheels. They are represented by a first order model

$$\frac{\omega^*_{(left,right)}}{U} = \frac{K}{\tau s + 1} \tag{3}$$

where U is the voltage applied to the motor and $\omega^*_{(left,right)}$ are angular velocities generated by each motor. τ is the time constant ($\tau = 0.63s$) and K is the gain ($K = 5.3$).

2.1 Controller Design

The objective of this section is to present how to control the robot to track any trajectory. Two levels of controllers are required. The first one is needed to control the angular velocities of the motors. PI controllers are implemented. The second one controls the linear and angular velocities of the robot. Let us consider the controller presented in (Toibero et al., 2007), where the robot can reach a desired target point $[x_d \; y_d \; \theta_d]$. Errors are defined as

$$\tilde{x} = x_d - x, \;\; \tilde{y} = y_d - y \tag{4}$$

and the tracking error (Eq. 5a) and the orientation error (Eq. 5b) are calculated as

$$d = \sqrt{\tilde{x}^2 + \tilde{y}^2} \tag{5a}$$

$$\tilde{\theta} = \theta_d - \theta = \tan^{-1}(\tilde{y}/\tilde{x}) - \theta \tag{5b}$$

According to (Toibero et al., 2007), the following control actions are defined

$$v = \frac{v_{max}}{1 + |d|} d\cos(\tilde{\theta}) \tag{6a}$$

$$\omega = \frac{v_{max}}{1 + |d|}\cos(\tilde{\theta})\sin(\tilde{\theta}) + K_{\tilde{\theta}}\tanh(k_\theta \tilde{\theta}) \tag{6b}$$

where v_{max} is the maximum linear velocity that the robot can reach ($v_{max} = 0.3m/s$) and $K_{\tilde{\theta}}, k_\theta$ are constants. Those controllers are stable according to (Toibero et al., 2007) using the Lyapunov candidate function V_t.

$$V_t = \tilde{\theta}^2/2 + d^2/2 \tag{7}$$

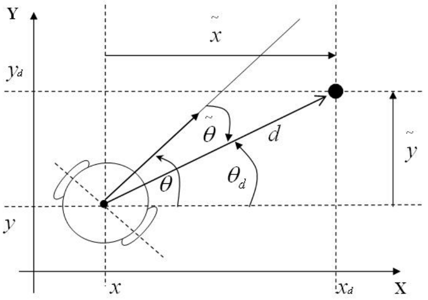


Figure 2: Robot Model.

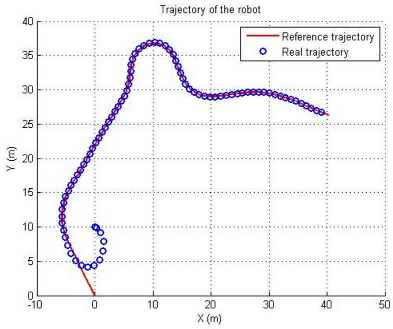


Figure 3: Real trajectory of the robot.

Fig. 3 shows the reference trajectory with initial conditions $[X_0, Y_0, \theta_0] = [0, 10, 0]$ and the real trajectory of the robot using those controllers. X, Y and θ are measured each sampling time ($T_s = 0.03s$). These results are obtained with Matlab/TrueTime. The PI controllers are discretized and the voltage applied to the motor is obtained with a zero order hold. The trajectory is known in advance and the references x_d, y_d and θ_d are updated every 0.28s.

3 CONTROL OVER NETWORK

In this section, the controller is digitized and Zigbee wireless network is integrated (see Fig. 4). The effects of network on control system are analyzed.

3.1 WPAN 802.15.4

ZigBee is a specification for small, low-power digital radios based on the IEEE 802.15.4 standard for wireless personal area networks (WPANs). The low cost allows the technology to be widely deployed in wireless control and monitoring applications. The low power-usage allows longer life with smaller batteries. The mesh networking provides high reliability and larger range. The main reasons for adopting this wireless network in our application is its low power consumption and its ad-hoc networking capabilities.

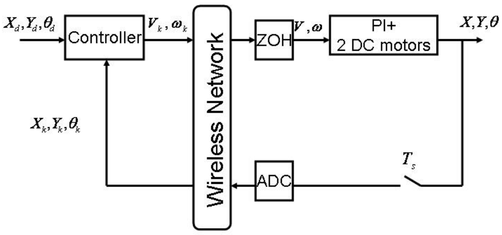


Figure 4: Control over Zigbee network.

3.2 Control Over Zigbee

Control feedback loops are closed through a real-time network as shown in Fig. 4.
To perform this study, the following conditions are considered:

- A bit rate of 250 kb/s for Zigbee in the physical layer.
- The sensor flow and the controller flow use 248 *bits* each. The sensor flow uses 3.3% and the controller one uses 3.3% of the network capacity with a sampling period T_s equal to 30 *ms*.

TrueTime simulator is used to simulate Zigbee Network. Two tasks are programmed, the first is the controller task that generates the controller flow and the second one is a periodic sensor task that generates sensors flow. The controller task is event-triggered, which means that the controller calculates and sends the control signals V_k and ω_k when it has received all measures X_k, Y_k and θ_k. With the TrueTime Simulation of ZigBee Network, there is only the CAP period (Contention Access Period which uses CSMA/CA protocol), therefore there is no priority mode like CFP (Contention Free Period) which allows Guaranteed Time Slots (GTS) (see (van den Bossche et al., 2007) for more information about 802.15.4 protocol).
The sensor and control data are critical to keep the stability of the system, hence packet losses are undesirable. Practically, loss of packets cannot be entirely eliminated in wireless networks. A lot of methods e.g. Forward Error Correction (FEC) (Kurose and Ross, 2004) are proposed to reduce the probability of packet losses. The retransmission of lost packets is proposed too but for a real-time application, this is an inappropriate solution. When a packet has been discovered to be lost and is retransmitted, the sensors state evolve to a newer one and thus the retransmitted packet will be based on the old information and the calculated control will be wrong. The solution to ensure the best control is to do *over sampling*, with a sampling rate higher than what is needed, which therefore makes control more tolerant to packet losses. However, this solution increases the Use Request Factor (URF) and causes more packet delays and losses (Mechraoui et al., 2008). The minimal sampling period to ensure a good QoC is 0.3 s in our example. To increase the QoC over wireless network and decrease the probability of losing packets of critical data, the sampling period of 30 *ms* is required. Fig. 5 shows that the robot can communicate with the station only within the coverage range of the station. When it losses the communication, it keeps the last value of the linear and angular velocities and it deviates from the reference trajectory. The progress of WPAN and the util-

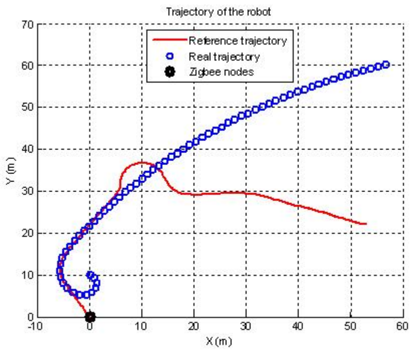


Figure 5: Trajectory of the robot with one Zigbee station.

isation of this kind of wireless networks allow the robot to find another station (PAN) to communicate for control. Therefore, we assume that we have different stations that the robot can communicate with. The next section explains how to change stations to ensure sufficient QoS.

4 STATION SELECTION

4.1 Handoff Decision

For several stations, mobility management is proposed through redundancy in network communication by searching for the best available protocol based on a cost function. Horizontal and vertical handoff can be proposed. A horizontal handoff (HHO) is the link transfer between two network access points that use the same network technology while vertical handoff (VHO) is between two network access points using different network technologies. (Chen et al., 2004) describes a smart decision model to decide the best network interface and best time for VHO. (Stevens-Navarro and Wong, 2006) gives a comparison between four different VHO decision algorithms w.r.t to bandwidth and delay for different traffic classes. (Wang et al., 1999) describes a policy-enabled handoff system that allows users to express policies, based on the estimation of current network conditions and determining the best network at any moment by characteristics e.g. cost, performance and power consumption. (Nkansa-Gyekye and Agbinya, 2007) proposes a distributed additive weighting based VHO mechanism to reduce the processing overhead in the mobile terminal by delegating the calculation of handoff metrics for network selection to the visiting networks. (Angermann and Kammann, 2002) evaluates different structures for cost functions and presents a simulation for the influence of various parameters.

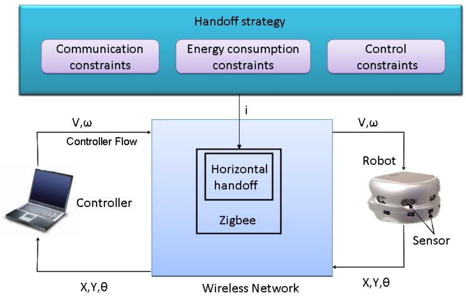


Figure 6: Network HHO strategy for a mobile robot control.

The challenge in handoff is to maintain the application session alive while the physical connection interface is changed. A generic weighted 'Network Rating Function' evaluation for each network is based on

$$NRF_i = W_{RSSI}N_{RSSI,i} + W_{NL}N_{NL,i} + W_D N_{D,i} \quad (8)$$

$$W_{RSSI} + W_{NL} + W_D = 1 \quad (9)$$

where W_{RSSI}, W_{NL} and W_D are weighting functions of RSSI, network load (NL) and delay (D) respectively. $N_{RSSI,i}$, $N_{NL,i}$ and $N_{D,i}$ indicate the scores of interface. NRF_i is between 0 and 1. W_i is the weight of the factor i, which emphasizes the importance of each contributing factor and $N_{i,j}$ represents the normalized score of the interface $'j'$ for factor $'i'$.

The embedded logic in the mobile robot, assigns different "weights" to the handoff decision parameters in order to determine the level of importance of each parameter for the information to be exchanged between neighbors. In our case $W_{RSSI} > W_{NL} > W_D$. The best station connection interface at any given moment is then chosen as the one that achieves the highest score among all candidate interfaces.

In our scenario, factors within $N_{i,j}$ include QoS parameters i.e. the RSSI, the network load (NL) and Delay or latency (D) of the candidate network. Now we calculate NRF_i and NRF_j to choose accordingly (which ever is greater) the station that maximizes the

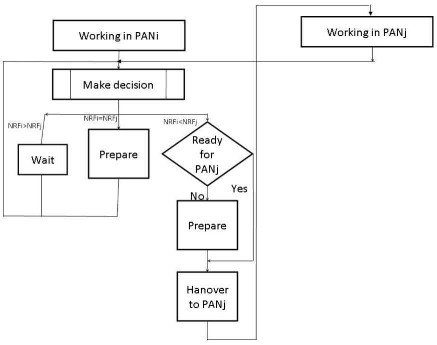


Figure 7: HHO state diagram.

decision criteria (Fig. 7).

Additionally, there is a corresponding function for each term $N_{RSSI,i}$, $N_{NL,i}$, and $N_{D,i}$, and the ranges of the functions are bounded between 0 and 1. The functions are given in Eq. (10).

$$N_{RSSI,i} = \frac{e^{P_r}}{e^{P_s}} \tag{10a}$$

$$N_{NL,i} = 1/e^{NL} \mid 0 \leq NL \leq 1 \tag{10b}$$

$$N_{D,i} = \begin{cases} 1 & \text{if } S_i \text{ is the current station \& RSSI} > -48\text{dBm} \\ 0 & \text{if not} \end{cases} \tag{10c}$$

In our case, the transmitted power is equal to $-3dBm$ and the receiver signal threshold is equal to $-48dBm$.

The RSSI is calculed as follow

$$P_r = \frac{1}{d^{\alpha}} P_s \tag{11a}$$

$$RSSI(dBm) = 10log(P_r(mW)) \tag{11b}$$

where P_r and P_s are the power in mW received and sent respectively. d is the distance between the two nodes in meters, and α is a parameter that can be chosen to model different environments.

Fig. 8 shows the trajectory of the robot with changes in station if necessary (QoS is the best to ensure the best QoC). The stations positions are

- For the first station (S_1) (0, 5) (m)
- For the second station (S_2) (-3, 30)
- For the last station(S_3) (40, 35).

For this experiment, the weight values are $W_{RSSI} = 0.8$, $W_{NL} = 0.15$ and $W_D = 0.05$.

We assume that S_1 and S_2 have a common coverage area, and S_3 is far. According to the initial position of the robot (here $(X_0, Y_0, \theta_0) = (0,0,0)$) the robot communicates first with $S1$. It continues its trajectory until the robot detects the second station. In this case, Khepera executes Algorithm 1.

To understand this algorithm, symbols are used. Ⓡ is the mobile robot, S_i is the current station, $S_{\{1,2,..k\}}$ are all the stations and $S_{\{m,...,n\}}$ are the stations that are detected by the robot. We note the fact that the robot detects a set of stations by the symbol $\bowtie$.

When the robot is out of range of all stations, it keeps the last values of the angular and the linear velocities which is not a good solution because it can result in an important error with respect to the reference trajectory and cause problems (see Fig.8). We propose in this case to change the controller and use an embedded one.

Figures 10 and 11 show the change of stations. In Fig.10, the simulation is made with NL=0 for all stations. The change from S_1 to S_2 depends thus on the distance between the robot and the station. The robot changes the station only if the RSSI of the current station is poor. In Fig.11, the simulation is made under the assumption that the $NL_{S_1} = 0.8$ which means that the network is 80% loaded and $NL_{S_2} = 0.2$. In this scenario the robot decides to change the station according to the RSSI of both stations and the network

Algorithm 1: Decision algorithm.

```
if Ⓡ $\bowtie \{S_m, ..., S_n\} \mid m \in \{1, ..., k\}$ and
$n \in \{1, ..., k\} \mid m \neq n$ then
    calculate $NRF$ for all $S_m, ...S_n$
    for all $i$ and $l \in \{m, ..., n\}$ do
        if $NRF_{S_i} < NRF_{S_l}$ then
            $i = l$
        else if $NRF_{S_i} > NRF_{S_l}$ then
            $i = i$
            Wait
        else
            $i = i$
        end if
    end for
end if
```

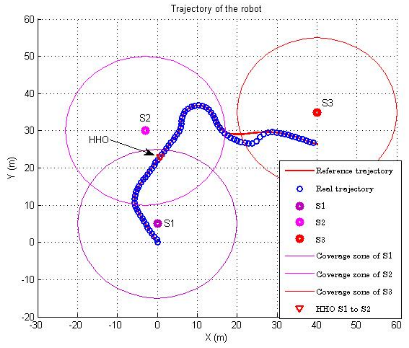


Figure 8: Trajectory of the robot with HHO with NL=0.

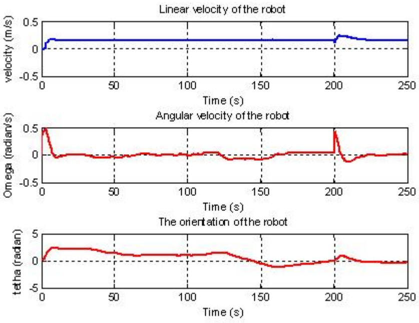


Figure 9: Evolution of the linear and angular velocities.

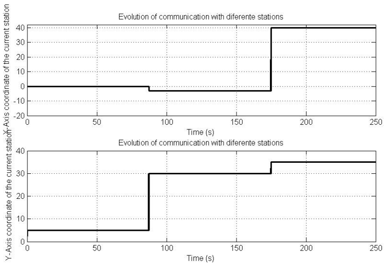


Figure 10: Evolution of communication with different stations with HHO with NL=0 (corresponds to Fig.8).

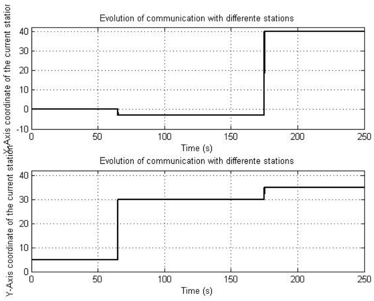


Figure 11: Evolution of communication with different stations with HHO with $NL_{S_1} = 0.8$ and $NL_{S_2} = 0.2$.

load of each station. Comparing Fig.10 and Fig.11 the HHO of the station is quite sensitive to parameter NL.

4.2 Control Decision

As the robot moves outward towards another control station, the signal strength decreases. The low value of RSSI is sensed by the mobile robot, which then requests the station for an estimation of next way-point. The control station sends the trajectory information as well as broadcasts the "standby" message to all control stations in the infrastructure to minimize the time of connection of robot for the next cell. Thus, a PAN slot is reserved in advance for the coming robot as the trajectory is already estimated. This proposition permits to maintain a sufficient QoC despite a degradation of the QoS.

The embedded control design is the same than the distributed one. The only difference is that the sampling period is changed. The controller computes control signals each 0.3 s. This change is very important to reduce the energy consumption and also the execution time. Algorithm 1 and 2 is executed ("*poor*" in Algorithm 2 means that the RSSI of the current station comes near the receiver signal threshold). Fig.12 show the simulation when the control is switched within the non covered area.

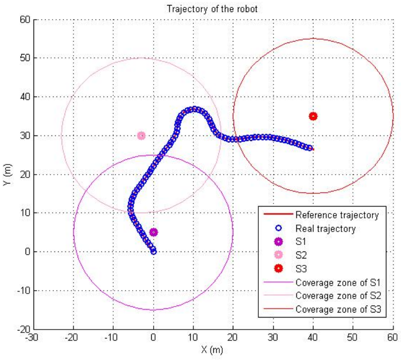


Figure 12: The trajectory of the robot with switching controller.

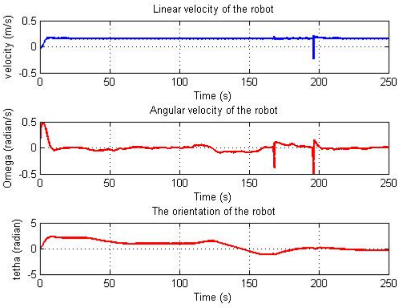


Figure 13: Evolution of linear and angular velocities of the robot with switching controller.

5 CONCLUSIONS

This work aims to study, in a co-design approach, the influence of a wireless network QoS on the QoC of a teleoperated robot. A wireless network based control of a unicycle mobile robot in a distributed infrastructure mode is described. The simulations are performed using Matlab/TrueTime toolbox. The problem faced was a loss of communication when the mobile robot moves out of range or there is an excessive network load that prohibits successful communication.

The first strategy consists of adapting the network QoS to the control requirements. An algorithm for reconfiguration by HHO is proposed as a solution to maintain communication between control stations and the mobile robot in multi stations scenarios. The HHO is made with a criterion that takes into account

Algorithm 2: Decision algorithm.

```
if $NRF = poor$ and $\circledR \bowtie S_i$ then
    Switch controller
end if
```

the QoC for the robot as well as the QoS of wireless network.

The second strategy deals with the reconfiguration of the control, for the robot to be autonomous, when the communication link is out of order. Therefore, if the robot is out of range of all stations, the control mode is switched to embedded control, increasing the sampling period to reduce computations and the robot is completely autonomous. Integration of WLAN could be a choice for extended zone coverage for mobile robots.

In the future work, a quantification of QoS and QoC will be dealt. A combination of infrastructure and adhoc architecture will also be investigated in order to maintain sufficient QoC in multi robots perspective.

ACKNOWLEDGEMENTS

This work is partially supported by the Safe NeCS project funded by the French Agence Nationale de la Recherche under grant ANR-05-SSIA-0015-03.

REFERENCES

Aicardi, M., Casalino, G., Bicchi, A., and Balestrino, A. (1995). Closed loop steering of unicycle like vehicles via Lyapunov techniques. In *IEEE Robotics and Automation Magazine*, volume 2(1), pages 27–35.

Angermann, M. and Kammann, J. (2002). Cost metrics for decision problems in wireless ad-hoc networking. In *IEEE CAS Workshop on Wireless Communications and Networking, Pasadena, California, USA*.

Benzerrouk, A., Adouane, L., Martinet, P., and Andreff, N. (2008). Toward an hybrid control architecture for a mobile multi-robot systems. In *Proceeding of the 3^{rd} National Conference on Control Architectures of Robots, Bourges, France*.

canudas de Wit, C. and Sordalen, O. (1992). Exponential stabilization of mobile robots with non-holonomic constraints. In *IEEE Trans. Automatic Control*, volume 37, pages 1792–1797.

Chen, L.-J., Sun, T., Chen, B., Rajendran, V., and Gerla, M. (2004). A smart decision model for vertical handoff. In *Proceedings of the 4^{th} International Workshop on Wireless Internet and Reconfigurability, Athens, Greece*.

Fierro, R. and Lewis, F. (1996). Practical point stabilization of a nonholonomic mobile robot using neural networks. In *Proceeding Conference on Decision and Control, Kobel, Japan*.

Hespanha, J., Naghshtabrizi, P., and Xu, Y. (2007). A survey of recent results in networked control systems. In *Proceeding of IEEE special issue on Technology of Networked Control Systems*, volume 95, pages 138–162.

Kurose, J. and Ross, K. W. (2004). *Computer Networking: A Top-Down Approach Featuring the internet*. Addison Wesley, 3^{rd} edition.

Lambercy, F. and Caprari, G. (2007). Khepera3 user manual version 2.2. Technical report.

Martinez, S., Corts, J., and Bullo, F. (2007). Motion coordination with distributed information. In *IEEE Control Systems Magazine*, volume 27(4), pages 75–88.

Mechraoui, A., Thiriet, J.-M., and Gentil, S. (2008). Control of an intelligent robot system over a wireless network. In *Proceeding of the 23^{rd} IAR Workshop on Advanced Control and Diagnosis, Coventry, United Kingdom*.

Nkansa-Gyekye, Y. and Agbinya, J. (2007). Vertical handoff decision algorithm for umts-wlan. In *Proceeding of the 2^{nd} International Conference on Wireless Broadband and Ultra Wideband Communications, Sydney, Australia*.

Schwager, M., Slotine, J., and Rus, D. (2007). Decentralized, adaptive control for coverage with networked robots. In *Proceeding of IEEE International Conference on Robotics and Automation, Roma, Italy*.

Stevens-Navarro, E. and Wong, V. (2006). Comparison between vertical handoff decision algorithms for heterogeneous wireless networks. In *Proceeding of the 63^{rd} IEEE Vehicular Technology Conference*, volume 2, pages 947–951.

Toibero, J. M., Carelli, R., and Kuchen, B. (2007). Switching control of mobile robots for autonomous navigation in unknown environments. In *Proceeding of IEEE International Conference on Robotics and Automation, Roma, Italy*.

van den Bossche, A., Val, T., and Campo, E. (2007). Prototyping and performance analysis of a qos mac layer for industrial wireless network. In *Proceeding of the 7^{th} IFAC International Conference on Fieldbuses and Networks in Industrial and Embedded Systems, Toulouse, France*.

Wang, H., Katz, R., and Giese, J. (1999). Policy-enabled handoffs across heterogeneous wireless networks. In *Proceeding of the 2^{nd} IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, Louisiana, USA*.

Wang, L., Chen, A., and Chen, H. (2007). Network selection with joint vertical and horizontal handoff in heterogeneous wlan and mobile wimax systems. In *Proceedings of the 63^{rd} IEEE Vehicular Technology Conference, Melbourne, Australia*.

Zhu, H., Yang, Q., and Kwak, K. (2005). Performance analysis of fast handoff with mobility prediction. In *Proceeding of IEEE International Symposium on Communications and Information Technology, Beijing, China*.

RANDOM VARIATES GENERATING METHODS OF TIME-BETWEEN-FAILURES FOR THE REPAIRABLE SYSTEMS UNDER AGE-REDUCTION PREVENTIVE MAINTENANCE

Chun-Yuan Cheng
Dept. of Industrial Engineering and Management, Chaoyang University of Technology
168, Jifong E. Rd., Wufong, Taichung County, 41349, Taiwan
cycheng@cyut.edu.tw

Renkuan Guo
Department of Statistical Sciences, University of Cape Town, Cape Town, South Africa
Renkuan.Guo@uct.ac.za

Mei-Ling Liu
Dept. of Industrial Engineering and Management, National Taipei University of Technology, Taipei, Taiwan
The Office of Academic Affairs, Chaoyang University of Technology, Taichung County, Taiwan
mlleou@cyut.edu.tw

Keywords: Simulation method, Time-between-failure random variates, Preventive maintenance, Age reduction.

Abstract: Based on the theoretical model, a numerical method is usually necessary for obtaining the optimal preventive maintenance (PM) policy for a deteriorating system since the theoretical model becomes complicated when the system’s hazard rate function is changed after each PM. It makes the application of the theoretical model not suitable for real cases. Moreover, the theoretical model assumes using infinite time span to obtain the long-term expected number of failures. Yet, in reality, the deteriorating systems always have a finite life time. Hence, an optimal solution might not be resulted as compared to the infinite time span. Therefore, we consider using the simulation method to obtain a range of the near-optimal PM policy. The critical step of the simulation method for obtaining a near-optimal PM policy is the generation of the random variates (RV). In this research, three methods are developed to generate the required RVs of the time-between-failures (TBF) for the finite-time-span preventive maintenance model with age reduction effect. It is found that there are no significant differences among three proposed RV generating methods when comparing the dispersion of the generated RV’s. However, the rejection method is the simplest method for obtaining the near-optimal PM policies. Examples of the near-optimal PM policies are also presented in this paper.

1 INTRODUCTION

Based on the theoretical model, a numerical method is usually necessary for finding the optimal preventive maintenance (PM) policy for a deteriorating system since the theoretical model becomes complicated when the system’s hazard rate function is changed after each PM. It makes the application of the theoretical model not suitable for real cases. Furthermore, by the theoretical model, the optimal policy is obtained based on the long-term failures occurrence under the assumption of the infinite time span. Yet, in reality, the life time of a system is always finite. Hence, the optimal solution from the theoretical model may not suitable for a single system with finite life time. In practical, a near-optimal PM policy might be good enough for the real applications. In order to obtain a near-optimal PM policy for the real situations, the simulation method is applied to generate random variates (RV) of the time between failures (TBF). However, recent literature survey has shown that little research has been done to obtain a near-optimal PM policy by using the simulation method.

The critical step of the simulation method for obtaining a near-optimal PM policy is the generation of the random variates (RV). In this research, three methods are developed to generate the required RVs of the time-between-failures (TBF) for the finite-time-span PM model with age reduction effect.

Based on the simulation method developed by Cheng (2005), the first proposed method applies the inverse transformation method to generate the random variates (RV) of the time between failures (TBF) for a PM model with age reduction effect. The algorithm assumes that the occurrence time of the last failure in the i^{th} PM cycle is irrelative to the occurrence time of the first failure in the i+1st PM cycle. This RV generating method for the TBF is called "the offset inverse transformation method" in this paper.

Intuitively, however, the occurrence time of the first failure in the i+1st PM cycle is affected by the occurrence time of the last failure in the i^{th} PM cycle since the failure occurrence of the system follows the non-homogenous Poisson process (NHPP) and the PM is imperfect (i.e., the PM will not renew the system to zero failure rate). Therefore, in this research, we have developed a modified inverse transformation method for generating the RVs of the TBF which is called "the trace-back inverse transformation method". The second proposed method assumes the occurrence time of the first failure in the i+1st PM cycle is affected by the occurrence time of the last failure in the i^{th} PM cycle.

Furthermore, since the rejection method is often applied to generating RVs of complicated distributions, we also present the third proposed method, the rejection method, for generating the RVs of the TBF under the age-reduced PM model. In this paper, the algorithms and the simulation results for the above three RV generating methods are presented and compared. An example of finding the near-optimal PM policy is provided by using the rejection method of RV generation.

2 THE BACKGROUND FOR THE THEORITICAL MODEL

2.1 Nomenclature

L the finite life time span for the system or equipment

T the time interval of each periodic PM

N the number of PM performed in the finite life time span (L)

k_i the generated number of failures in the i^{th} PM cycle, $i = 0, 1, \ldots, N$

$x_{i,j}$ the generated time between the j-1st and the j^{th} failures in the i^{th} PM cycle, $i = 0, 1, \ldots, N$; $j = 1, 2, \ldots, k_i$

$t_{i,j}$ the generated occurrence time of the j^{th} failure in the i^{th} PM cycle where $t_{i,j} = t_{i,j-1} + x_{i,j}$

x_{i,k_i+1} the generated time between the last (k_i^{th}) and the k_i+1st failures (not existing) in the i^{th} PM cycle

t_{i,k_i+1} the generated occurrence time of k_i+1st failure (not existing) in the i^{th} PM cycle, i.e., t_{i,k_i+1} exceed the time of the i^{th} PM cycle

γ the reduced age after each PM

$w_{i,j}$ the generated effective occurrence time (age) of the j^{th} failure in the i^{th} PM cycle where $w_{i,j} = t_{i,j} - i\gamma$

$U_{i,j}$ the random number required for the generation of $x_{i,j}$

$\lambda(t)$ Original hazard rate function (before the 1st PM action)

$\lambda_i(t)$ Hazard rate function at time t where t is in the i^{th} PM cycle and $\lambda_0(t)=\lambda(t)$

$F(t)$ the cumulated distribution function (CDF) of the TBF at age t

$R(t)$ the reliability at age t

C_{pm} Cost of each PM

C_{mr} Minimal repair cost of each failure

TC The total maintenance cost function in the finite life time span

2.2 Assumptions

- The system has a finite useful life time L.
- The system is deteriorating and repairable over time where the failure process follows the non-homogenous Poisson Process (NHPP) with increasing failure rate (IFR). Weibull distribution with hazard rate function: $\lambda(t) = \left(\frac{\beta}{\theta}\right)\left(\frac{t}{\theta}\right)^{\beta-1}$ is used to illustrate the examples in this paper, where β is the shape parameter and θ is the scale parameter.
- The periodic PM actions with constant interval (T) are performed over the finite time span L.
- The system's age can be reduced γ units of time to result in a younger age (called the effective age) after each PM. Hence, the hazard rate function at time $t_{i,j}$ in the i^{th} PM cycle can be written as

$$\lambda_i(t_{i,j}) = \lambda(t_{i,j}-i\gamma) = \lambda(w_{i,j}). \quad (1)$$

- Minimal repair is performed when failure occurs between each PM.
- The time required for performing PM, minimal repair, or replacement is negligible.

2.3 The Theoretical Model

Based on the theoretical PM model with age reduction proposed by Cheng *et al.* (2004) and Yeh and Chen (2006), the optimal PM policy is obtained by the following steps. The first step is to find the expected cost rate function for the PM model as shown below.

$$C(T,N) = \frac{(N-1)C_{pm} + C_{pr} + C_{mr}\Lambda(T,N)}{NT}, \quad (2)$$

where $\Lambda(T, N)$ is the expected number of failures occurred in the finite time span and is defined as

$$\Lambda(T,N) = \sum_{i=0}^{N-1}\int_{iT}^{(i+1)T}\lambda_i(t)dt \quad (3)$$

with $\lambda_i(\cdot)$ being defined in Eq. (1). Second step is to obtain the time interval of PM (T) as a function of N by taking the partial derivative of T of the above expected cost rate function and letting it equal to zero, i.e., $\frac{\partial C(T,N)}{\partial T} = 0$

Third, the optimal value T^* and N^* of the theoretical model can be obtained by numerically searching $\min_N C(T,N)$, N = 1, 2, … since the cost rate function is a convex function. The hazard rate function of the PM model with age reduction is illustrated in Figure 1.

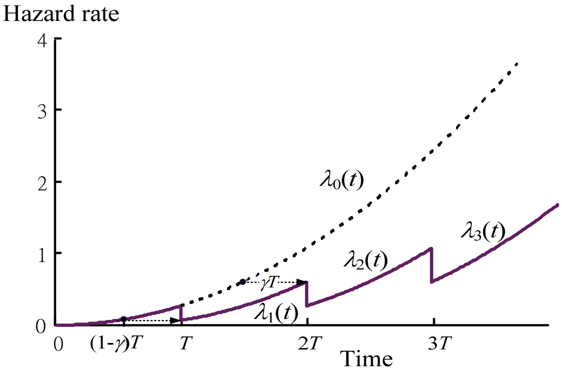


Figure 1: The hazard rate function of the PM Model with age reduction.

3 THE RV GENERATING METHODS OF THE TBF

3.1 The Offset Inverse Transformation Method

This RV generating method assumes that the occurrence time of the last failure in the i^{th} PM cycle is irrelative to the occurrence time of the first failure in the i+1st PM cycle. Thus, when the generated occurrence time of the k_i^{th} failure is within the i^{th} PM cycle but the occurrence time of the k_i+1st failure exceeds the i^{th} PM cycle, i.e., $t_{i,k_i+1} > iT$, we discard the k_i+1st failure and start to generate the occurrence time for the first failure of the i+1st PM cycle.

According to the concept of the inverse transformation method, if x is the time between failures, then, we have

$$U = F(x). \quad (4)$$

However, since the PM model assumes that the minimal repair is performed at each failure occurred between each PM. Therefore, we can re-write Eq.(4) as

$$U_{i,j} = F(x_{i,j}|t_{i,j-1}) = 1 - R(x_{i,j}|t_{i,j-1}). \quad (5)$$

Then, for the age-reduction PM model, we apply Eq.(1) to Eq. (5) and it can be resulted as the following equation.

$$1-U_{i,j} = R(x_{i,j}|t_{i,j\text{-}1}) = \exp\left\{-\int_{t_{i,j-1}}^{t_{i,j-1}+x_{i,j}}\lambda_i(t')dt'\right\} \quad (6)$$
$$\text{for } i = 0,1,...,N;\ j = 1,2,...,k_i$$

which, according to Eq. (1), can be expressed as function of effective age as follows.

$$1-U_{i,j} = R(x_{i,j}|t_{i,j\text{-}1}) = \exp\left\{-\int_{w_{i,j-1}}^{w_{i,j-1}+x_{i,j}}\lambda(t)dt\right\} \quad (7)$$
$$\text{for } i = 0,1,...,N;\ j = 1,2,...,k_i.$$

where $w_{0,j} = t_{0,j}$, $w_{i,0} = t_{i,0} - i\gamma = iT - i\gamma$; $w_{i,j} = t_{i,j} - i\gamma = t_{i,j\text{-}1} + x_{i,j} - i\gamma$. When the TBF of a system is a Weibull random variable, based in Eq. (7), we can generate the TBF random variates by the following equation.

$$x_{i,j} = \left\{-\theta^\beta \ln(1-U_{ij}) + \left[t_{i,j-1} - i\gamma\right]^\beta\right\}^{1/\beta} - t_{i,j-1} + i\gamma \quad (8)$$
$$\text{for } i = 0,1,...,N;\ j = 1,2,...,k_i.$$

The algorithm for the offset inverse transformation method is presented as follows.

(1) Specify the values of the following parameters: β, θ, γ, N, T, L and let i = 0.

(2) Let $t_{i,0} = iT$.
(3) Let $j = 1$.
(4) Generate random number $U_{i,j}$.
(5) Obtain the value of $x_{i,j}$ according to Eq.(8);
let $t_{i,j} = t_{i,j-1}+x_{i,j}$.
(6) If $t_{i,j} < iT$, let $j = j + 1$ and go back to (4)
else go to (7).
(7) If $t_{i,j} < L$, let $i = i + 1$ and go back to (2)
else stop.

It can be seen that the occurrence time of the first failure in the i+1st PM cycle does not relate to the occurrence time of the last failure (t_{i,k_i}), i.e.,

$$t_{i+1,1} = t_{i+1,0} + x_{i+1,1} = (i+1)T + x_{i+1,1}.$$

3.2 The Trace-back Inverse Transformation Method

The proposed second method is modified from the offset inverse transformation method. For the following reasons: (1) the failure occurrence of the system follows the non-homogenous Poisson process (NHPP); (2) the PM is imperfect (i.e., the PM will not renew the system to zero failure rate), this generating method assumes that the occurrence time of the first failure in the i+1st PM cycle is affected by the occurrence time of the last failure in the ith PM cycle. Hence, the theoretical concept for the generation of $x_{i+1,1}$ is shown below.

$$R(x_{i+1,1}\,|\,t_{i,k_i}) = \Pr\{T' > t_{i,k_i} + x_{i+1,1}\,|\,T' > t_{i,k_i}\}$$
$$= \frac{\Pr\{T' > t_{i,k_i} + x_{i+1,1}\}}{\Pr\{T' > t_{i,k_i}\}} = \frac{R(t_{i,k_i} + x_{i+1,1})}{R(t_{i,k_i})},$$

where

$$R(t_{i,k_i} + x_{i+1,1}) = R(t_{i+1,1})$$
$$= \exp\left[-\sum_{l=0}^{i}\int_{lT}^{(l+1)T}\lambda_l(t')dt' - \int_{(i+1)T}^{t_{i,k_i}+x_{i+1,1}}\lambda_{i+1}(t')dt'\right]$$

and

$$R(t_{i,k_i}) = \exp\left[-\sum_{l=0}^{i-1}\int_{lT}^{(l+1)T}\lambda_l(t')dt' - \int_{iT}^{t_{i,k_i}}\lambda_i(t')dt'\right]$$

It turns out that

$$R(x_{i+1,1}\,|\,t_{i,k_i}) = \exp\left\{-\left[\int_{t_{i,k_i}}^{(i+1)T}\lambda_i(t')dt' + \int_{(i+1)T}^{t_{i+1,1}}\lambda_{i+1}(t')dt'\right]\right\}.$$

Then, let $U_{i+1,1} = U_{i,k_i+1} = 1 - R(x_{i+1,1}\,|\,t_{i,k_i})$. For the Weibull case, we can generate the first TBF random variate of the i+1st PM cycle by the following equation.

$$x_{i+1,1} = \begin{Bmatrix}\left[(i+1)(T-\gamma)\right]^{\beta} + \left(t_{i,k_i} - i\gamma\right)^{\beta} \\ -\left[(i+1)T - i\gamma\right]^{\beta} - \theta^{\beta}\ln(1-U_{i+1,1})\end{Bmatrix}^{1/\beta} - t_{i,k_i} + (i+1)\gamma \qquad \text{for } i = 0,1,2,\ldots,N. \tag{9}$$

The algorithm for the trace-back inverse transformation method is provided below.

(1) Specify the values of the following parameters: β, θ, γ, N, T, L.
(2) Let $i = 0$, $t_{0,0}=0$.
(3) Let $j = 1$.
(4) Generate random number $U_{i,j}$.
(5) Obtain the value of $x_{i,j}$ according to Eq.(8);
let $t_{i,j} = t_{i,j-1}+x_{i,j}$.
(6) If $t_{i,j} < iT$, let $j = j + 1$ and go back to (4)
else go to (7).
(7) If $t_{i,j} < L$,
obtain the value of $x_{i+1,1}$ according to Eq.(9);
let $t_{i+1,1} = t_{i,k_i} + x_{i+1,1}$;
let $i = i + 1$ and $j = 2$;
go back to (4)
else stop.

It can be seen that the occurrence time of the first failure in the i+1st PM cycle depends on the occurrence time of the last failure (t_{i,k_i}), i.e.,

$$t_{i+1,1} = t_{i,k_i} + x_{i+1,1}.$$

3.3 The Rejection Method

It can be seen from Eq.(4) or Eq.(5) that the hazard rate function is changed when performing a PM. This makes the formula for generating the TBF random variates shown in Eq.(6) and Eq.(7) very complicated. Therefore, the rejection method is applied in this research.

In the rejection method, two random numbers, say U_1 and U_2, are required for generating each RV. Suppose $\lambda_i(t)$ is the hazard rate function of the ith PM cycle. U_1 is used to generate a RV from a hazard rate function with a simple formula, say $\lambda(t)$ where $\lambda(t) \geq \lambda_i(t)$ for any $t \geq 0$. Then, the RV generated by using U_1 is accepted if $U_2 < \lambda_i(t)/\lambda(t)$.

In this research, we use the original hazard rate function $\lambda(t)$ (i.e., the hazard rate function before the first PM) to generate the RV of the TBF corresponding to U_1. For the Weibull case, we can obtain the TBF formula as the following equation.

$$x_m = \left[-\theta^{\beta}\ln(1-U_1) + (t_{m-1})^{\beta}\right]^{1/\beta} - t_{m-1}. \tag{10}$$

The algorithm of the rejection method is presented as follows.

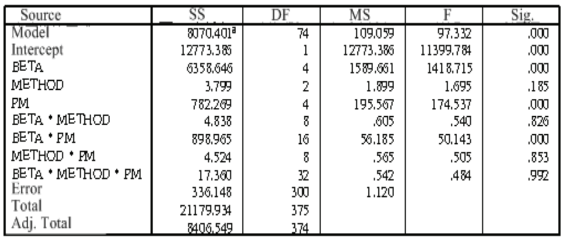
```
(1) Specify the values of the following parameters: $\beta$, $\theta$, $\gamma$, $N$, $T$, $L$.
(2) Let $t_{0,0}$=0, $t_0$=0.
(3) Let $m$ = 0, $i$ = 0, $j$ = 1.
(4) Generate random number $U_1$.
(5) Obtain the value of $x_m$ according to Eq. (10);
    let $t_m = t_{m-1}+x_{i,j}$.
(6) If $t_m < iT$, go to (7)
    else go to (10).
(7) Generate random number $U_2$
(8) Calculate $\lambda(t_m)$ and $\lambda_i(t_m)=\lambda(t_m-i\gamma)$.
(9) If $U_2 \le \lambda_i(t_m)/\lambda(t_m)$, let $t_{i,j} = t_m$; $j = j+1$; $m = m+1$;
    go back to (4)
    else $j = j$; $m = m+1$; go back to (4).
(10) If $t_m < L$, let $i = i + 1$ and $j$ = 1; go back to (7)
    else stop.
```

It can be seen that the rejection method is easy to use since it does not need to derive the formula of $R_i(t)$ for i = 1, 2, …, N.

4 EXAMPLES AND DISCUSSION

In the examples, let the finite life time period (L) be 6 time units and the PM interval (T) be 1 time unit. The values of parameters are set as: θ = 0.4; N = 5; C_{pm} = a+bi = 5+100i for the i^{th} PM; C_{mr} =3.1036. Then, we construct 25 experiments for each RV generating method, which consist of 5 different β values, each with 5 replicates. There are 30 runs for each experiment. We compare the differences between the mean number of failures obtained from Eq. (3) and the sample averages from the three RV generating methods. The analysis of variance (ANOVA) for the number of failures generated is also provided in Table 1. It can be seen that the three RV generating methods do not have significant different. Parameter β and the number of PM performed do significantly affect the number of failures generated, which demonstrates the validity of the simulation models.

Table 1: The ANOVA of the generated number of failures.



Source	SS	DF	MS	F	Sig.
Model	8070.401ª	74	109.059	97.332	.000
Intercept	12773.386	1	12773.386	11399.784	.000
BETA	6358.646	4	1589.661	1418.715	.000
METHOD	3.799	2	1.899	1.695	.185
PM	782.269	4	195.567	174.537	.000
BETA * METHOD	4.838	8	.605	.540	.826
BETA * PM	898.965	16	56.185	50.143	.000
METHOD * PM	4.524	8	.565	.505	.853
BETA * METHOD * PM	17.360	32	.542	.484	.992
Error	336.148	300	1.120		
Total	21179.934	375			
Adj. Total	8406.549	374			

Table 2: Parameters applied in the PM model.

β	θ	L	h	a	b	C_m
3.2	0.4	2	0.19	5	100	3.1036

4.1 The Near-Optimal Solution

Table 2 shows the parameter values used in the proposed simulation models as well as in the theoretical model of Yeh and Chen (2006). By using the rejection method, Table 3 presents the 30-run simulation results for N = 1 to 6. The smallest (best) total maintenance cost of each run is highlighted by shadow background.

It can be seen from Table 3 that, for each N, the average value of TC from the 30-run simulation is very close to the value obtained by using the theoretical method based on Yeh and Chen (2006). Both methods (simulation and theoretical) provides the same optimal policy of N^*=3 and γ^*=0.4781. Again, it has demonstrated that the experiment results obtained by simulation methods are consistent with those obtained by the theoretical model when large sample runs are generated.

It should be noted that the best solution of N, γ, and TC (marked with shadow) resulted from each simulation run are different from those obtained by the theoretical model. It is because the optimal solution of the theoretical model is obtained by taking the expected cost rate over the infinite time interval or over the large number of systems in a finite time interval. However, the simulation method considers the situations of a single system in a finite time interval.

For a single system in a finite time span, according to Table 3, the best solutions of each run (with shadow) can be categorized into three near-optimal policies: (N=2, γ=0.6667), (N=3, γ=0.4781), and (N=4, γ=0.3655). Table 4 lists the simulation runs in each near-optimal policy and presents the average, the smallest, and the largest minimal TC of the near-optimal policy. Among these best solutions, the average of the minimal TC (184.1143) is significantly different from the theoretical minimal TC (189.7280). The results have demonstrated that the theoretical PM model might not be suitable for a single system over a finite time interval.

Hence, in practical, when considering a single system to be preventively maintained in a finite time period, especially for short time period, more than one single near-optimal policy is suggested. In this example, either (N=2, γ=0.6667) or (N=3, γ=0.4781) or (N=4, γ=0.3655) may be chosen as the best (near-optimal) PM policy.

Table 3: The results of the 30 Simulation runs.

Run#	N	1	2	**3**	4	5	6
	γ	1	0.6667	**0.4781**	0.3655	0.2957	0.2483
1		216.730	189.894	**180.155**	194.132	194.575	210.016
2		229.144	196.101	192.570	**187.925**	210.093	194.498
3		250.869	196.101	**189.466**	197.236	213.197	206.912
4		250.869	186.790	**180.155**	200.340	188.368	213.120
5		204.315	199.205	204.984	**187.925**	219.404	197.602
6		263.284	**189.894**	201.880	194.132	197.679	197.602
7		213.626	**165.065**	183.259	197.236	185.264	203.809
8		232.248	214.723	**192.570**	194.132	206.990	194.498
9		241.558	**177.480**	183.259	200.340	191.472	206.912
10		216.730	189.894	**180.155**	194.132	188.368	206.912
11		219.833	211.619	**204.984**	209.650	213.197	216.223
12		222.937	189.894	189.466	**181.718**	197.679	197.602
13		247.766	**177.480**	180.155	209.650	197.679	216.223
14		247.766	196.101	**180.155**	206.547	197.679	203.809
15		198.108	196.101	189.466	**181.718**	200.782	194.498
16		216.730	192.998	**183.259**	200.340	185.264	203.809
17		226.040	205.412	**189.466**	191.029	194.575	206.912
18		195.004	202.308	211.191	**191.029**	210.093	191.394
19		216.730	**183.687**	186.362	191.029	200.782	213.120
20		226.040	199.205	**183.259**	203.443	200.782	206.912
21		232.248	186.790	189.466	**184.822**	194.575	203.809
22		204.315	**196.101**	198.777	197.236	197.679	200.705
23		207.419	186.790	**180.155**	206.547	188.368	206.912
24		210.522	208.516	195.673	**187.925**	206.990	197.602
25		219.833	205.412	**195.673**	203.443	197.679	213.120
26		257.076	192.998	**173.948**	215.858	191.472	219.327
27		216.730	211.619	195.673	**172.407**	210.093	188.291
28		216.730	**189.894**	201.880	200.340	197.679	216.223
29		210.522	**168.169**	180.155	206.547	191.472	216.223
30		247.766	**186.790**	189.466	187.925	197.679	200.705
Avg.		225.316	193.101	**189.569**	195.891	198.92	204.843
Theo.		221.495	191.076	**189.728**	192.850	197.222	202.051

Table 4: The near-optimal Policies of the Simulation.

Policy 1 (N^*=2, γ^*=0.6667)		Policy 2 (N^*=3, γ^*=0.4781)		Policy 3 (N^*=4, γ^*=0.3655)	
Run#	Min. TC	Run#	Min. TC	Run#	Min. TC
6	189.8940	1	180.1552	2	187.9252
7	165.0652	3	189.4660	5	187.9252
9	177.4796	4	180.1552	12	181.7180
13	177.4796	8	192.5696	15	181.7180
19	183.6868	10	180.1552	18	191.0288
22	196.1012	11	204.9840	21	184.8216
28	189.8940	14	180.1552	24	187.9252
29	168.1688	16	183.2588	27	172.4072
30	186.7904	17	189.4660		
		20	183.2588		
		23	180.1552		
		25	195.6732		
		26	173.9480		
Runs	9	Runs	13	Runs	8
Avg.	181.6177	Avg.	185.6462	Avg.	184.4337
Max.	196.1012	Max.	204.9840	Max.	191.0288
Min.	165.0652	Min.	173.9480	Min.	172.4072
Overall average of min. TC: 184.1143					

5 CONCLUSIONS

The proposed three simulation methods are not significant different in generating the time-between-failure RVs .for the PM model with age reduction. The rejection method seems simple and easy to use in practical.

For the infinite time span, the results from the simulation method are very close to those obtained by the theoretical model. However, for a finite time span, more than one near-optimal policy can be obtained by the simulation method. Each of the near-optimal solution can be the best PM policy for any single system having a finite life time period. The simulation results have demonstrated that the theoretical PM model might not always suitable for a single system in a finite time span.

The simulation method can be applied in solving more complicated real world situation, such as the consideration of the random shock in a PM model, which is difficult to be solved by the theoretical model.

ACKNOWLEDGEMENTS

This research has been supported by the National Science Council of Taiwan under the project number NSC96-2221-E-324-010.

REFERENCES

Cheng, C.-Y. and Liaw, C.-F., 2005. Statistical estimation on imperfectly maintained system, *European Safety & Reliability Conference 2005 (ESREL 2005)*, Jun. 27-30, 2005, Tri-City, Poland, pp 351-356.

Cheng, C.-Y. Liaw, C.-F., and Wang, M., 2004. Periodic preventive maintenance models for deteriorating systems with considering failure limit, *4th International Conference on Mathematical Methods in Reliability—Methodology and Practices*, Jun. 21-25, 2004, Santa Fe, New Mexico.

Murthy, D. N. P. and Nguyen, D. G., 1981. Optimal Age-Policy with Imperfect Preventive Maintenance, *IEEE Transactions on Reliability* Vol.R-30, No.1, pp.80-81.

Pongpech, J. and Murthy, D. N. P., 2006. Optimal Periodic Preventive Maintenance Policy for Leased Equipment, *Reliability Engineering & System Safety*, Vol.91, pp.772-777.

Ross, S. M., 1997. *Simulation*, Academic Press, San Diego, pp.62-85.

Yeh, R. H. and Chen, C. K., 2006. Periodical Preventive-Maintenance Contract for a Leased Facility with Weibull Life-Time, *Quality & Quantity*, Vol.40, pp.303-313.

TOWARDS A COMPUTATIONALLY EFFICIENT RELATIVE POSITIONING SYSTEM FOR INDOOR ENVIRONMENTS

An RFID Approach

Md. Suruz Miah and Wail Gueaieb
School of Information Technology and Engineering, University of Ottawa
800 King Edward Avenue, Ottawa, Ontario, Canada
{*suruz.miah,wgueaieb*}@*uOttawa.ca*

Keywords: Relative positioning system, RFID, Received signal strength, Mobile robot navigation.

Abstract: The recent advancements of Radio Frequency IDentification (RFID)-based localization approach has necessitates the development of effective solutions for mobile robot navigation systems in an indoor and/or outdoor environment. Among the most common problems pertaining to the modern RFID-based robot navigation systems are that multiple reference RF stations or excessive number of sensors are utilized for the location sensing with RFID, however, particularly in indoor environments, spatial layout or cost problems limit the applicability of those approaches. The contribution of the current manuscript is to devise a simple computationally efficient relative positioning system for indoor environments through a modified RFID tag architecture. The validity of the proposed RFID-based RPS is demonstrated using the real data collected in a typical indoor environment.

NOMENCLATURE

N	Total number of RFID tags
$\hat{p}$	Estimated robot position
p	True robot position
p_i	Position of tag i
$\hat{e}$	Robot position error
ΔRSS	Received signal strength difference
RSS_i	Average RSS value of tag i
RSS_{ji}	j^{th} RSS value of tag i

1 INTRODUCTION

Due to the advent of RFID and RFID systems (Nasri et al., 2008), and their applications in the field of robotics (Milella et al., 2007), positioning systems have been used to deliver location information in indoor and/or outdoor environments. The primary role of such localization systems is to estimate and report geographical information pertaining to the data processing unit associated with a mobile robot for the purpose of management, enhancement, and personalization services. The current manuscript contributes to the design and implementation of a modular, cost-effective, and an easy-to-implement mobile robot navigation algorithm in cooperation with an open RFID hardware architecture.

Most of the RFID-based navigation systems suggested in the literature are tailored along with the localization systems where the central task of an RFID system is to estimate the position of a mobile robot at a certain time instant. In the current work, an RFID reader is mounted on a mobile robot and some RFID tags are placed at 3-D locations (ceiling, for example) in an indoor environment. At every time instant, the reader broadcasts a time-varying Radio Frequency (RF) signal to all tags in its operating range and tags simply response back to the reader with their Received Signal Strength (RSS) measurements. These RSS values are then used by the mobile robot to approximate its relative position with respect to a desired path that the robot has to follow. Despite the significant contributions of RFID systems and RSS measurements in the literature to date, the localization problem of a mobile robot remains some significant technical challenges that must be overcome. Hence, our effort is devoted to the development of a positioning system for an indoor mobile robot where the previous methods might not work. The main contributions of the current work is to devise a computationally efficient relative positioning system for indoor environments using a modified RFID tag architecture. This approach is different from the existing RSS-based localization methods (Graefenstein and Bouzouraa, 2008) in that it uses the RSS measurements of the RF signal transmitted by the RFID reader. This is simply because the passive tag circuit is energized from the RF signal broadcasted by the reader. Hence, the RSS value in the tag circuitry is more significant than that in the RFID reader.

The rest of the paper is outlined as follows. Some of the most commonly used robot navigation and/or localization systems are given in section 2. Section 3 describes the proposed RFID-based relative positioning architecture followed by fundamentals of RFID system theory and its limitations. The RFID system implementation is discussed in section 4 followed by some real-time experimental results illustrated in section 5. Finally, conclusions with some future research avenue are drawn in section 6.

2 RELATED WORK

Mobile robot navigation and/or localization system has been the subject of several studies. Among the most common and popular navigation algorithms suggested in the state of the art are dead-reckoning-based, landmark-based, vision-based, behavior-based navigation techniques. Each of these navigation techniques has its own advantages and disadvantages, although it is difficult to rate them objectively. However, some aspects can be unequivocally compared, such as the computational complexity, the navigation accuracy, or the amount of information a priori required for the proper operation of the algorithm.

The fundamental idea behind the dead-reckoning navigation techniques is that they provide position, heading, translational, and rotational velocities of an autonomous mobile robot. These techniques are widely used due to their simplicity and easy maintenance (D'Orazio et al., 1993). The shortcomings of this technique is obviously that small precision errors and sensor drifts inevitably lead to increasing cumulative errors in the robot's position and orientation over time, unless an independent reference is used periodically to correct the errors. As an alternative to dead-reckoning-based methods, natural or artificial landmarks have been used at various locations in the environment in order to better estimate the position of the mobile robot (Lin and Lal Tummala, 1997; Yi and Choi, 2004). Nevertheless, a landmark-based navigation strategy relies on identification and subsequent recognition of distinct features or objects in the environment that may be a priori known or extracted dynamically. Due to sensors noise and possible dynamic changes of the operating environment, the recognition process of features or objects might become quite challenging. Given the shortcomings of the landmark-based techniques, some researchers shifted their interest to vision-based navigation systems. Vision sensors can have wide field-of-view, can have millisecond sampling rates, and can be easily used for trajectory planning (Desouza and Kak, 2002). Yet, some disadvantages of vision include lack of depth information, image occlusion, low resolution and the requirement for extensive data interpretation (recognition). As the development of different autonomous robot navigation techniques in real-world environments constitutes one of the major trends in current research on robotics, one important problem is to cope with the large amount of uncertainty inherited from natural environments. As such, soft computing techniques have received a considerable attention in recent years. Numerous navigation techniques have been suggested in the state of the art using some tools of computational intelligence such as fuzzy logic, neural network, neuro-fuzzy system, genetic algorithm, or several combinations of them.

With these concerns in mind, several works have considered localizing a mobile robot based on the application of emerging RFID technology owing to its wide availability, non-touch recognition system that transmits and processes the information on events and environments using a wireless frequency and small chips. Since an RFID system can recognize at high-speed and send data within various distances, the application of RFID technology has been increased and RFID systems have been applied for the robot technology recently (Kulyukin et al., 2004).

Hahnel et al. studied to improve the localization with a pair of RFID antennas (Hahnel et al., 2004). They presented a probabilistic measurement model for RFID readers that allow them to accurately localize the RFID tags in the environment.

In addition, robot's position estimation techniques can be classified as range-based and bearing-based. The main idea behind range-based techniques is to trilaterate the robot's position using some known reference points and the estimated distances at those points in the environment. Distances can be estimated from either RSS measurements or time-based methods. Although a small subset of such works have explored the use of Time of Flight (ToF) (Lanzisera et al., 2006) or Time Difference of Arrival (TDoA) measurements, RSS is generally the feature of choice for indoor positioning. This is due to the fact that RSS measurements can be obtained relatively effortlessly and inexpensively. In addition, no extra hardware (e.g., ultrasonic or infra-red) is needed for network-centric localization (Youssef, 2004). On the other hand, bearing-based schemes use the direction of arrival (DoA) of a target. However, these schemes require multiple range sensors in order to be better suited for mobile robot applications (Kim and Chong, 2009).

3 PROPOSED RFID-BASED RELATIVE POSITIONING (RPS)

The fundamental problem in most real-world localization systems is to produce position estimate from past observations on a discrete grid of points in an environment. Despite the significant limitations of RSS measurements stated in the literature, the proposed approach seeks a function modeled by

$$\hat{p} = f(p_1, \ldots, p_N),$$

where N is the total number of RFID tags placed in a 3D workspace, $p_i = (x_i, y_i, z_i)$ with $1 \leq i \leq N$ represents the coordinates of an RFID tag in the world coordinate system, and f is a function of RSS measurements associated with the RFID tags. $\hat{p} = (x_r, y_r, z_r)$ is the estimated relative position of the robot with respect to the desired path on the ground. In the current work, the position estimation is restricted to the 2D space due its simplicity, as such, z_r, which is the height information, is simply ignored. To quantify the navigation accuracy, the error model is defined by

$$\hat{e} = \|\hat{p} - p\|,$$

where p is the true position of the mobile robot.

In order to compute the estimated relative position of the mobile robot using an RFID system, RFID tags are arranged in a fixed pattern on the ceiling, for instance, as depicted in Figure 1. An RFID reader is mounted on the mobile robot and four tags are attached to the ceiling. The points P1, P2, P3, and P4 define the orthogonal projections of the four tags on the ground. The robot is supposed to navigate along the virtual desired path defined by the projection points.

3.1 Technical Background of an RFID System

We now review the fundamental properties of a commercially available RFID system in the market. RFID is a type of automatic object identification system. The principle of an RFID system consists of storing an individual static binary code to every object that need to be identified and the automatic seizing of information via radio waves. An RFID system is mainly composed of three main components: a tag, an RFID reader, and a host computer (Peris-Lopez et al., 2006).

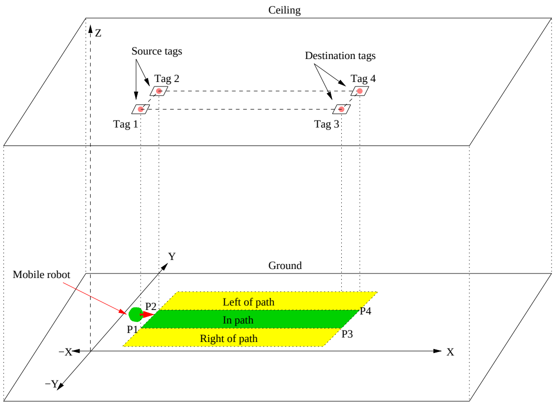


Figure 1: Relative position system setup.

The tag is composed of a microchip with some basic storage capabilities, and a coupling element such as antenna coil for communication. An RFID reader is generally composed of an RF module, a control unit, and a coupling element to interrogate electronic tags via RF communications. The purpose of the host computer is to execute a special purpose software in order to store, process, and analyze the data acquired by the reader. In the current work, an RFID reader is interfaced with the robot's central processing unit to perform further processing of tags' information.

3.2 Relative Positioning Technique

As mentioned above, most of the existing RFID systems available in the market provide only static information which limit its applicability in many real-world proximity-based RFID applications. In the current work, we propose a navigation strategy for guiding a mobile robot in an indoor environment using a customized RFID tag architecture that allows to encode some dynamic information along with its existing static ID. Figure 2 depicts a customized model of an RFID tag employed in the current research. The tag receives an RF signal transmitted by the reader which is then rectified to get its RSS value. In the present RFID system, the tag has some processing capability to convert the RSS value into an 8-bit binary code. As can be seen in figure 2, the RSS measurement of the RFID reader query is embedded with the tag's existing static binary ID (16-bit in this case) which is then backscattered to the RFID reader. It is important to articulate the fact that the reader architecture of the proposed RFID system requires no customization as it would read the 24-bit (16-bit tag-ID + 8-bit RSS) frame in exactly the same way it normally reads tag-IDs. The RFID reader extracts the frame backscattered by the tag which is then passed to the

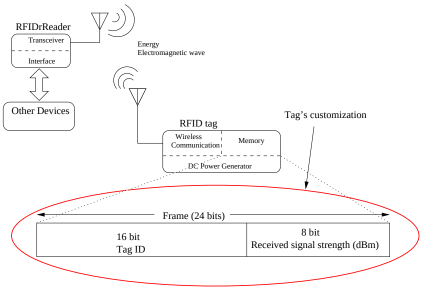


Figure 2: Proposed RFID architecture.

processing element on the robot's board to decode it into a tag-ID and an RSS value. The RSS values are used to approximate the relative position of the mobile robot with respect to its desired path.

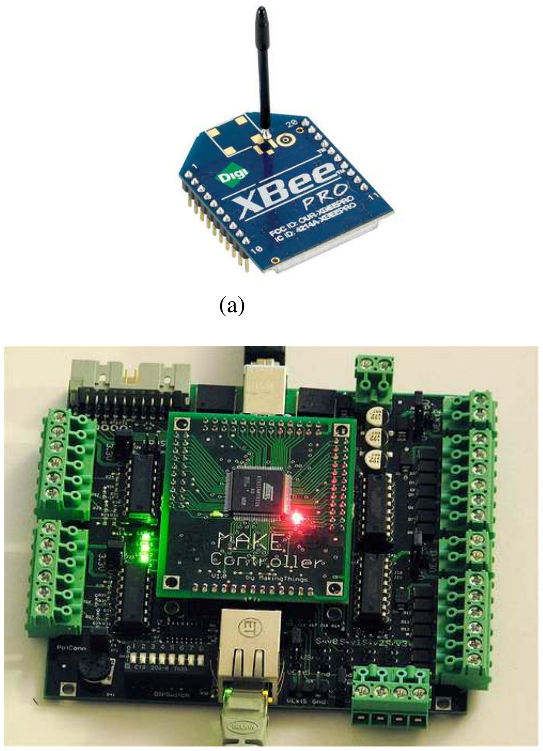
We now explain how the relative position of the robot can be approximated by incorporating tags' RSS values in an indoor environment. In this work, the robot is presented with a set of four tag-IDs, $S = \{1,2,3,4\}$, for instance, where tags with IDs 1 and 2 define the source (starting) point, and the tags with IDs 3 and 4 define the destination, respectively. Note that the tag coordinates in the world coordinate system are not necessarily known. The robot computes its position with respect to the desired path by extracting and decoding the frames backscattered by fours tags defining the path. The RSS values are then used to model the relative position which is defined by

$$\Delta RSS = (RSS_1 + RSS_3) - (RSS_2 + RSS_4) \ , \qquad (1)$$

where the RSS_i with $1 \leq i \leq 4$ represents the average RSS value associated with tag i. The RSS samples received from each tag are passed to an M-point moving average filter for better estimation. The filter is modeled as

$$RSS_i = \frac{1}{M} \sum_{j=1}^{M} RSS_{ji} \quad \forall i \in S \ , \qquad (2)$$

where RSS_{ji} is the j^{th} RSS value at tag i. The significance of ΔRSS is the amount of robot's divergence from its desired path. Ideally, ΔRSS is closest to nil when the robot is on the right track. It diverges from zero as the robot moves drifts away from its path. The sign of ΔRSS then depends on the side of the path the robot is located.



(a)

(b)

Figure 3: RFID system (a) RF module used to emulate an RFID tag[4] and (b) Make controller board used to emulate an RFID reader[2].

4 RFID SYSTEM IMPLEMENTATION

The proposed RFID system is emulated using the XBee Pro Modules[1] shown in Figure 3(a) as an integrated RF solution. The modules include MC13193 RF chip by freescale, which is compliant to the IEEE 802.15.4 norm (Graefenstein and Bouzouraa, 2008). One of the XBee Pro modules is attached to the Make Controller (MC) board[2] (figure 3(b)) to emulate a commercial RFID reader. The MC board secures the communication between the emulated RFID reader and the robot.

In order to obtain an RSS value from a tag i, $1 \leq i \leq 4$, the reader broadcasts a message with its own static address. The tags are simply configured to reply to the reader's query with their individual binary frames. As mentioned above, each tag's frame consists of its 16-bit static address and 8-bit RSS value.

[1] http://www.digi.com/products/wireless/point-multipoint/xbee-pro-series1-module.jsp, http://www.digi.com

[2] http://www.makingthings.com/, http://www.makingthings.com

The reader simply extracts and decodes the frames in order to get the tag's ID and the corresponding RSS value and then passes them along to the mobile robot for further processing.

5 EXPERIMENTAL RESULTS

The purpose of this section is to provide details on the experimental evaluation of the proposed relative positioning system using the emulated customized RFID tag architecture. The performance is evaluated using real data in a research laboratory that reflects a typical indoor operating environment.

The experiments of the proposed RFID-based RPS were carried out at discrete points distributed over approximately $3 \times 6 \times 2$ m test area of a research laboratory with four tags attached on four different posts ($\approx$ 2 m high). The test area is divided into uniform square grids of 30×30 cm^2. The layout of the test environment is depicted in Figure 4. The orthogonal projection points of the four tags on the ground are what we call herein S1 and S2 for source, and D1 and D2 for destination. The desired trajectory is the line linking the midpoints of the lines connecting S1 and S2, and D1 and D2, respectively.

To test the proposed RPS concept, ΔRSS is computed at 30 different locations: 10 on the desired robot trajectory (shaded area in figure 4), 10 on its left, and 10 on its right. The results are revealed in Table 1. Each RSS value (in dB) is the output of an 8-point moving average filter as defined in (2). It can be seen from Table 1 that the data corresponding to the left of the path is globally larger than zero, which confirms that the robot is indeed out of its desired trajectory. However, the same conclusion cannot be generalized on the data collected on the right of the path since it is generally close to that collected on the path. This may be due to several reasons. The side test locations are only 0.6 m off the path, which is an insignificant distance compared to the height of the emulated RFID tags (2 m). In other words, the distance between S1 and robot location 5, for instance, is not significantly different from that between S1 and location 5R. This is a main source of ambiguity which contributes to this lack of precision. We believe that ΔRSS would be more distinguishable across the three regions if the side locations were at least 3 m off the path. This threshold highly depends on the RF signal attenuation with the distance traveled. It is also important to investigate better noise filtering techniques to filter the severe noise experienced at the testing scene. The lab at which the experiments were conducted contain an abundance of metal cabinets and obstacles of various materials. Such a choice was made on purpose to test worst condition scenarios.

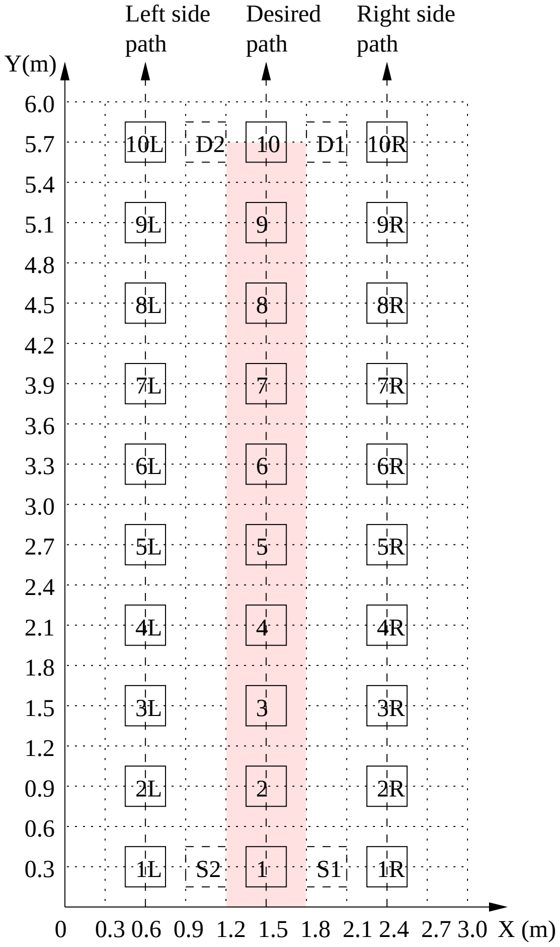


Figure 4: Experimental setup of the proposed navigation system.

6 CONCLUSIONS

The rising prominence of location estimation in many real-world applications necessitates the development of an appropriate positioning system in an indoor environment. Due to the ubiquity of such localization systems, the proposed RFID-based localization system provides a suitable and a cost-effective solution for devising such systems. In this paper, we have examined the problem of relative positioning system using RSS measurements of a modified RFID tag architecture and have proposed a novel guidance principle for a mobile robot to navigate in an indoor environment based on the strength of the RF signal transmitted by the RFID reader. As the first contribution of the

Table 1: Performance of the relative positioning system.

Position	S1	S2	D1	D2	ΔRSS
1L	41	41	58	53	5
2L	44	44	58	54	4
3L	47	46	59	54	6
4L	50	45	55	54	6
5L	50	48	50	50	2
6L	51	52	49	46	2
7L	53	57	53	40	9
8L	53	58	50	41	4
9L	54	55	48	42	5
10L	63	60	43	38	8
Average					**5.1**
Std. Dev.					**2.3**
1	28	36	53	56	-11
2	42	38	54	57	1
3	44	45	50	50	-1
4	47	46	54	56	-1
5	50	46	52	54	2
6	51	52	49	49	-1
7	52	54	49	42	5
8	55	53	41	49	-6
9	57	56	38	44	-5
10	57	61	39	36	-1
Average					**-1.8**
Std. Dev.					**4.5**
1R	37	42	62	57	0
2R	45	43	58	54	6
3R	43	47	56	51	1
4R	55	47	58	53	13
5R	54	51	49	49	3
6R	54	54	52	50	2
7R	56	57	49	48	0
8R	55	54	44	47	-2
9R	53	53	39	45	-6
10R	57	59	37	42	-7
Average					**1.0**
Std. Dev.					**5.8**

current work, spatial relative positioning is proposed to address the variability of tags' RSS patterns over the workspace. The proposed method was evaluated using real data from a typical office environment. Although the preliminary results reported in the present manuscript reveal what might be a promising indoor RPS method, more effort needs to be done to bring the proposed technique to a more mature stage.

REFERENCES

Desouza, G. N. and Kak, A. C. (2002). Vision for mobile robot navigation: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):237–67.

D'Orazio, T., Ianigro, M., Stella, E., Lovergine, F. P., and Distante, A. (1993). Mobile robot navigation by multi-sensory integration. In *Proceedings of 1993 IEEE International Conference on Robotics and Automation ICRA*, pages 373–9, Atlanta, GA, USA.

Graefenstein, J. and Bouzouraa, M. E. (2008). Robust method for outdoor localization of a mobile robot using received signal strength in low power wireless networks. In *2008 IEEE International Conference on Robotics and Automation. The Half-Day Workshop on: Towards Autonomous Agriculture of Tomorrow*, pages 33–8, Pasadena, CA, USA.

Hahnel, D., Burgard, W., Fox, D., Fishkin, K., and Philipose, M. (2004). Mapping and localization with RFID technology. In *Proceedings - IEEE ICRA*, number 1, pages 1015–1020, New Orleans, LA, United States.

Kim, M. and Chong, N. Y. (2009). Direction sensing RFID reader for mobile robot navigation. *IEEE Transactions on Automation Science and Engineering*, 6(1):44–54.

Kulyukin, V., Gharpure, C., Nicholson, J., and Pavithran, S. (2004). RFID in robot-assisted indoor navigation for the visually impaired. In *2004 IEEE/RSJ IROS*, pages 1979–84, Sendai, Japan.

Lanzisera, S., Lin, D. T., and Pister, K. S. J. (2006). RF time of flight ranging for wireless sensor network localization. In *Proceedings of the Fourth Workshop on Intelligent Solutions in Embedded Systems, WISES 2006*, pages 165–176, Vienna, Austria.

Lin, C.-C. and Lal Tummala, R. (1997). Mobile robot navigation using artificial landmarks. *Journal of Robotic Systems*, 14(2):93–106.

Milella, A., Vanadia, P., Cicirelli, G., and Distante, A. (2007). RFID-based environment mapping for autonomous mobile robot applications. In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM*, Zurich, Switzerland.

Nasri, N., Kachouri, N., Samet, M., and Andrieux, L. (2008). Radio Frequency IDentification (RFID): Working, design considerations and modelling of antenna. In *2008 5th International Multi-Conference on Systems, Signals and Devices, SSD'08*, Amman, Jordan.

Peris-Lopez, P., Hernandez-Castro, J. C., Estevez-Tapiador, J. M., and Ribagorda, A. (2006). RFID systems: a survey on security threats and proposed solutions. In *Personal Wireless Communications. IFIP TC6 11th International Conference, PWC 2006. Proceedings*, pages 159–70, Albacete, Spain.

Yi, S.-Y. and Choi, B.-W. (2004). Autonomous navigation of indoor mobile robots using a global ultrasonic system. *Robotica*, 22:369–74.

Youssef, M. (2004). *The Horus WLAN location determination system*. PhD thesis, University of Maryland, Maryland.

THE PERFORMANCE OF OPC-UA SECURITY MODEL AT FIELD DEVICE LEVEL

Olli Post, Jari Seppälä
Department of Automation Science, Tampere University of Technology, Tampere, Finland
olli.post@tut.fi, jari.seppala@tut.fi

Hannu Koivisto
Department of Automation Science, Tampere University of Tecnology, Tampere, Finland
hannu.koivisto@tut.fi

Keywords: OPC-UA, IPSec, Field device security.

Abstract: This paper discusses the performance of OPC UA security model at field device level. Process networks have traditionally been isolated networks but today there is interest to integrate process networks to manufacture and office network. Remote management of field devices via Internet is also gaining interest. This requires implementation of TCP/IP in field devices. However, this causes process networks not being isolated anymore and attention must be paid to the security of process networks. OPC UA is a specification for data transfer in automation systems that can be used to integrate information, horizontally and vertically. Security has also been considered in OPC UA but security measures implemented by OPC UA are too heavy to be uses in field devices. Thus, implementing security profile for authentication without encryption in OPC UA or running OPC UA on IPSec without its own security profile is proposed.

1 INTRODUCTION

Today, there is a growing interest towards the integration of TCP/IP to the process networks. In process networks, security has traditionally been based on access control. Traditional fieldbus based process networks have been isolated networks. Security has been based on restricting physical access. Therefore it has been assumed that there are neither passive nor active attacks in process network. Security measures in isolated process networks have been targeted against user errors. However, this isolation is not the case anymore as TCP/IP is merged to field devices. This allows field devices to be managed over Internet using web applications but this also provides a path for an attacker from Internet to process network using attacks that are well trained in the Internet. It seems that the focus on attacks on automation systems is shifting from internal attacks towards external attacks (Treytl et al., 2005). Still, backdoor accesses such as desktop modems, wireless networks, laptop computers and trusted vendor connections are remarkable sources of attacks (Byres & Hoffman, 2003). The shift can be inflicted because of the path to automation systems that TCP/IP produces. It appears that the external attacks aren't targeted specifically to automation systems but they inflict them as well (Treytl et al., 2005). There is also much interest towards wireless techniques in process network. Controlling access to a wireless media is very hard, if not impossible. Considering these changes in process networks it is clear that the assumption of a secure media in process network is no longer valid. Therefore security against intentional misuse must be considered.

2 SECURITY AT THE FIELD DEVICE LEVEL

2.1 Concepts of Security

Security can be divided into sub concepts and examine security using these sub concepts. These sub concepts are confidentiality, integrity and availability. Confidentiality guarantees the data from unauthorized disclosure, integrity guarantees that data is transferred unaltered in the information channel. Availability is reachability of data for authorized users.

The confidentiality of the data can be assured by encrypting data, the integrity of data can be assured using hash codes to authenticate data. Availability is more complex concept and any single technique can't assure it. It's also usually reverse requirement compared to confidentiality and integrity. Securing availability requires common practices and techniques to ensure that all the field devices and fieldbus in process network are fully functional continuously.

2.2 Device Level Limitations

The International Society of Automation (ISA, 2004) has defined distinctions between process networks and office networks that create differing requirements for security. Specific features for process network, that are important in the sense, of this paper is that field devices have little resources, unwanted incidents can cause serious damage to property, injuries and even death to people, events in the network are time critical, data and services must be available, integrity of data is very important and that data in process network isn't confidential.

It is a well-known fact that encryption causes much more delay in communication than authentication. For example in IPSec, encryption takes multiple times more time than authentication (Elkeelany et al., 2002). Because the events in the process automation are time critical and data has importance for only short period of time confidentiality isn't a requirement for process network. Availability on the other hand is an important requirement for process network, because missing control or measurement data can inflict serious damage to property and people. Another requirement for process network is integrity of data, because modification of messages and unauthorized messages can also inflict serious damage.

3 MINIMUM REQUIREMENTS FOR FIELD DEVICE LEVEL SECURITY

3.1 Security Requirements for Process Network

The requirements for security in process control are *availability* of data and data *integrity.* However availability in process network can't be guaranteed with just a single technique. It can be assured with security policies and different techniques. Therefore it's out of scope of this paper. As processing time is scarce resource in field devices, requirements for process network, in the sense of this paper, can be compressed to following sentence. In process network, data integrity has to be assured as little process time as possible without endangering the keys used in authentication.

3.2 Processing Time Consumed by Authentication

Processing time consumed by authentication is dependent on the used algorithm and length of the key. Therefore short keys would be better in field device level than longer keys. European Network of Excellence in Cryptology divides algorithms to secure or not secure (ECRPYT) (2008). Key lengths on the other hand, can only be secure enough, because every key is possible to break using brute force. However, it should be noted that if a key is adequate today it doesn't mean that it's still adequate in future. Automation systems can be used even for decades and same cryptographic keys are probably used in process networks from start-up to shutdown. Therefore, it shouldn't be possible to break algorithms and keys during periods between yearly maintenance for decades to come. It isn't possible to concretize this because future is hard to predict but it's not advisable to use algorithms and key lengths defined as not secure.

4 OPC UA

4.1 Introduction to OPC UA

OPC means open connectivity via open standards in industrial automation and the enterprise systems that support industry. OPC UA is the specification that is supposed to integrate data exchange in automation, horizontally and vertically. There are nine other OPC standards that are in use. These specifications are used in different purposes and they all have their own niche. OPC UA on the other hand is suppose to operate in all those different niches and ultimately replace all the other OPC specifications completely. The motivation to start the standardization of this unification was compatibility issues in integration of different specifications. OPC UA responds this by offering a unified interface to be used in all the networks in automation. OPC UA specification is already released and reference models are soon to be ready.

4.2 OPC UA Security Model

The security model of OPC UA is specified in part 2 of the specification by OPC Foundation (2009). This document describes how security can be assured using OPC UA. First, secure channel is established to guarantee confidentially, integrity and application authentication. Second, secure session is established between server and client to guarantee user authentication and authorization. It should be noted that confidentiality is not a requirement at field device level and it consumes more calculation power than integrity.

Security of data transfer in OPC UA is specified in part 4 of the specification by OPC Foundation (2009). Secure data transfer between clients and servers in OPC UA is based on certificates issued by certificate authority (CA). OPC UA client and server both have application instance certificates, which are sent to the other member of communication channel while establishing secure channel. Both parties validate received certificates from CA. After secure channel have been established client starts to establish a session with server by sending its software certificate to server. While application instance certificates identify instances, software certificates identify particular users. Server responds to this request by sending its own certificates and once again both members validate received certificates from CA. Certificates validated in OPC UA are X.509 certificates. In field device level verifying every received certificate from CA would cause significant delay to data transfer. Therefore, due to X.509 hierarchical nature it would be feasible for automation system provider to act as CA. For example PLC could act as CA for all the field devices connected to it.

The security profiles of OPC UA are specified in part 7 of the specification by OPC Foundation (2009). There are three security profiles available in OPC UA: Basic128Rsa15, Basic256 and none. Basic128Rsa15 is a suite of security algorithms that include aes128 for encryption, sha1 for authentication and rsa15 for key wrap. Similarly basic256 includes aes256 for encryption, sha1 for authentication and RsaOaep for key wrap. Security policy none doesn't include any security algorithms. There are also asymmetric equivalents for symmetric algorithms but they are probably too calculation expensive to be used in field device level to guarantee security.

OPC UA Stack is specified in part 6 of the specification by OPC Foundation (2009). OPC UA is located at the application layer in OSI model. In figure 1 is depicted OPC UA stack compared to OSI model. From figure 1 can be seen that OPC UA stack and OSI model overlap. For example transport layer is done again in OPC UA stack. UA Transport Layer establishes session between two entities as does transport layer in OSI model.

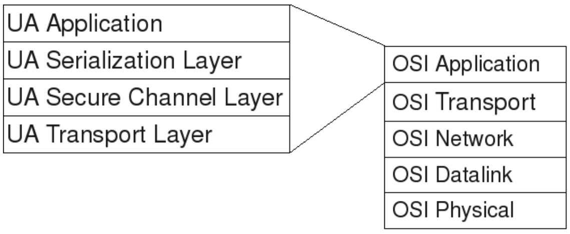


Figure 1: OPC UA stack in OSI model.

5 IPSEC

IPSec is a network layer protocol that can assure data confidentiality and integrity, origin identification and prevent replay attacks (Douligeris et al., 2007). IPSec consists of three elements. First element is security mechanisms. In IPSec there are two of them: authentication header (AH) for authentication and encapsulating security payload (ESP) for encryption. Security mechanisms can also be united to guarantee both encryption and authentication. Second element is security association. This is an agreement on which security mechanisms are used between two members in data transfer. Third element is the infrastructure for key management. It is used to agree an SA between two members.

There are also two modes for transferring data: transport and tunnel. In transport mode ESP mechanism encrypts and optionally authenticates IP payload. AH on the other hand, authenticates payload and also selected portions of IP header. In tunnel mode IP packet is encapsulated inside another IP packet. This way inner IP packet is examined only by the end-points of the data transfer. Thus, data integrity and confidentiality of the whole inner IP packet can be guaranteed.

Another security solution providing data integrity for TCP/IP based field device could be TLS (Dierks & Allen, 1999). It offers the same security as IPSec and it is implemented in common web browsers, which makes it a good choice for remotely configure field devices (Treytl et al., 2004). However, in process network control and measurement data are one-way traffic. There is no need to acknowledge received packets. Data has value for only a short period of time. If a packet is lost on transfer is doesn't matter because another packet is sent shortly after previous. Therefore, there is no need to establish connection between field devices and connectionless UDP would be better solution than TCP. TLS can't be used over UDP but UDP can be packed to IPSec (Alshamsi & Saito, 2005). Thus, IPSec was chosen to under inspection in this paper.

6 PERFORMANCE ANALYSIS

This paper tries to determine whether the security model of OPC UA is efficient in data transfer or could there be another solution for secure data transfer in process network, still allowing OPC UA services to be used.

All three OPC UA security profiles, none, basic128 and basic256, were measured as well as IPSec AH. Also basic128 and basic256 data authentication without encryption were measured. However it should be noted that authentication without encryption is not an OPC UA security profile. By doing also these measurements IPSec AH and authentication done by OPC UA security profiles can be compared. Three packet sizes were used in calculations 1024 bytes, 10 240 bytes and 102 400 bytes to measure delay caused by security measures.

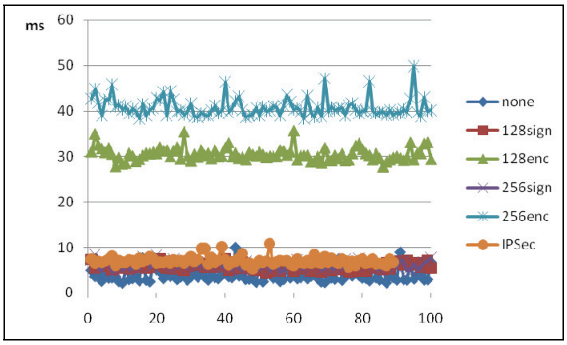


Figure 2: Delay inflicted using 102 400 bytes packet size.

In figure 2 is presented measurements done using packet size 102 400. Although this packet size isn't realistic in field device level it depicts the overall situation well. From figure 2 can be seen that encryption causes much more delay compared to authentication and security profile none. It can be also seen that measurements for all authentication algorithms and security policy none were similar. Therefore it can be said that because confidentiality isn't a requirement in field device level and because encryption adds a lot of overhead to measurement, encryption is not feasible solution to guarantee field device level security.

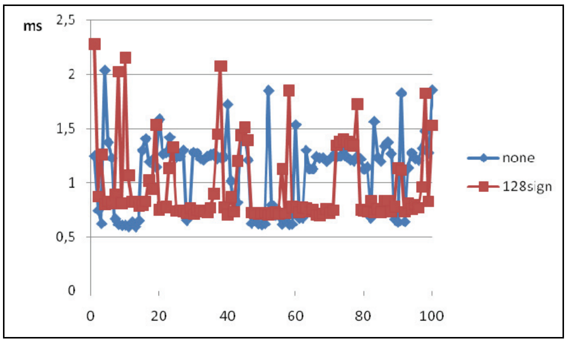


Figure 3: Delay inflicted using packet size 1024 bytes.

In figures 3 and 4 are presented delay inflicted using 1024 bytes packet size. In figure 3 are presented security profile basic128 and none. In figure 4 are presented security profile basic256 and IPSec AH. From figure 3 and 4 can be seen that delay caused by all of these security profiles is alike. It can't be said whether one is better than the other. More measurements are needed for to draw conclusions. However measurements clearly show that in small packet sizes authentication doesn't cause significant delay compared to security profile none.

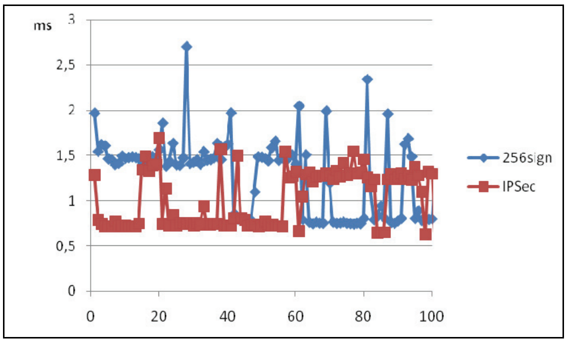


Figure 4: Delay inflicted using 1024 bytes packet size.

7 CONCLUSIONS AND FUTURE WORK

Preliminary results show support the hypothesis that OPC UA security models are not efficient enough to be used in field device level data transfer because there isn't plain authentication supported. Therefore it's suggested that either new security profile for authentication is included or that IPSec is used along OPC UA to guarantee integrity at automation field device level. This way field devices can be remotely managed over TCP/IP and still assure integrity of data efficiently at field device level. Data transfer in OPC UA rests on x.509 certificates. In future it should be considered whether there would be better solution for field device level. For example by PLC acting as CA.

REFERENCES

Alshamsi, A., Saito, T., 2005. A technical comparison of IPSec and SSL. In: IEEE (Institute of Electrical and Electronics Engineers), The 19th International Conference on Advanced Information Networking and Applications. Tamkang, Taiwan 28-30 March 2005.

Byres, E. & Hoffman D., 2003. The Myths and Facts behind Cyber Security Risks for Industrial Control Systems. In: ISA (International Society of Automation), Process Control Conference 2003.

Dierks, T. & Allen, C, 1999. The TLS Protocol Version 1.0, Request for Comments: 2246.

Douligeris, C. et al., 2007. *Network Security Current Status and Future Directions*. Hoboken, NJ: Wiley-IEEE Press.

Elkeelany, O,; Matalgah, M.M., Sheikh, K.P., Thaker, M., Chaudhry, G., Medhi, D. & Qaddour, J., 2002. Performance Analysis of IPSec Protocol: Encryption and Authentication. In: IEEE (Institute of Electrical and Electronics Engineers), International Conference on Communications 2002.New York, United States of America 28 April - 2 May 2002.

European Network of Excellence in Cryptology, 2008. Yearly Report on Algorithms and Keysizes (2007-2008) [Online] Available at: http://www.ecrypt.eu.org/ecrypt1/documents/D.SPA.28-1.1.pdf [Accessed 25 March 2009].

International Society of Automation, 2004. ISA-TR99.00.02-2004 Integrating Electronic Security into the Manufacturing and Control Systems Environment

OPC Foundation, 2009. OPC Unified Architecture

Specification, Part: 2 Security Model, Release 1.01

OPC Foundation, 2009. OPC Unified Architecture Specification, Part: 4 Services, Release 1.01

OPC Foundation, 2009. OPC Unified Architecture Specification, Part: 6 Mappings, Release 1.00.

OPC Foundation, 2009. OPC Unified Architecture Specification, Part: 7 Profiles, Release 1.00.

Treytl, A., Sauter, T. & Schwaiger, C., 2004. Security measures for industrial fieldbus systems - state of the art and solutions for IP-based approaches. In: IEEE (Institute of Electrical and Electronics Engineers), IEEE International Workshop on Factory Communication Systems. Vienna, Austria 22-24 September 2004.

Treytl, A., Sauter, T. & Schwaiger, C., 2005. Security measures in automation systems-a practice-oriented approach. In: IEEE (Institute of Electrical and Electronics Engineers), 10th IEEE Conference on Emerging Technologies and Factory Automation. Catania, Italy 19-22 September 2005.

A SYSTEM-ARCHITECTURE FOR ROBOTIC MOVEMENTS OF GOODS

Approaches Towards a Cognitive Material Flow System

Dennis Ommen, Carsten Beth, Jens Kamenik
OFFIS Institute for Information Technology, Escherweg 2, Oldenburg, Germany
{dennis.ommen, carsten.beth, jens.kamenik}@offis.de

Axel Hahn
Department of Computing Science, Carl von Ossietzky University, Ammerländer Heerstrasse 114-118, Oldenburg, Germany
hahn@wi-ol.de

Keywords: Network Robotics, Mobile Robots and Autonomous Systems, Robot Design, Development and Control, Intelligent transportation technologies and systems.

Abstract: Flexibility, throughput, maintainability, scalability, reliability and low cost: That are the main optimization criteria of material flow systems (MFS). The most of this criteria are diametrical and so hardly to improve considerably with today's existing transportation devices and their static control structures. Hence a new approach of a transportation systems with cooperating robotic units and a novel cognitive environment will be presented. This approach will combine different research areas like robotics and wireless sensor networks to achieve a higher degree of flexibility.

1 INTRODUCTION

This paper will discuss aspects that will lead to a novel, cognitive material flow system (MFS). These aspects are: modularity, energy consumption, dynamic sensor integration, and computational architecture.

A few years ago carrying speed and throughput were the main performance metrics in MFSs. In the future properties like flexibility, modularity, reconfigurability and redundancy will play a decisive role. Reasons are the growing number of different products and product variants and thereon shorter product life-cycle and the growing complexity of products and processes. At the same time the product quality shall by high, the price low and the delivery time short. The movements of goods in a transfer station thus have to be organized in a flexible manner to fulfill these partly conflicting requirements with minimal stock of inventory.

Nowadays it isn't possible anymore to build a MFS for a transfer station which will last for 10 years or longer without being rebuilt substantially. In the future a MFS has to be reconfigurable by design, it must be possible to change the layout with a small amount of time and cost to be able to react on changing requirements (Windth, 2006). The static control structure of former systems has proved to be too inflexible. Therefore a new dynamic control approach an a new robotic system is needed to overcome these issues.

To build such new systems it has to be investigated (1) how to modularize the transportation system and how to identify the modules of single transportation unit (horizontal/vertical actuators, energy supply, controller, communication, etc.), (2) how these modules can be enabled to automatically acting as single transportation units, (3) how the different transportation units can cooperate with each other by using agent-based technologies to achieve a common purpose, (4) how to integrate all the necessary sensor information into a cognitive environment.

Paper Organization. The remainder of the paper is organized as follows: Section 2 introduces the new approach of a cognitive MFS. Section 3 discusses the related work and will show a research trend. Section 4 presents the cognitive MFS approach. The sections 5 and 6 highlights the systems requirements and also approaches towards a cognitive MFS. Section 7 presents concluding remarks.

2 APPROACH

A cognitive MFS is characterized by (1) strict modularization where all modules are able to take individual, autonomous decisions of there acting, (2) cooperation of the modules in order to form a larger entity and/or to perform tasks collectively, (3) the incorporation of the environment in form of intelligent sensors, (4) dynamic reconfigurability of the system through adding, removing or rearrangement of modules, (5) goods accompanied by intelligent SW components cooperating with the before-mentioned entities in order to reach their destination.

The approach of decentralized cooperating autonomous logistic units, where goods and the transportation system autonomously make decisions, can be a way to realize the requirements drafted above (Scholz-Reiter et al., 2007b), (Scholz-Reiter et al., 2007a). Regarding where the decision is taken, the approaches can be separated into two clusters (1) good driven (Scholz-Reiter et al., 2006), (Scholz-Reiter et al., 2007a): An embedded device attached to the package escorts the goods to its destination. During the transportation process the embedded device cooperates with the environment to achieve its goal. (2) Transportation system driven: The environment around the goods takes the decisions. Here a possible process looks like that: With the arrival and identification of a good at the entry gate the environment creates autonomously the specific transport order that from now on escorts each autonomous transportation unit that handles these goods.

These autonomous transportation units are able to make their own decisions and to cooperate with each other. They can decide which transportation order they accept and how to deliver the good to the desired sink. Furthermore they achieve a high degree of freedom through the ability of cooperation with other units. To raises the degree of flexibility even more the transportation units consists of 1...n interchangeable modules. These modules are pluggable in vertical and horizontal direction to build an unique transportation unit. The abilities that these unique transportation unit now has, is derived from the currently used modules (Günthner et al., 2008b). The control paradigm of smart independent, autonomous transportation units shall lead to positive emergence with the promise to cope with the high dynamic of logistic systems (Windth, 2006). The main goals of such distributed systems are increased flexibility, carrying speed and throughput, increased maintainability, scalability and reliability through redundancy and decreased livecycle cost.

3 RELATED WORK

This section will provide an overview of the state of the art of robotic MFSs. It will not only focus on systems that are available on the free market but also on the current research. This paper will separate these products/projects into the following three categories: (1) central control and autonomous behavior: These MFSs are controlled via a central instance, where all the decisions regarding the transportations order are scheduled. These robotic systems are usually application specific and, hence there is no need for cooperation between the robots to fullfill a goal. (2) local control with autonomous behavior: Because of local control the presented system are scalable, flexible and failsafe. Thereon, the installation and reconfiguration costs are lower in comparison to central controlled systems. The robots act autonomously and don't cooperate. (3) local control with swarm behavior: robots cooperate with each other to achieve a common goal. This requires the ability to communicate with each other and to make local decisions.

Central Control with Autonomous Behaviour. The Kiva warehouse management system by Kiva Systems (Guizzo, 2008), (Wurman et al., 2008) is a commercial system for commissioning of products in stocks with small parts. The stock consists of many adjustable shelves with a matrix like structure on a flat ground. Small autonomous robots (Drive Units, DU) are able to drive under a shelf, lift it up and bring it autonomously to other locations, e.g. a picking station. Orders are accepted from a warehouse management system by a central computer (Job Manager, JM) which is responsible to schedule the DUs and picking stations as well as the shelf space at the station. After receiving a transportation request from the JM the DUs are responsible for their own task planning, path planning plus motions planning and control. Communication between the agents is done with XML messages at the higher level transmitted by wireless technologies. Because of the agent based architecture the system is highly scalable and can grow with the requirements, where the centralized Job Manager is a limiting part. A disadvantage is the limited field of application domains. The system is mainly useful for order picking processes that have a high degree on manual work.

Local Control with Autonomous Behaviour. Multishuttle is a product by Siemens Dematic AG developed together with Fraunhofer-Institute for Material Flow and Logistics in Dortmund,Germany. The modular system consists of autonomous vehicles driving rail-bounded inside of a warehouse system. The vehicles can drive at the horizontal direction and they can autonomously load and unload product carrier (at the same time). Rails are laid in several stacked levels. They take care for both - guiding the vehicles and energy delivery. Movement at the vertical direction is done by lifts. Transport orders are communicated to the vehicles by WLAN. The rail-bound energy delivery leads to a lower weight and price. But for the same reason the vehicles are bound to the warehouse, they can't deliver goods in the total area of the delivery station. In contrast to traditional warehouse systems like a shelf access equipment, the Multishuttle system is scalable. Thereon, the throughput be increased with some additional vehicles.

The system Servus form the Austrian company Servus Robotics (Servus Robotics, 2006), (Robotics, 2005) is intended for intra-logistics assembly automation. Like Multishuttle the system is rail bounded. The vehicles are able to act autonomously. They accept transportation orders through an infrared or WLAN interface. Additional information of the goods, like necessary processing steps, is stored at the vehicle. The goods themselves don't need to be intelligent. Additional actuators can be build upon the vehicles, e.g. to be able perform processing steps while the goods are carried. Energy is not supplied by the rail, unlike the Multishuttle, instead each vehicle has its own fast rechargeable energy supply.

Local Control with Swarm Behaviour. Another project investigating in robotic conveyers is the KARIS project of the Institute for Conveying Technology and Logistics (IFL) at the University of Karlsruhe, Gemany. They have presented a robotic transportation system (Baur et al., 2008), that consists of several homogeneous transportation units which are able to drive at the floor or stand at the floor while acting as a conveyer. The wheels thereby are turnable at 360 degrees providing free movement at the horizontal plane. A KARIS unit is able to carry payload by its own or if the charge is too large or too heavy, many KARIS units build a swarm and carry the payload together. If a large throughput is required several units can be combined to build a continuos conveyer with sorter function. Swarm building and acting is the actual research work at IFL.

The institute for Materials Handling, Material Flow, Logistics (IML) in Munich (Gemany) proposed a concept for future material handling systems (Günthner et al., 2008b), (Günthner et al., 2008a) consisting of low-scale autonomic transportation units. All transportation vehicles are small and have a simple and basic design causing a low price because of high volume production. For special roles they shall be able to be equipped with manipulators like a lift fork, roller or belt conveyer. They shall be autonomous with their own intelligence and communication options. If a task can't be achieved by a single vehicle, more of them shall form a swarm and act together.

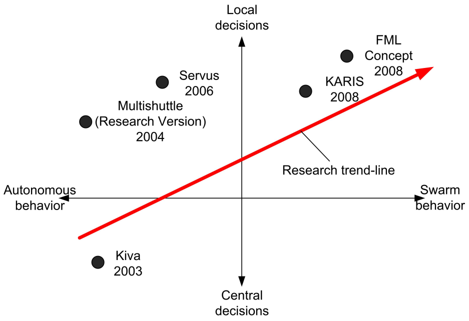


Figure 1: Trendline of robotic material flow systems.

Conclusion on the Related Work. As it is depicted in Figure 1 there is clear trend line towards autonomous robotic systems that can act in a swarm to achieve common goals in MFSs. The discussed research projects KARIS is an elaborated robotic system that shows that these systems can act in two ways: as a discontinuous or continuous conveyer. Nevertheless, this systems has no flexibility regarding the transported goods. The concept of the IML has this ability because of its changeable manipulators. Thereon, it can pickup different kinds of goods, like pallets and mixed cargo. But both approaches are limited to operate on the floor. In the following this paper will present an promising approach that self adapts to the transported goods and the layout of the transport area.

4 COGNITIVE MATERIAL FLOW SYSTEMS

A transfer station scenario with cognitive transportation units (CTU) is drafted in figure 2. Here the station consists of entrance and exit areas, a storing area and a working area between the entrance and the exit. Goods are delivered, e.g. by trucks at the ports of the entrance area. On the other side goods are removed at the ports of the exit area. The CTUs (red vehicles in the picture) are responsible for good transportation (goods are represented as pallets at the picture). According to the requirements CTUs can act as continuous conveyors or discontinuous conveyors equipped with different manipulation units depending on the kind of goods. In general the CTUs are modular. Because of the unified model it is possible to combine arbitrary modules in vertical and horizontal manner. Communication between the CTUs and the environmental sensors is done wireless. In figure 2 two types of communicating sensors are shown. The red one is a mobile sensor (CTU in its rule as a sensor); the green one is a fixed location sensor. The shown scenario raises some questions to the aspects of modularity, energy supply, sensor data delivery, dynamic sensor integration, and communication. These aspects are highlighted in the following sections.

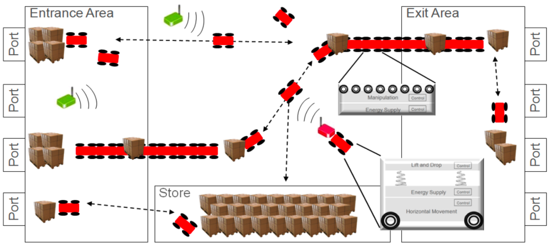


Figure 2: A cognitive material flow scenario.

5 COGNITIVE TRANSPORTATION UNIT

Modularization. As it has been stated in the previous section the CTU consists of modules that have different abilities. This key approach of modularization gives cognitive MFSs the flexibility to adapt not only to fluctuation in the load but also to adapt to different kind of goods that need to be transported. So needs a parcel an other manipulator as an mobile shelve. The decomposition of the CTU leads to the following kinds of modules: (1) A conveyer module for fast transportation of goods in horizontal direction. If many of this modules stands next to each other they can act like a continuous conveyer system. (2) an elevating module, for movements in vertical direction to lift or lowering goods. This module is needed, e.g. to transport an mobile shelve or to compensate difference in the height during the pick up process of a good. (3) A manipulator module can either work in combination with the conveyer module or with the elevating module to grab or release a good. To be able to handle different kind of goods, e.g. pallets or mobile stock, there should also different kind of manipulators. (4) A power supply unit, that energizes the system. Like the other modules this is also designed-for-purpose. So can the power supply unit be based on batteries or can even be a stationary power supply to support or load other CTUs. (5) The movement unit allow the CTU to act as an discontinuous conveyer system, with different kind of this modules the CTU are able to drive on the floor and also to drive into a stock.

Because the power supply is an mandatory on, there are at least 20 CTU combinations possible, which will leverage the MFS to respond in an even more flexible way. The modules can either be stacked vertical (on top of each other) , e.g. to build discontinuous CTU (with modules (1)-(5)) or horizontal to build a continuous CTU with module type (1). There should be no limitation regarding the used number of modules to build an CTU. The modularization requires new ways of communication, control and perception between the CTU modules and then between the CTUs themselves.

Towards a CTU System Architecture. The previously described modularization has also its impact on the computational architecture of the CTU. Here are different communication channels mandatory: (1) The vertical channel handles the communication between the modules of the CTU. This communication channel have hard criteria regarding the reliability of data transmission and real-time requirements, e.g. because of used closed control loops that set the speed of the CTU wheels. Thereon this channel is typically wired and represented by a field bus with high bandwidth, like CAN or Flexray. (2) A horizontal channel: This communication interface is used to interact with (a) other CTUs and (b) the cognitive environment, which is discussed in detail in the next section. A communication between the CTUs occurs, when goods need to be transfered between them or when transportation orders need to be negotiated or for swarm cooperation. The channel (2) with its different interfaces is a wireless one, that has to fulfill special requirements regarding energy optimization or real-time. Because of the numerous communication interfaces, that have divers API and physical characteristics, an abstraction layer is needed that makes the communication to the CTU modules and CTU itself transparent. This abstraction layer (middleware) has to support Quality of Service parameters that specify the fault tolerance or real-time level of the communication. Furthermore this Middleware has to support different embedded devices, that are used inside the CTU modules or the sensors from the cognitive environment. This can differ in used microprocessor architecture, e.g. from 8-Bits to 32-Bits architectures and their program storage size (flash-size).

Energy Efficiency. As stated in (Overmeyer et al., 2007) optimization under changing general conditions has still to be favorable for the overall logistic system. A battery is a limiting factor for the time *t* of useful work. The strategies the CTUs are using to fulfill a task directly influence the energy drain of the battery. For example, if the CTUs try to greedily minimize the waiting time criterion in (Overmeyer et al., 2007) they drive with the highest speed to the nearest source, catch a palette and drive with highest speed to the sink of the palette. If all CTUs choose this egoistic strategy they start to block each other because the most efficient path from the source to the sink is overused. At the end they may not be able to fulfill the task at all because the strategy is to energy consuming for the whole system. For this kind of system the optimization criteria has to be reformulated in a way considering energy consumption. A trade-off has to be found between the energy consumption and the application needs. As an example, for high prioritized costumers the focus lies on speed and energy is less considered and for low priority costumers energy is considered more. For the longest availability of the overall system a uniform distribution of energy might be useful. But, this optimization may lead to the fact that all CTUs have to recharge at the same time and the system is unavailable; an additional optimization criterion might then be that the mean number of recharging CTUs is e.g. not higher than 15% of all CTUs in the system. The following assumptions are made: Batteries recharging takes a significant long time. During this period the CTU is not able to do valuable work, e.g. it is unavailable. Three states characterize the (simplified) CTU: driving, turning and lifting. Every system state has characteristic power consumption at a time *t* and the overall system energy consumption at time *t* is:

$$E_{AutSys} = E_{Driving}(t) + E_{Turning}(t) + E_{Lifting}(t) \quad (1)$$

Additionally the power consumption depends on the load (kg), the speed $(\frac{m}{s})$ and acceleration $(\frac{m}{s^2})$ of the CTU. If the CTUs now receive a task they have to (1) choose a strategy for the task and (2) estimate the time they will spend in each system state, (3) calculate the overall energy for the chosen strategy and (4) compare the energy consumption with the given optimization criterion - if this is violated go back to (1). With this approach an energy optimized strategy for logistic systems can be found.

6 COGNITIVE ENVIRONMENTS

The basic cognitive capabilities are perception, reasoning, learning and planning. A cognitive environment consists of systems that show a similar strategic behavior like human individums do. For the modelling of cognitive human processes, e.g. cognitive systems, different architectures have been developed (Laird et al., 1987) (Anderson and Lebiere, 1998). In these architetures the perception (sensing) of the environment and the storage (memory) of sensor data are important basics for learning and reasoning. In former logistic systems like forklifts the only sensory information comes from the limited human perception. The same problem have autonomous logistic systems e.g. driver-less systems that rely on built-in sensor information. Their view of the environment is limited to the perception ability of the integrated sensors.

Sensor Abstraction. In the proposed transfer station scenario sensors are usable by everyone. This expands the view of the environment to the whole scenario. Every autonomous system is able to get this view to raise the correctness of their decisions. For example, in (Riedmaier, 2008) the only sensor information was the soil condition of the track. With only this information the speed of forklifts could be optimized and the handling of palettes could be increased about 5%. In the proposed transfer station scenario external sensors are attached to the walls able to detect movement of non-cooperative and cooperative systems. With the help of these sensors a CTU is able to drive with high speed towards an intersection where it otherwise would not be able to sense if another robot is crossing it and, therfore had to reduce the speed. To unify sensor communication wireless- and wired communication media have to be integrated into every sensor. So, every sensor can act as fixed external sensor or if necessary can be attached to an autonomous robot to improve their sensing capabilities. As the cognitive logistic system allows a task adapted flexible restructuring, fixed sensors are a problem. To allow flexibility, the CTUs are able to move the sensors to a new position. From the modeling point of view, sensors are agents running a sensing task for a long time. In the restructuring process of the logistic system they get a new task and autonomously decide if they can fulfill this task (1) with their sensing capabilities and (2) at their actual position. If (1) is not fulfilled they have to reject the task, if (2) is not fulfilled they can require help from an autonomous robot to replace them to a better position. Agents are the abstraction of the real embedded devices in the proposed cognitive logistic system and consequentially sensors are abstracted with agents as well - this unifies the whole transfer station scenario world view. Further usage of the network of sensors can be as a communication relay for the CTUs and sensors. Due to limited propagation of radio waves in logistic in-door facilities the assumption that all sensors can communicate with each other is not true. Therefore, using the multihop capability of modern wireless sensor networks is a good way to extend the range of the sensors and the CTU's communication system.

Sensor Data Memory. Sensor data is important for decision making of the control algorithms. Therefore, it needs to be protected from communication and sensor node failures. For industrial environments WirelessHART (HART Communication Foundation, 2007) is a standardized protocol for reliable wireless communication and can be used in the proposed transfer station scenario. It provides robust self-organizing and self healing mechanisms to encounter communication failures. But networked sensors have far more potential, in a cognitive system they can be used as distributed observers. A distributed observer is a sensor with its own memory that stores a snapshot of the past. It is similar to the human short-term memory (with low-capacity) and is used in many cognitive modelling architectures (Laird et al., 1987) (Anderson and Lebiere, 1998). This kind of sensors can answer questions about situations of a larger context, which is usful for coordination and optimization purposes. Distributed sensors have an area to observe. For example, a fixed sensor knows about the robot traffic in his area and can therefore give a usage estimation of the path belonging to his observation area. Technically, sensors now have to store their data instead of just sending real-time data to the CTUs. The CTUs then ask the sensors for certain events in their stored history snapshot. For fault-tolerance reasons, sensors are allowed to replicate their data to other fixed or mobile sensors. They can use different replication strategies to trade-off data availability for energy and vice versa.

7 CONCLUSIONS AND FUTURE WORK

Present state-of-the-art projects were considered as too domain specific and not able to raise the flexibility of logistic systems comprehensively. Therefore, this paper proposed modular principle that raises the flexibility of the system. Energy is an important factor for battery driven autonomous robots, therefore strategies for the trade-off between energy consumption and timelines were discussed. Furthermore, a unified sensor integration scheme was proposed that raises the cognitive perception ability of the whole logistic system and a sensor data concept that enables the idea of a distributed observer was shown. At the moment the proposed models are being implemented and in a next step they will be simulated. The goal of the simulation is to find the best granularity of the modularization and to find the best cooperating strategies for autonomous logistic systems. As a next step a test bed implementing figure 2 for validation of the chosen strategies will be created.

REFERENCES

Anderson, J. R. and Lebiere, C. (1998). *The Atomic Components of Thought*. Lea.

Baur, T., Schönung, F., Stoll, T., and Furmans, K. (2008). Formationsfahrt von mobilen, autonomen und kooperierenden Materialflusselementen zum Transport eines Ladungsträgers. *Fachkolloquium der Wissenschaftlichen Gesellschaft für Technische Logistik e. V.*, 4.

Guizzo, E. (2008). Three engineers, hundreds of robots, one warehouse. *IEEE Spectrum:*, pages 26–34.

Günthner et al. (2008a). Vom Prozess zum Ereignis - ein neuer Denkansatz in der Logistik. In *Jahrbuch Logistik*, pages 224–228.

Günthner, W. A., Durchholz, J., Kraul, R., and Schneider, O. (2008b). Technologie für die Logistik des 21. Jahrhunderts. In *Kongressband zum 25. Deutschen Logistik-Kongress*, pages 360–393.

HART Communication Foundation (2007). *WirelessHART Technical Datasheet*. HART Communication Foundation, 1.0b edition.

Laird, J. E., Newell, A., and Rosenbloom, P. S. (1987). Soar: An architecture for general intelligence. *Artif. Intell.*, 33(1):1–64.

Overmeyer, L., Falkenberg, S., Heiserich, G., and Jungk, A. (2007). Innovative Gestaltung von Intralogistik durch Kopplung kleinskaliger Systeme. In *16. Deutscher Materialfluss-Kongress*, pages 171–179.

Riedmaier, S. (2008). Neue Möglichkeiten für Schmalgangstapler. *Hebezeuge Fördermittel*, 12:762–763.

Robotics, S. (2005). Na, Servus! *Automation*, 6.

Scholz-Reiter, B., Jagalski, T., and Bendul, J. (2007a). Bienenalgorithmen zur Selbststeuerung logistischer Prozesse. *Industrie Management*, pages 7–10.

Scholz-Reiter, B., Jagalski, T., and de Beer, C. (2007b). Selbststeuerung logistischer Prozesse in Produktionsnetzen. *Industrie Management*, pages 19–22.

Scholz-Reiter, B., Rekersbrink, H., and Freitag, M. (2006). Kooperierende Routingprotokolle zur Selbssteuerung von Transportprozessen. *Industrie Management*, 22.

Servus Robotics (2006). Wie von Geisterhand. *Automation*, 4.

Windth, K. (2006). Selbststeuerung intelligenter Objekte in der Logistik.

Wurman, P. R., R., R. D., and Mountz, M. (2008). Coordinating hundreds of cooperative, autonomous vehicles in warehouses. *AI Magazine*, 29/1:9–19.

THE FINS PROTOCOL FOR COMPLEX INDUSTRIAL APPLICATTIONS
A Case Study

Júlio Costa, Nuno Carvalho
Industrial Electronics Department/ALGORITMI, University of Minho, Guimarães, Portugal
a40528@alunos.uminho.pt, a42114@alunos.uminho.pt

Filomena Soares
Industrial Electronics Department/ALGORITMI, University of Minho, Guimarães, Portugal
fsoares@dei.uminho.pt

José Machado
Mechanical Engineering Department/CT2M, University of Minho, Guimarães, Portugal
jmachado@dem.uminho.pt

Keywords: Complex Control Systems, Industrial networks, Industrial protocols, PLCs networks.

Abstract: This paper presents a comparative approach on the use of different industrial networks configurations and industrial communication protocols. Some aspects, that may influence the right choice of the most indicated protocol for each industrial network configuration, are discussed. It is presented a case study and two configurations networks implementing two industrial communication protocols. The respective advantages and disadvantages are presented. All the detailed aspects including the data exchange are presented too. The obtained results are extrapolated for other similar industrial applications.

1 INTRODUCTION

This works appears on the context of developing and implementing new solutions for industrial networks implementation. This line of research is being developed by a team from the School of Engineering of University of Minho and involves some departments of the School.

The first results, here presented, are the first one obtained from an initial study that it is intended to be more complex and exhaustive.

Industrial communications have significantly evolved since their appearance in the 1970s. Faster and more reliable communication protocols have been proposed and deployed in industrial applications (IEC 61784-2).

The necessity that the companies have to improve their competitiveness has lead to many developments on this field, related with more complex industrial networks applications and with more complex communication protocols elaboration. This increasing of competitiveness is a constant objective for all the companies in general and for the Portuguese companies, in particular.

In order to facilitate the management and control of manufacturing processes it is, currently, very important the flexibility of the implemented management and control systems for the manufacturing processes. For that accomplishment, it is necessary a fast access at the information, means that allow a fast decisions according the manufacturing process behavior and, more important, the possibility of improvement of the manufacturing systems efficiency.

With the development of the communication of the industrial networks, with the evolution of the industrial communication protocols and the increasing of the exigency level - characteristic of the manufacturing process control - the knowledge and the know-how associated at these realities is becoming crucial on the development and improvement of competitiveness of the industrial companies.

In this paper it is intended to compare and conclude about industrial network configurations and to compare industrial communication protocols too. Some propositions for the best communication protocol to be applied on some industrial network configurations are also presented.

In order to achieve the main goals proposed on this paper, the paper is organized as follows: Section 1 used to present the challenge of the work. In section 2 it is presented a background about industrial networks and industrial communication protocols. Further, section 3, it is presented a case study that permits the application of two industrial network configurations and, also, two industrial communication protocols application. Section 4 is devoted to the presentation of the developed work followed by the Section 5, where are presented and discussed the obtained results. Finally, section 6, there are presented the main conclusions of this study and some guidelines for the future work.

2 BACKGROUND

In this section it is presented a brief overview of industrial networks and some of the most used communication industrial protocols.

2.1 Industrial Networks

The industrial networks can be implemented considering several types of controllers.

Among these controllers, the Programmable Logic Controllers (PLCs) are the most used due to their robustness when submitted to industrial environments which are characterized by adverse conditions (like magnetic fields, vibrations, dust, noise, among others).

With the current increase of industrial networks, the availability of user friendly environments and software tools that allow a better use of the industrial networks capabilities is also improved.

The access to different network nodes must be fast and must allow supervising all the processes even if they are physically independent.

An industrial network may have different components; therefore, it implies that the connection type between these components may be different, leading to the need of using sub-networks. Thus, to define some order and criteria on these links, it can be considered a set of hierarchical levels related to a common industrial network. These hierarchical levels can be defined by different ways and using different criteria. Nevertheless, the pyramid CIM (Computer Integrated Manufacturing) (ISA-dS95.01-1999) is a good approach for illustrating these levels (Figure 1).

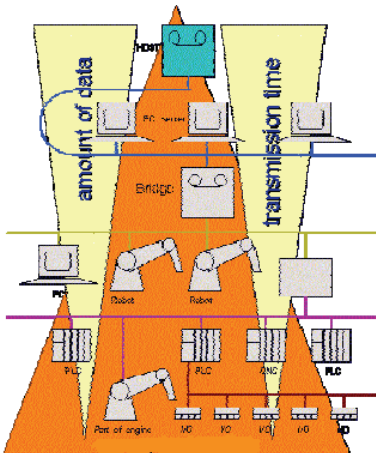


Figure 1: CIM Pyramid (ISA-dS95.01-1999).

The CIM Pyramid is divided in levels concerning the type of application to be controlled. The considered levels are Management, Control, Process, and, finally, the Inputs and Outputs variables.

The Management level is concentrated on all the information concerning to the network. Usually, it is used a Personal Computer (PC) in which it can be seen all the performance of the plant using a Supervisory Control and Data Acquisition (SCADA) system (Pires and Oliveira, 2006).

At the Control level it is established the connection between PCs and PLCs.

The Process level is characterized by the controllers and PLCs for the industrial process control.

Finally, the Inputs and Outputs level includes the sensors and actuators devices. This is the lowest level of the CIM Pyramid. It is also the closer level to the plant, where the network is applied.

In the Management level, as in the Control level, the type of network used is a Local Area Network (LAN) (IEEE 802.1AB-2005) as for example the Ethernet (Felser, 2005).

At the Process level other types of networks are used. One of the most implemented is the PROFIBUS network (PROFIBUS International, Liu *et al*., 2007). Also, at this level, the Actuator/Sensor Interface (AS-I) network may be used (Lee, 2001).

The Ethernet appears with the main goals of reducing costs, increasing dependability, sharing the information and the physical resources in the same transmission environment by using a coaxial cable.

The Ethernet technology has, as physical devices, the coaxial cables with small and large diameter, or the plaited pair of cables.

With the Ethernet network some topologies are possible: star, tree or ring type configurations. For the communication between the several devices there exist some transmission environments: the Simplex, where the transmission is done in a unilateral direction; the half duplex, where the transmission is done from and to each device; and the full duplex, where each device simultaneously transmits and receives information.

The Profibus network has different functionalities for its communication protocols: the profibus Fieldbus Message Specification (FMS), the Distributed Peripherals (DP) and the Process Automation (PA), where the physical transmission is done by RS485. The profibus FMS is a protocol used on the PCs and PLCs communication, but Ethernet network is substantially increasing on this domain application. The DP profibus is used for the communication between small PLCs and for the communication between PLCs and the controllers. With the transmission environment RS485, it can be used a complexity until 32 devices, including the first initial node of the connection. Usually, this node is a small PLC. The PA profibus network is implemented to link sensors and actuators, connected to a master PLC that centralizes all the relevant data to the control system.

The AS-I network is used for the lowest level of automation systems. There are about 80 international developer companies that use this type of network.

This is a low cost network and easy to expand. Like Profibus, it is allowed the use of a maximum of 32 devices. The maximum allowed length is about 100 meters.

2.2 Industrial Communication Protocols

With the increasing of the competitiveness and the set of different PLC products existing in the market, it is usual, in an industrial plant, to coexist different types of PLCs. The communication between these systems is necessary in order to accomplish all the benefits proposed by the industrial networks.

For the communication between these physical devices, different solutions in the set of industrial communication protocols are used. The advantages of universal protocols (open protocols) seem natural, because they allow the exchanging of data and information between different types of systems.

In this group of protocols, one of the most used is the serial communication protocol. But there are others, like the Synchronous Serial Interface (SSI) and the Bi-directional Synchronous Serial Interface (BiSS). As open protocol, the Profibus (previously described) is also very used.

There are, also, other protocols that are restrict and proprietary of the controllers' manufacturers. For instance, the Hostlink and the Factory Interface Network Service (FINS) protocols are two examples of a large set of these closed protocols (Kizza, 2005).

The main advantage of using closed protocols is improving the simplicity of network implementation and configuration. The manufacturers of these protocols have well adapted software tools and a very structured set of configurations that considerably help the designers.

The main advantage of open protocols is that they can be used and shared by different devices from different manufacturers. Using these protocols it is possible to exchange data and information between several commercial devices. The characteristics of these protocols are similar, no matter the device manufacturer, so different companies use them as a way to promote their own products and also to increase the competiveness between the device manufacturers.

In fact, if it is necessary to expand the industrial network, adding new devices, these protocols have real advantages when compared to the closed protocols. In addition, they are at low cost. The main reason to decrease the cost of these protocols is that the devices manufacturers intend to increase the competitiveness (Kizza, 2005).

3 CASE STUDY

The automated line production which was used in this study is a didactic Modular Production System (MPS) of the Mechanical Engineering Department Automation Laboratory of University of Minho, in Portugal. Although being didactic, this equipment is a well achieved simulation of a real system. Its command module is being used in real line production systems. All the control tasks are assured by a Programmable Logic Controllers (PLCs) Network specially designed for the purpose (figure 2).

This system is composed by five modules, named as follows:

Module 1 – Distribution
Module 2 – Test

Module 3 – Processing
Module 4 – Transport
Module 5 – Separation

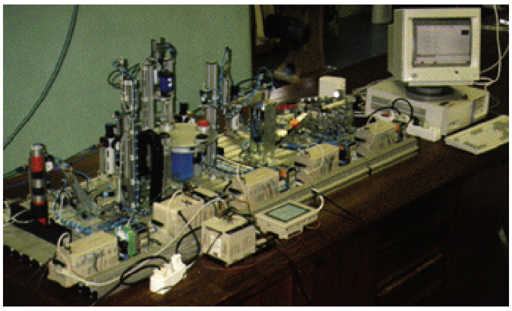


Figure 2: Modular Production System.

These modules have an independent control, each one being controlled by a single PLC, all the PLCs being controlled by a PC.

The identification of the component type is made in the module 2. The control programming assure that on the module 5, the components are assorted by size, colour or material, as well as rejected components, each one being directed to an appropriated conveyor.

In order to obtain some results comparing the communication protocols it was decided to configure the control structure in two different kinds of networks really implemented in industrial systems.

In a first step, the PLCs corresponding to each module were connected in a network, in a parallel configuration, as shown in figure 3. All the networked PLCs are at same level of control (network N1).

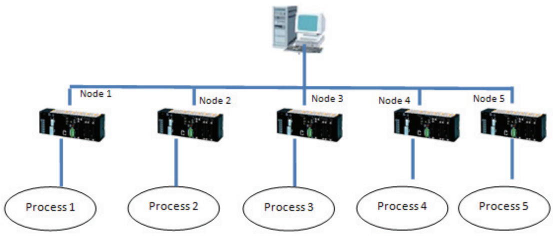


Figure 3: Scheme of the implemented network (step 1 of the study).

In a second step, the MPS was separated into five independent modules, where each one represents a sub-network, as illustrated in figure 4, network N2.

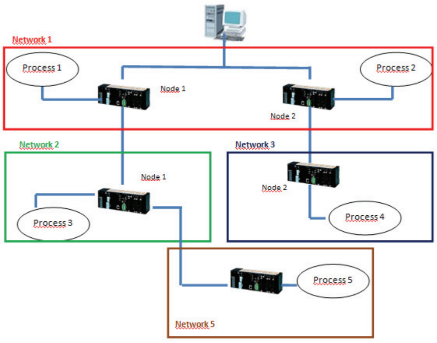


Figure 4: Scheme of the implemented network (step 2 of the study, with sub-networks).

The used protocols for this study were the Hostlink and FINS protocols from OMRON Company (www.omron.com).

On the first step approach it was used the Hostlink protocol and on the second step approach it was used the FINS protocol.

4 DEVELOPED WORK

Networks N1 and N2 were implemented (www.omron.com).

In the case of network N1, Hostlink protocol is often employed. Each PLC has a dedicate identification number (ID). The configuration frame includes: the PLC ID number, the definition of the action to be performed, e.g. to read a process variable value (counting pieces in a process line production) or to send a command value to the working system (switching on an actuator).

The command frame includes the following fields (Figure 5): constant parameters definition, the first one indicates the frame starting point and the terminator parameter designates the ending point; the node number is the PLC ID number for communication; the header code and the text are the definition of the action and the data to be exchanged in the communication process, respectively.

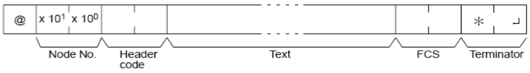


Figure 5: Hostlink command frame.

The response frame to the previous command is shown in Figure 6. The start and ending points are identical to the command frame. The difference is in the end code parameter definition which corresponds to an indicator of success or error in the transmission line established.

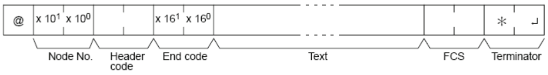


Figure 6: Hostlink response frame.

This protocol is adequate for using in a small network with parallel PLCs configuration which can be a constraint when working with complex control systems. To overcome this limitation and when sub-industrial networks are implemented (Network N2), FINS protocol is an adequate solution.

Figure 7 shows the structure of the command frame sent to the network to communicate to the PLC. The frame is similar to the Hostlink protocol but it includes the specific FINS command, the action to be performed, the target sub-network and the corresponding PLC, in order to establish the communication. This frame is detailed in figure 8.

In the frame it must be defined the destination PLC and to where (which network and PLC) the response message should be returned.

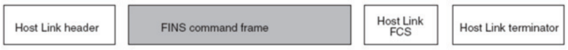


Figure 7: FINS protocol frame.

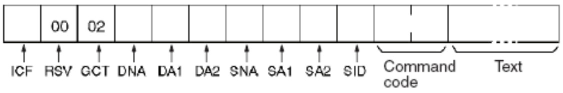


Figure 8: Parameter specification in FINS command.

Between the FINS characteristic parameters, the parameters DNA, DA, SNA and SA1 are particular important, as they define the PLC communication command target and the destination of the response message. DNA is the destination network address, DA1 is the destination node address and in order to define to where the response message should be sent, the SNA, source network address ans SA1, source node address, must be configured.

5 RESULTS DISCUSSION

Hostlink and FINS protocols were tested and compared in two types of industrial networks: a simple network N1, represented in Figure 3 and a more complex one, N2, shown in Figure 4.

First, network N1 was tested. Two values were read from the PLC memory position Core Input Output (CIO) starting from position number 10. Table 1 shows the configuration of the command frame for writing, sent by the personal computer (PC) to the PLC, by using both protocols, Hostlink and FINS.

Table 1: Command frame for writing.

Protocol	Frame
Hostlink	@00WR000A0001000237*
FINS	@00FAF000000000102B0000A0000020001 000200*

Table 2 shows the response frame sent by the PLC to the PC, also employing both protocols.

Table 2: PLC response frame to writing command.

Protocol	Frame
Hostlink	@00WR0045*
FINS	@00FA00400000000102000040*

The test was repeated but for reading command of two values in PLC memory CIO which are in position 9. Tables 3 and 4 show the command for reading sent by the PLC and the corresponding response frame sent by the PLC.

Table 3: Command frame for reading.

Protocol	Frame
Hostlink	@00RR000900034A*
FINS	@00FAF000000000101B000090000 0278*

Table 4: Answer table from PLC.

Protocol	Frame
Hostlink	@00RR0000000001000243*
FINS	@00FA00400000000101000000010 00240*

Analyzing Tables 3 and 4 it is verified that the frame lengths are different. This is due to the fact that the configuration parameters are diverse. In this case, network N1, Hostlink protocol is easier to configure, the frames are shorter, being more adequate for the application system.

A second configuration was tested, network N2, where the MPS process is controlled by the PLC connected to a specific sub-network.

As Hostlink protocol cannot be used in industrial systems where sub-networks are configured, only the FINS protocol was implemented.

Table 5 presents the command for writing sent from the PC to the PLC positioned in a sub-network and the corresponding PLC response command. The command consists of writing two values starting in position 10 of CIO memory.

Table 5: FINS writing command/response frames.

FINS protocol	Frame
Command code	@00FAF800002010100000000000102B0000A000002000100020A*
Response command	@00FA00C000020000000102000001010000000025002232*

Table 6 tests the command for reading two positions in PLC (placed in the sub-network) CIO memory starting from position number 10.

Table 6: FINS reading command/response frames.

FINS protocol	Frame
Command code	@00FAF800002010100000000000101B0000A0000020A*
Response command	@00FA00C000020000000101000001010000000025002231*

As it can be seen in Tables 3 to 6, the frames lengths are different in all tested cases, being the FINS frame larger than the Hostlink. FINS protocol needs more parameters to configure the communication. For a correct and successful data transmission, all the FINS parameters must be defined even if they have null value.

In the Hostlink code, the writing and reading command frames make use of two specific characters, namely, RR and WR, respectively.

In FINS protocol the code is implemented using two hexadecimal values, four characters. For example, the code 0101 is for reading and 0102 is for writing. Both can be used to read and write in any PLC memory position. On the contrary, Hostlink protocol needs other commands to write in a different memory position. Both frames signal when the communication is successful.

Apart from having different frame lengths, Hostlink and FINS have also different data transmission capacity. FINS has a maximum capacity of 1115 characters while Hostlink has a lower capacity, 131 characters.

In summary, with FINS protocol we can access the whole network, including the PLCs that are in a sub-network. By using such a network it is possible to monitor and manage the whole line production from a working place.

For a correct use of both protocols, it is necessary to know the network type. If two PLCs are connected by a profibus link, the PLC slave cannot be accessed if both master and slave PLCs are in the same network as the PC. In profibus network, the slave device periodically sends to the master the memory positions, configure by the network manager.

6 CONCLUSIONS AND FUTURE WORK

This paper presents part of the on-going work regarding industrial networks design for complex systems.

An automated line production, a didactic Modular Production System (MPS), was used as the case-study. In spite of being didactic, this equipment is a well achieved simulation of a real world controlled system. All the control tasks are assured by a Programmable Logic Controllers Network specially designed for the purpose.

The communications protocols Hostlink and FINS used as information coordination methods between the PLCs and the production equipment control system were described and tested.

Although being a proprietary communication protocol, FINS becomes particular important due to its simplicity, economy of time and development costs.

In the near future, we are going to implement, test and discuss other types of industrial protocols using the demonstration system. An extensively comparative study for evaluating the protocols' performance will be carried on.

REFERENCES

IEC 61784-2: Industrial communication networks – Profiles - Part 2: Additional fieldbus profiles for realtime networks based on ISO/IEC 8802-3, available at www.iec.ch

ISA-dS95.01-1999, Draft Standard, Enterprise - Control System Integration, Part 1: Models and Terminology, November, available at 1999www.mel.nist.gov/sc5wg1/isa95part1-d14.pdf)

Pires, P. S. M. and Oliveira, L. A. H. G., 2006, Security aspects of SCADA and corporate network interconnection: An overview, *Proceedings of the International Conference on Dependability of Computer Systems (DEPCOS-RELCOMEX'06)*, pages 127–134, May.

IEEE 802.1AB-2005 - IEEE Standard for Local and metropolitan area networks Station and Media Access Control Connectivity Discovery; available at www.ieee.org

Felser, M., 2005, Real-time Ethernet - industry prospective, *Proceedings of the IEEE*, Volume 93, Issue 6, June Page(s): 1118 – 1129

PROFIBUS International: PROFINET: Technology and Application, System Description, Document number: 4.132, Issue April 2006, available at www.profibus.com

Liu, J., Fang, Y., Zhang, D., 2007, PROFIBUS-DP and HART Protocol Conversion and the Gateway Development, *Second IEEE Conference on Industrial Electronics and Applications*

Lee, K., 2001, Sensor Networking and Interface Standardization, *IEEE Instrumentation and Measurement Technology Conference*, Budapest, Hungary, May 21-23.

Kizza, J.M., 2005, *Security Threats to Computer Networks,* Computer Communications and Networks Part II, Springer.

Spratt, Michael P.; Albrecht, Alan; Curcio, Joe; Dove, Dan; Goody, Steve, 1994, An Overview of IEEE 802.12 Demand Priority, HP Labs Technical Reports, available at
http://www.hpl.hp.com/techreports/94/HPL-94-17.html

PRODUCT REPRESENTATION TO SUPPORT VALIDATION OF SIMULATION MODELS IN COMPUTER AIDED ENGINEERING

Andreas Kain, Andreas Gaag and Udo Lindemann
Institute of Product Development, Technische Universitaet Muenchen, Boltzmannstr. 15, 85748 Garching, Germany
kain@pe.mw.tum.de, gaag@pe.mw.tum.de, lindemann@pe.mw.tum.de

Keywords: Model validation, System modelling, Multiple domain matrix (MDM), Flexible multibody simulation.

Abstract: Computer aided engineering (CAE) provides proper means to support New Product Development (NPD) by simulation tools. Simulation furthers early identification of product characteristics to reduce costs and time. The applicability of simulation models in NPD strongly depends on their validity, thus validating a simulation poses a major issue to provide correct experimentation results. The authors propose a matrix based approach to combine solution neutral system representation, solution specific product representation, and product behaviour in order to raise system comprehension to support validation of simulation models. A case study exemplifies the suggested approach. This paper illustrates the matrix based product representation at composing a flexible multibody simulation of a highly dynamic linear shafting machine tool. The approach supports preprocessing and validation of a flexible multibody simulation model.

1 INTRODUCTION

New product development (NPD) nowadays grounds on simulation tools provided by computer aided engineering (CAE). It becomes reasonable to evaluate engineering design in early stages before starting physical prototyping and thus enables early anticipation of product characteristics. Simulation also assists further development of existing products or establishing a line of products.

As summarized in (Musselman 1994; Robinson and Bhatia 1995; Robertson and Perera 2002) a simulation project comprises interpretive, developmental, and analytical facets. Modelling includes problem formulation, model conceptualization, data collection, model building, verification, validation, analysis, documentation and implementation.

Validation requires that the model is an accurate representation of the system being modelled taking into account the modelling purpose (Robinson and Bhatia 1995; Sargent 2004). The modelling purpose includes requirements on the model itself. Reasoning and derivation of conclusions by experimentation with the model requires successfully model credibility and thus completed validation. Thus validating a simulation poses a major issue to provide correct experimentation results. The authors propose a matrix based system representation to support validation of simulation models in CAE.

The paper contains in section 2 background information. Section 3 introduces a matrix based product representation to raise system comprehension and thus to support system validation in CAE. Section 4 illustrates a case study of supporting a flexible multi body simulation in further developing a machine tool. Section 5 discusses the application of the suggested approach in the case study. Section 6 concludes the paper.

2 BACKGROUND

According to (Sargent 2004) analysis and modelling derive a conceptual model based on the problem entity which represents the system. The conceptual model represents the system for a particular study. Implementation of the conceptual model leads to a computerized model. Validation of this computerized model by operational validation proves that the model's output behaviour represents sufficiently the problem entity for the model's intended purpose. VDI 3681 emphasises that validation is the proof that a system satisfies the requirements (VDI 2005). Bender narrows down the term validation as "*doing the right things*" contrary to the term specification that comprises "*doing the* *things properly*" (Bender 2005). Sargent summarizes and details several validation techniques (Sargent 2004): (1) Animation, (2) Comparison to other models, (3) Degenerate Tests, (4) Event Validity, (5) Extreme Condition test. (6) Face Validity, (7) Historical data validation, (8) Internal Validity, (9) Multistage Validation, (10) Operational Graphics, (11) Parameter Variability – Sensitivity Analysis, (12) Predictive Validation, (13) Traces, and (14) Turing Tests. Either the developer, or the user or a third party conduct one or more of these techniques either concurrently with the development of the simulation model or afterwards.

In product development concerned with not merely mechanical products several types of relations connect components systematically such as function, structure, and behaviour (Pahl and Beitz 1995; Ariyo, Eckert et al. 2006). A physical form with a specific structure characterizes design artefacts and enables to carry out function. The product structure comprises parts that interact amongst other and cause behaviour.

Based on these various approaches of product representation in NPS have been developed (e.g. seePahl and Beitz 1995; Lindemann 2007). Solution neutral and/ or solution specific system/product representations exist. Solution neutral representations support to lose fixation to specific physical solutions to further generating new conceptual ideas. E.g. functional modelling describes a system abstractly without sticking to specific solutions.

As Browning states the design structure matrix (DSM) is a well established method for handling complex systems (Browning 2001).

Relations within one domain such as function or structure fill the DSM in order to reveal interdependencies between elements. Maurer summarizes and details linking several DSMs by applying domain mapping matrices (DMM), that contain relations between elements of different domains, to gain multiple domain matrices (MDM) (Maurer 2007). Thus MDM methodology enables to interconnect solution neutral representation, e.g. by functional modelling, and solution specific representation e.g. by component structure. Interpretation and application of MDMs is a recent research task, e.g. interpretation of the meaning of specific patterns such as cycles (Biedermann and Lindemann 2008).

Based on system representations methods such as Failure Mode and Effects Analysis (e.g. ((VDA) 1999) or SAE J-1739) guide to reason about e.g. root causes in a structured manner by pointing to relations and evaluating these relations in NPD. They support to document problem solving tasks and application results in overall improvement of the product itself.

Multibody simulations reveal the kinematic behaviour of steep bodies. Schiehlen reviews the history of multibody systems in detail (Schiehlen 1997). A multibody system comprises bodies, force elements, and joints within a global reference frame (Schwertassek, Wallrapp et al. 1999). Additionally flexible multibody systems (fMBS) are capable to handle constrained deformable bodies that undergo large displacements, including large rotations (Shabana 1997).

3 METHOD

The authors propose a matrix based product representation to raise system comprehension and thus to support system validation in CAE. Besides the interconnection of the functional perspective on the system and the component structure of a product the suggested approach takes into account the dynamic behavior of a product (see Fig.1).

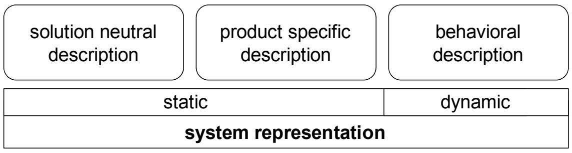
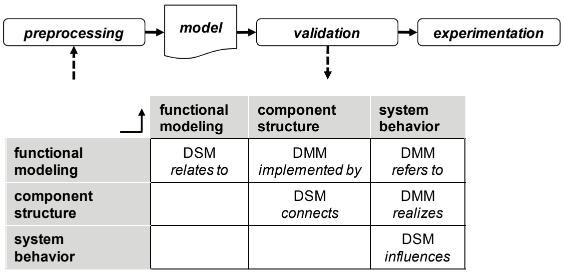


Figure 1: Components of the proposed system representation.

Creation and interpretation of the proposed product representation result in a deep understanding of the discussed product by raising awareness of interrelations between the considered domains. In CAE this understanding supports to define the modelling purpose properly. Additionally extensive collection of specifications enriches preprocessing of the simulation model (see Fig. 2).



	functional modeling	component structure	system behavior
functional modeling	DSM *relates to*	DMM *implemented by*	DMM *refers to*
component structure		DSM *connects*	DMM *realizes*
system behavior			DSM *influences*

Figure 2: Matrix based system representation supports modelling.

The matrix based system representation identifies relevant elements within the integrated domains and supports model conceptualization by incorporating experience and knowledge gathered along the product lifecycle. The product representation finally assists validation of the numerical simulation model applied in CAE. According to the taxonomy of validation techniques proposed by (Sargent 2004) the suggested approach furthers historical data evaluation, whereas the data proofs is the model behaves as the system does. The following section summarizes a case study carried out together with an industrial partner. This technique may be applied by the developer assisted by the user concurrently with the development of the model.

4 CASE STUDY

In this case study a specific linear shaping machine tool for fabricating crankshafts is modeled. The authors apply the matrix based product representation to support machine system simulation as fMBS.
Measuring operation induced oscillations at the machine tool itself confirmed the existence of structural oscillations. The fMBS model is to represent the structural bending induced by mass forces that cause lower fabrication quality by deflecting the tool from the manufacturing part. By representing this problem entity the simulation provides a means to finally evaluate design concepts of sub assemblies to reduce the structural machine misbehavior. Based on detailed product comprehension the main purpose of applying the suggested approach is to carry out system analysis to support validation of the simulation model.

Figure 3 depicts a simplified component structure of the shaping machine. It consists of (1) machine bed on that the (2) machine column is mounted. The (3) shaping head is connected to the machine column and comprises the (4) tool that moves highly dynamic up and down to machine the (5) part that is fixed to the machine bed. Within the shaping head the cutting tool moves up and down along vertical-axis up to 700 times per minute with a shifted weight of about 20 pounds and up to 20g. Due to the moved mass mass-forces induce bending of the whole machine structure that limits processing quality.

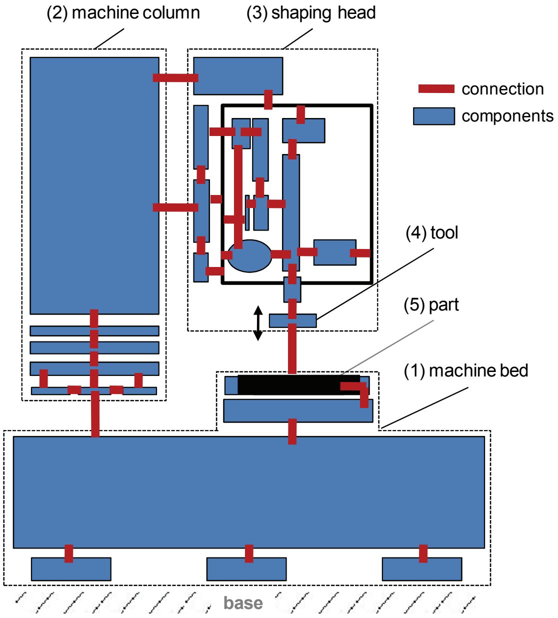


Figure 3: Simplified component structure of the shaping machine.

Physical components' specification, the assembly structure, and constraints between components are input data for modeling. Detecting tooth flank quality of the manufacturing part is an indirect measure of structural bending and denotes the machine tool behavior. Machine tool parameters (hydraulic system pressure, lateral offset of the column, …) as well as cutting parameters (feed, speed, …) influence the machine behavior. Each shaping application of particular crankshafts requires specific cutting parameters, whereas machine parameters are quite independent to select. fMBS is considered a means to raise the awareness of the actual structural bending during cutting conditions in a new scale.

5 DISCUSSION

Figure 4 exemplifies information extracted from the proposed matrix based product representation. Aggregated information summarized vital aspects of the system. It represents the domains component, function, and behaviour. The mechanical parts are connected by the flux of force (jack screw, machine column, and guide rail of machine column) and are interlinked to the functional modelling (perform feed, vary part position) and the machine behaviour (lateral offset of machine column). Additionally component specification such as stiffness, damping and geometrics is attached.

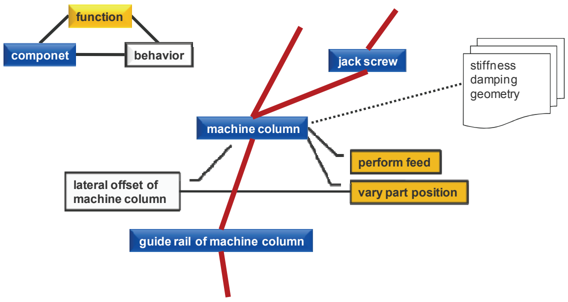


Figure 4: Aggregated cluster of information.

This kind of data aggregation supports to set up an enhanced fMBS simulation regarding important modelling parameters and thus supports focusing the modelling purpose and checking if the model's characteristic is as consistent to the system as needed. The assembly shaping head is a rather complex mechanical and functional structure and needs to be discussed in detail regarding the modeling purpose. Comprehension of interrelations within this assembly is a key to become aware of the system and thus vitally determines the preprocessing of the fMBS. When modeling the system the developers focus on representing the machine complexity as far as needed, especially when integrating machine parameters and flexible parts. The matrix based machine tool representation supported the determination of both the appropriated system boundary and the level of detail in preprocessing the fMBS. Besides it also supported the identification of particular parameters, which were primarily considered less important to sufficiently represent the structural behavior of the machine tool. The matrix based representation provided the base for this information to become worthy. Besides the matrix based representation also measurements of operation induced structural oscillations, and physical experiments supported the validation of the fMBS. Concurrent model validation enables to mature the fMBS simulation model further. In order to provide a means to evaluate the cause of tool deflection a properly validated fMBS is needed. Currently the fMBS represents the deflection of the tool identified by indirectly by measuring crank shafts, but sensitivity analysis is still been carried out. In modeling the iterative approach is quite time consuming and it becomes difficult to determine when the model is completely validated. Validation of the model takes place quite objectively by integrating the model developer and the user systematically.

6 CONCLUSIONS

The exemplified case study has proven that the suggested matrix based product representation could successfully support preprocessing and validation of a fMBS. Supported by the method specifications and machine parameters are identified to be integrated in the fMBS to represent structural bending induced by moved mass. Applying the suggested approach of matrix based product representation enables a holistic view of the system regarding component structure, functional modeling, and product behavior to support both preprocessing and validation of the simulation model. The significance of the suggested matrix based product representation strongly interrelates with the level of detail gained in each domain.

The authors will detail the presented case study further more to deeply illustrate the method and will apply the suggested approach to different products to enrich the application areas. Another task will be to evaluate the transfer of the suggested matrix based product representation to other simulation methods in CAE.

REFERENCES

(VDA), V. d. A. e. V., Ed. (1999). *Sicherung der Qualität vor Serieneinsatz*. Qualitätsmanagement in der Automobilindustrie. Frankfurt am Main, Henrich Druck+Medien.

Ariyo, O. O., C. M. Eckert, et al. (2006). On the use of functions, behavior and structural relations as cues for engineering change prediction. *International Design Conference - Design 2006*. Dubrovnik, Croatia.

Bender, K., Ed. (2005). *Embedded Systems - qualitätsorientierte Entwicklung: Qualitätssicherung bei Embedded Software*. Berlin, Springer Verlag.

Biedermann, W. and U. Lindemann (2008). *Cycles in the Multiple-Domain Matrix – Interpretation and Applications*. 10th International DSM Conference, Stockholm, Hansa.

Browning, T. R. (2001). *Applying the Design Structure Matrix to System Decomposition and Integration Problems: A Review and New Directions*. IEEE Transactions on Engineering Management.

Lindemann, U. (2007). *Methodische Entwicklung technischer Produkte*. Berlin, Spriger-Verlag.

Maurer, M. (2007). *Structural Awareness in Complex Product Design* Munich, Dr. Hut.

Musselman, K. J. (1994). *Guidelines for simulation project success*. Simulation Conference Proceedings, 1994. Winter.

Pahl, G. and W. Beitz (1995). *Engineering Design: A Systematic Approach*. London, Springer Verlag.

Robertson, N. and T. Perera (2002). "Automated data collection for simulation?" *Simulation Practice and Theory* 9(6-8): 349-364.

Robinson, S. and V. Bhatia (1995). *Secrets of successful simulation projects*. Simulation Conference Proceedings, 1995. Winter.

Sargent, R. G. (2004). *Validation and verification of simulation models*. Simulation Conference, 2004. Proceedings of the 2004 Winter.

Schiehlen, W. (1997). "Multibody System Dynamics: Roots and Perspectives." *Multibody System Dynamics* 1(2): 149-188.

Schwertassek, R., O. Wallrapp, et al. (1999). "Flexible Multibody Simulation and Choice of Shape Functions." *Nonlinear Dynamics* 20(4): 361-380.

Shabana, A. A. (1997). "Flexible Multibody Dynamics: Review of Past and Recent Developments." *Multibody System Dynamics* 1(2): 189-222.

VDI, Ed. (2005). *Classification and evaluation of description methods in automation and control technology, VDI/VDE 3681* Düsseldorf.

DYNAMICAL CLUSTERING TECHNIQUE TO ESTIMATE THE PROBABILITY OF THE FAILURE OCCURRENCE OF PROCESS SUBJECTED TO SLOW DEGRADATION

M. Traore, E. Duviella and S. Lecoeuche
Departement Informatique et Automatique, Ecole des Mines de Douai, France
{*traore, duviella, lecoeuche*}*@ensm-douai.fr*

Keywords: Supervision, Pattern Recognition, Non-stationary data, AUDyC.

Abstract: In this paper, we propose a supervision method which aims at determining pertinent indicators to optimize predictive maintenance strategies. The supervision method, based on the AUto-adaptative and Dynamical Clustering technique (AUDyC), consists in classifying in real time measured data into classes representative of the operating modes of the process. This technique also allows the detection and the tracking of the slow evolutions of the process modes. Based on the AUDyC technique, a method is proposed to estimate the probabilities of the failure occurence of components in real time. This method is illustrated on the real case of a temperature controller.

1 INTRODUCTION

Maintenance strategies consist in improving the safety and the reliability of industrial processes, taking into account their characteristics and the cost of maintenance plans (Grall et al., 2002). Amongst the three principal types of maintenance strategies which are proposed in the literature (Muller et al., 2004), *i.e.* the corrective, the preventive and the predictive maintenance strategies, the predictive maintenance allows the anticipation of failures and the optimal selection of maintenance actions, by the estimation in real time of the current state of the process components. This strategy is generally based on supervision methods and the estimation of the failure occurrence probabilities of the components of the process. The initial selection of the components which are essential to supervise, is performed by a dysfunctional analysis of the failure modes and their effects (FMEA: Failure Mode and Effects Analysis). Then, the interactions between each component are modelled by a Fault Tree formalism (Lassagne, 2000), (Vesely et al., 1981). Finally, the Fault Tree can be quantified by using the concept of Probability Functions by Episode (PFE) which allow the association of a probability of occurrence function to each component. In (Desinde et al., 2006), the PFE of the components are supposed to be known *a priori* and resulted from factory tests of feedback methods. We propose in this paper a supervision method allowing of determine the PFE in real time. The supervision methods based on mathematical models of the process can not be used for complex processes or when no physical model is available. In these cases, supervision approaches which consist in extracting relevant and sensitive informations of the component state by using directly the sensor signals, are more efficient. These supervision methods gather Pattern Recognition (PR) techniques which involve the state of a component by the analysis of evolutive data. The PR techniques include for exemple dynamic classification algorithms for evolutive data defined in (Lurette and Lecoeuche, 2003), which are dedicated to associate a state to one of the several operating modes of the system. FMMC (Min-Max Fuzzy Clustering) (Mouchawed and Billaudel, 2002) or AUDyC (AUto-adaptive and Dynamical Clustering) techniques allow the detection and the tracking of fast and slow evolutions of non-stationary data, and the diagnosis of the current state of the process. AUDyC approach is specially adapted to the supervision of slow evolutions or drifts due for exemple to ageing phenomenon (Lecoeuche et al., 2004). It allows the classification of the observed data according to classes which correspond to the operating modes of the process, *i.e.* normal, current and default modes. Estimation techniques of the distances between the several classes have been proposed to quantify the positioning of each classe. In this context, the main difficulty is to estimate the probabilities of the failure occurrence of components according to the dynamic data classification, and finally to provide indicators allows the improvement of predictive maintenance strategies.

In this paper, we consider processes characterised by slow evolutions of their operating modes. We propose to use AUDyC technique to supervise the components of the process, to estimate the probability of occurrence of each component of the process. The problematic addressed in this paper is detailled in the section 2. The supervision method by AUDyC is presented in Section 3. In Section 4, we present the methods proposed to estimate the probability of the failure occurrence of components of the process. Finally, the proposed methods are applied to a temperature controller.

2 PROBLEMATIC

Considering processes subjected to slow drifts of their current mode towards default modes, we propose a method which aims at determining indicators like probabilities of the failure occurrence of components. These indicators can be used to optimize predictive maintenance plans. The first step of maintenance strategy consists in a FMEA of the process to determine the corresponding Fault Tree, to specify the elementary component and the interactions between each component. The FMEA of the process leads also to the determination of components which are necessary to be supervised. The Fault Tree is quantified by using Probability Functions by Episode (PFE) (*see* Figure 1), where $PFE(E_x)$ which denotes the PFE of the event E_x is expressed by relation (1). The PFE of events associated to elementary components, *i.e.* E_1 to E_4, are used to compute the PFE of others events, E_5 and E_6.

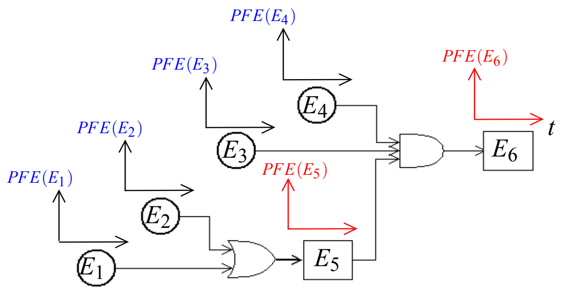


Figure 1: Fault Tree and PFE associated to components.

$$PFE(E_j) = ((p_1^{E_j}, t_1), \cdots, (p_n^{E_j}, t_n)) \qquad (1)$$

$\forall\, t_i \quad p_i^{E_j} = p^{E_j}(t_i)$, where $p^{E_j}(t_i)$ is the failure occurrence probability of the event E_j of the component j at time t_i.

The components which have to be monotored being known, it is necessary to select the variables which are characteristics of the component state. Three states are considered: normal, current and default modes. The goals of the dynamic data classification technique is to classify the measured data according to normal, current or default classes in real time. The estimation of characteristics of the current class leads to the detection and the tracking of drifts. The normal and default classes are known *a priori*, and are represented in the data representation space (*see* Figure 2). The slow drift of an operating mode has for effect of gradual change of the data from the normal class to the default class. The goal is to characterize in term of PFE the drift of an operating mode from the normal mode to the default mode. For that, we use AUDyC technique as modelling technique and estimation techniques of the distances between classes, as Euclidean and Kullback-Leibler distances. The AUDyC technique and the estimation methods of the distances are presented in the next section.

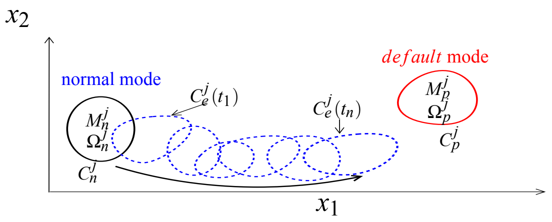


Figure 2: Slow drift operating.

3 CURRENT CLASS MODELLING BY AUDYC TECHNIQUE

The supervision method based on the AUDyC technique aims at monitoring each component of the process and at determining their mode. An operating mode is represented by a Gaussian class C_k^j which is characterized by a center M_k^j and a matrix of covariance Ω_k^j. These parameters are estimated in real time according to the observed data contained into the observation vector which is denoted $X^i = [x_1^i, x_2^i, \cdots, x_d^i]$ in the d space dimensions. The AUDyC algorithm consists in updating the class parameters recursively on a sliding window of width N_{fen} taking into account the cardinality of the class C_k^j, *i.e.* $Card(C_k^j)$. The steps of the algorithm, detailed in (Lecoeuche et al., 2004), are presented thereafter:

- If $Card(C_k^j)$=nb ¡ N_{fen}: Add information

$$M_k^j(t) = M_k^j(t-1) + \frac{1}{nb+1}(X(t) - M_k^j(t-1))$$
$$\Omega_k^j(t) = \frac{nb-1}{nb}\Omega_k^j(t-1)+$$
$$\frac{1}{nb+1}(X(t) - M_k^j(t-1))^\top (X(t) - M_k^j(t-1)) \tag{2}$$

- If nb $\geq N_{fen}$: Add and remove information

$$M_k^j(t) = M_k^j(t-1) + \frac{1}{N_{fen}}(\delta X^+ - \delta X^-)$$
$$\Omega_k^j(t) = \Omega_k^j(t-1)+$$
$$\Delta X \begin{bmatrix} \frac{1}{N_{fen}} & \frac{1}{N_{fen}(N_{fen}-1)} \\ \frac{1}{N_{fen}(N_{fen}-1)} & -\frac{(N_{fen}+1)}{N_{fen}(N_{fen}-1)} \end{bmatrix} \Delta X^\top \tag{3}$$

where:

$$\begin{cases} \delta X^+ = X^{new} - M_k^j(t-1), \\ \delta X^- = X^{old} - M_k^j(t-1), \\ \Delta X = [\delta X^+ \quad \delta X^-]. \end{cases} \tag{4}$$

with $M_k^j(t)$ and $\Omega_k^j(t)$ respectively center and covariance matrix of the class C_k^j at time t, N_{fen} the width of the sliding window, $X^{new} = X(t)$, X^{old} the old data in the set affected to C_k^j.

Then, the distances between the normal, current and default classes can be computed according to the center and the covariance matrix of each class. The Euclidean distance corresponds to the distance between the center of two classes:

$$d_{Eu} = (M_1^j - M_2^j)^\top (M_1^j - M_2^j) \tag{5}$$

where M_1^j and M_2^j are the centers of the classes C_1^j and C_2^j respectively.
The Kullback-Leibler distance corresponds to the distance between two classes taking into account their shape, *i.e.* the covariance matrices, (Kullback and Leibler, 1951). In the general case (Anguita and Hernando, 2004), the distance between the classes C_1^j and C_2^j is expressed by:

$$d_{kl}(C_1^j, C_2^j) = \frac{1}{2}(M_1^j - M_2^j)^\top (\Omega_1^{-1} + \Omega_2^{-1})(M_1^j - M_2^j)$$
$$+ \frac{1}{2} trace(\Omega_1^{-1}\Omega_2 + \Omega_1\Omega_2^{-1}) - d. \tag{6}$$

where d is the dimension of the data representation space, $\Omega_1 = \Omega_1^j$ and $\Omega_2 = \Omega_1^j$ are the covariance matrices of the classes C_1^j and C_2^j. The second term of d_{kl}, *i.e.* $trace()$, is specifically impacted by the shape and the orientation of the classes.
Finally, the distances between the several modes are used to estimate the probabilities of the failure occurrence of each component, as detailed in the next section.

4 ESTIMATION OF FAILURE OCCURRENCE PROBABILITIES

The probability of the failure occurrence, denoted $p^{E_j}(t)$, is defined as the PFE of an elementary component, and is considered as an indicator of the deterioration of this component. It is estimated according to the distance covered by the current class towards the default class, $\alpha(t)$, due to slow drifts:

$$p^{E_j}(t) = 1 - \alpha(t) \tag{7}$$

with:

$$\alpha(t) = \frac{distance(C_p^j, C_e^j(t))}{distance(C_n^j, C_p^j)} \tag{8}$$

where C_n^j, C_p^j, and C_e^j are the normal, default and current classes. The distance between two classes is computed according to the Euclidean (5) or the Kullback-Leibler (6) methods. It is assumed that $0 \leq \alpha(t) \leq 1$.

- **Estimation of $p^{E_j}(t)$ based on Euclidean Distance**

The percentage of distance $\alpha_{Eu}(t)$ which is estimated according to the Euclidean distance (5), is used to determine the probability $p_{Eu}^{E_j}(t)$ according to the relation (7). The example shown in Figure 3 is considered to illustrate this method. Three classes for component j are represented: normal C_n^j, current C_e^j, and default C_p^j classes characterized by $(M_n^j,\ \Omega_n^j)$, $(M_e^j,\ \Omega_e^j)$, and $(M_p^j,\ \Omega_p^j)$, respectively.
The percentage of distance $\alpha_{Eu}(t)$ at each time t is given by:

$$\alpha_{Eu}(t) = \frac{d_{Eu}(M_p^j, M_e'^j(t))}{d_{Eu}(M_n^j, M_p^j)} \tag{9}$$

where the distances d_{Eu} are expressed by relation (5), and $M_e'^j$ is the orthogonal projection of the center M_e^j on the segment $[M_n^j\ M_p^j]$. The distance $d_{Eu}(M_n^j, M_p^j) =$

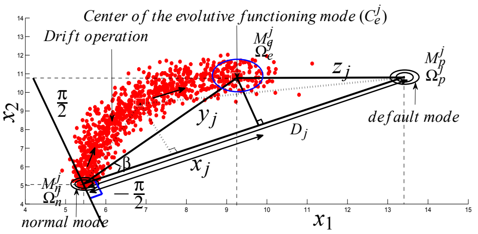


Figure 3: Evolution of a class from the normal class to the default class.

D_j is constant (10). The distance $d_{Eu}(M_n^j, M_e^{\prime j}(t)) = x_j$ is determined according to relation (11) from the triangle formed by the centers of the classes (*see* Figure 3). It is assumed that the current class can only evolve towards the default class. Consequently, the angle β is always include between $-\frac{\pi}{2} < \beta < \frac{\pi}{2}$, and the orthogonal projection of the center M_e^j is always included in the segment $[M_n^j\, M_p^j]$.

$$D_j = \sqrt{(M_n^j - M_p^j)^\top (M_n^j - M_p^j)} \tag{10}$$

$$x_j(t) = \frac{1}{2}\left[D_j + \frac{y_j^2(t) - z_j^2(t)}{D_j}\right] \tag{11}$$

with:

$$y_j(t) = \sqrt{(M_n^j - M_e^j(t))^\top (M_n^j - M_e^j(t))} \tag{12}$$

$$z_j(t) = \sqrt{(M_p^j - M_e^j(t))^\top (M_p^j - M_e^j(t))} \tag{13}$$

(14)

- **Estimation of $p^{E_j}(t)$ based on Kullback-Leibler Distance**

The Kullback-Leibler distance is used to estimate the percentage of distance $\alpha_{Kl}(t)$ and then to determine the probability $p_{Kl}^{E_j}(t)$, according to the relation (7). The percentage of distance $\alpha_{Kl}(t)$ at each time t is given by:

$$\alpha_{kl}(t) = \frac{d_{kl}(C_p^j, C_e^j(t))}{d_{kl}(C_n^j, C_p^j)} \tag{15}$$

where the distances d_{kl} are expressed by relation (6). The Kullback-Leibler distance between the class C_n^j and the class C_p^j is constant.

The percentage of distance $\alpha_{Kl}(t)$ is computed only when the current class C_e^j evolves towards the default class C_p^j. A criterion T_c^d is defined to verify this condition (16). Thus, $\alpha_{Kl}(t)$ is computed if and only if the criterion T_c^d is strictly negative.

$$\begin{cases} T_c^d = \dfrac{1}{N_{fen} - 1} \displaystyle\sum_{t=2}^{N_{fen}} sign(\Delta t), \\ \Delta(t) = d_{kl}(C_p^j, C_e^j(t)) - d_{kl}(C_p^j, C_e^j(t-1)) \end{cases} \tag{16}$$

- **Interpretation of $p^{E_j}(t)$ Computed According to Euclidean and Kullback-Liebler Distances**

The probabilities $p_{Eu}^{E_j}(t)$ and $p_{Kl}^{E_j}(t)$ are computed according to the Euclidean or Kullback-Liebler distances $\alpha_{Eu}(t)$ and $\alpha_{Kl}(t)$. To interprete and verify the pertinence of these indicators and thus the proposed methods, a scenario which consists in four current classes C_1^j to C_4^j which evolve to the normal class C_n^j towards the default class C_p^j, is considered and depicted in Figure 4. The classes C_1^j, C_2^j and C_3^j have the same centers but their matrices of covariance are different. The class C_4^j is characterized by different center and covariance matrice. The probabilities $p_{Eu}^{E_j}(t)$ and $p_{Kl}^{E_j}(t)$ are computed for the four classes. The results are given in Table 1.

Table 1: Probabilities computed for the classes.

	C_1^j	C_2^j	C_3^j	C_4^j
$p_{Eu}^{E_j}$	0,50	0,50	0,50	0,56
$p_{Kl}^{E_j}$	0,36	0,50	0,46	0,46

The Euclidean distance leads to the estimation of a same pourcentage $p_{Eu}^{E_j}$ for classes C_1^j, C_2^j and C_3^j, and to a pourcentage more important for the class C_4^j. Indeed, the center of the class C_4^j is nearest to the default class than the others classes C_p^j. This distance is easily interpretable but it does not take into account the shape and the orientation of the classes.

The Kullback-Liebler distance leads to the estimation of pourcentages $p_{Kl}^{E_j}$ different for the classes C_1^j, C_2^j and C_3^j. Although, the covariance matrix of the class C_1^j is smaller than the covariance matrix of the class C_3^j, the difference between the obtained pourcentages seems to be too important, and these indicators are not directly interpretable as the probabilities of the failure occurrence. Moreover, the pourcentages $p_{Kl}^{E_j}$ of the classes C_3^j and C_4^j are identical although the class C_4^j is nearest of the default class (*see* Figure 4). Finally, the Kullback-Liebler distance allows to take into account the shape and the orientation of the classes, but it is not directly usable for the estimation of the probabilities of the failure occurrence.

- **New Estimation Method of $p^{E_j}(t)$**

A new estimation method of the probability $p^{E_j}(t)$ is proposed to provide pertinent indicators which take into account in priority the position of the classes, but also, the remoteness, enlarging and rotation of these classes. It consists in a weigthed combination of $p_{Eu}^{E_j}(t)$ computed according to Euclidean distance and

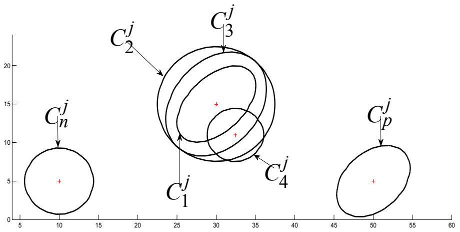


Figure 4: Scenario of evolution of classes from a normal class towards a default class.

p_ε which is computed according to the second term of the Kullback-Leibler distance (6). The probability $p^{E_j}(t)$ is expressed as:

$$p^{E_j}(t) = p_{Eu}^{E_j}(t) + \lambda\, p_\varepsilon \tag{17}$$

where λ ($0 < \lambda < 1$) is a weight coefficient. The parameter p_ε is function of the covariances matrices of the normal, default and current classes:

$$\begin{aligned} p_\varepsilon &= \frac{T_1}{T_1 + T_2} \\ T_1 &= trace(\Omega_e \Omega_p^{-1} + \Omega_e^{-1} \Omega_p) \\ T_2 &= trace(\Omega_p \Omega_n^{-1} + \Omega_p^{-1} \Omega_n) \end{aligned} \tag{18}$$

The coefficient λ is tuned in order to take into account the covariance matrices in the estimation of $p^{E_j}(t)$ without however obtaining too important differences between the distances from the classes. In Table 2, we presente the occurrence probabilities computed by relation (17) according to $\lambda = 1/10$.

Table 2: Failure occurrence probabilities.

	C_1^j	C_2^j	C_3^j	C_4^j
$p^{E_j}(t)$	0,53	0,55	0,54	0,60

If the value of λ is too small, the shape of the class is not taken into account, and that leads at considering only the Euclidean distance. If the value of λ is too big, the shape of the class has too much influence on the estimation of $p^{E_j}(t)$, and that leads to the same problem of interpretation than the distance of Kullback-Leibler. The proposed method is applied on a real scenario in the next section.

5 APPLICATION

A temperature controller is a process which is used to control the temperature of a client system. It is composed of an electric heater, a pump, a heat exchanger and a filter (*see* Figure 5). The components of this process are subject to failures related to slow degradations due to scaling and fouling essentially. If these failures are not taken into account early enough, they can cause the stop of the process.

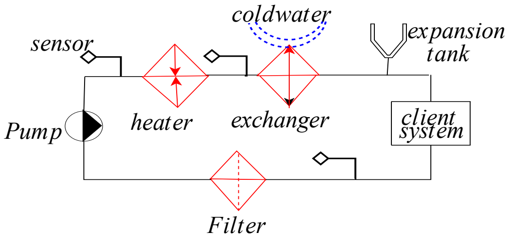


Figure 5: Thermo-regulator components.

The first step is the FMEA of the temperature controller which allows the determination of the Fault Tree of the process (*see* Figure 6). The Fault Tree is composed of three basic events associated to each component and a top event which correspond to the no temperature control. The basic events are:

- Failure of the heater (E_1)
- Failure of the exchanger (E_2)
- Failure of the filter (E_3)

the top event is:

- No temperature control (E_4)

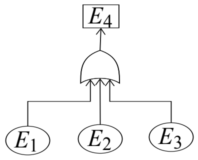


Figure 6: Fault Tree of the temperature controller.

The temperature controller is equipped by sensors located at the input and output of each component. These sensors measure the pressure of the fluid. An observation vector is done by $X_1 = (x_1, x_2, x_3)^\top$ where the three indicators are determined according to the measurements:

$$x_1 = \frac{P_{input\ heater} - P_{output\ heater}}{\Delta P_{pump}} \tag{19}$$

$$x_2 = \frac{P_{input\ exchanger} - P_{output\ exchanger}}{\Delta P_{pump}} \tag{20}$$

$$x_3 = \frac{P_{input\ filter} - P_{output\ pump}}{\Delta P_{pump}} \tag{21}$$

where x_1, x_2, x_3 are indicators to monitor the heater, the exchanger and the filter respectively.

The AUDyC technique allows the monitoring of elementary components of the temperature controller.

The estimation method of the occurrence probabilities, with a weight coefficient tuned as $\lambda = 1/20$, $\lambda = 1/50$, is used to estimate in real time the $p^{E_j}(t)_{j=1,2,3}$ (17) of each elementary component, and finally the PFE of top event (E_4) by propagation the basic events.

where the events E_1, E_2 and E_3 are independents. Thus, the PFE of the event E_4 is expressed as:

$$PFE(E_4) = ((p^{E_4}(t_1), t_1), \cdots, (p^{E_4}(t_n), t_n)) \qquad (22)$$

In the real scenario considered, the components of the temperature controller are subjected to drifts as depicted in Figure 7. The $PFE(E_4)$ determined according to the relation (22) are displayed in Figure 8. On this real scenario,the tune $\lambda = 1/20$ leads to a too important influence of p_ε, whereas $\lambda = 1/50$ presents a good compromise. La figure 8.a montre l'influence de la forme de la classe alors que la figure 8.b l'influence de la forme de la classe est moins important.

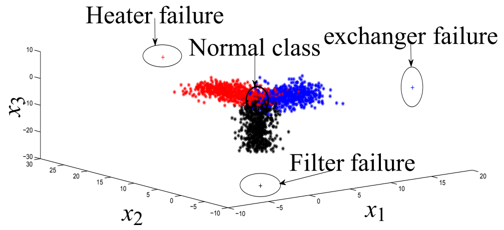


Figure 7: Drifts of operating of the components.

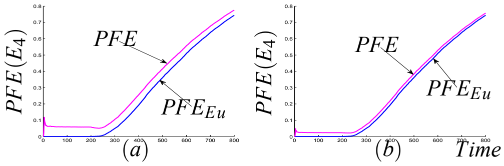


Figure 8: $PFE(E_4)$ according to (a) $\lambda = 1/20$, (b) $\lambda = 1/50$.

6 CONCLUSIONS

The supervision method proposed in this paper allows the estimation of the probability of failure occurrence of processes in real time. The dynamic clustering method is used to track the evolution of operating modes of processes by determining the characteristics of each class (center and covariance matrix).

The center and the covariance matrix being adapted by AUDyC, the Euclidean distance and trace of the covariance matrices are used to estimate the probability of the failure occurrence. The Euclidean distance does not allow to take into account the shape and the orientation of the class, and the Kullback-Leibler distance, are not easily interpretable. Then, a new method which is based on the weight combination between the probabilities estimated with the Euclidean distance and with the trace of the covariance matrices, is proposed and illustrated on real case. In futur works, we will propose a prognosis strategy based on this method to forecast the occurrence probability of events, and a step to tune the weight coefficients of the proposed method. The goal is to determine indicators to improve the predictive maintenance of processes. This will be implemented for predictive maintenance of the temperature controller and of measure the apport of the proposed methods.

REFERENCES

Anguita, J. and Hernando, J. (2004). Inter-phone and inter-word distances for confusability prediction in speech recognition. *Congreso de la Sociedad Espaola para el Procesamiento del Lenguaje Natural*, (33):33–40.

Desinde, M., Flaus, J. M., and Ploix, S. (2006). Tool and methodology for online risk assessement of process. In *Lambda-Mu 15 /Lille*.

Grall, A., Berenguer, C., and Dieulle, L. (2002). A condition-based maintenance policy for stochastically deteriorating systems. *Reliability Engineering and System Safety*, 76(2):167–180.

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *Annal of Mathematical Statistics*,22:79-86.

Lassagne, M. (2000). Applying a decision-analysis-based method to the evaluation of potential risk-reducing measures : The case of a floating production storage and offloading unit in the gulf of mexico. *SPE annual technical conference, Dallas TX , USA*.

Lecoeuche, S., Lurette, C., and Lalot, S. (2004). New supervision architecture based on on-line modelling of non-stationary data. *Neural Computing and Applications Journal*, 13:323–338.

Lurette, C. and Lecoeuche, S. (2003). Unsupervised and auto-adaptive neural architecture for on-line monitoring. application to a hydraulic process. *Engineering Applications of Artificial Intelligence*, 16:441–451.

Mouchawed, S. M. and Billaudel, P. (2002). Influence of the choice of histogram parameters at fuzzy pattern matching performance, int. journal of wseas transactions on system. *WSEAS Transactions on Systems*, 1:260–266.

Muller, A., Suhner, M.-C., Iung, B., and Morel, G. (2004). Prognosis-based maintenance decision-making for industrial process performance optimisation. *In 7th IFAC Symposium on Cost Oriented Automation (COA2004). Gatineau/Ottawa Canada*.

Vesely, W. E., Goldberg, F. F., Robert, N. H., and Haasl, D. F. (1981). *Fault Tree Handbook*. US nuclear Regulatory Commission, Washington D.C., USA.

POSTERS

SELF-SIMILARITY MEASURMENT USING PERCENTAGE OF ANGLE SIMILARITY ON CORRELATIONS OF FACE OBJECTS

Darun Kesrarat and Paitoon Porntrakoon
Autonomous System Research Laboratory, Faculty of Science and Technology, Assumption University
Ramkumhaeng 24, Huamark, Bangkok, Thailand
{darun, paitoon}@scitech.au.edu

Keywords: Self- Similarity, Face objects, Correlations.

Abstract: A 2D face image can be used to search the self-similar images in the criminal database. This self-similar search can assist the human user to make the final decision among the retrieved images. In previous self-similar search, a 2D face image comprises of objects and object correlations. The attribute values of objects and their correlations are measured and stored in the face image database. The similarity percentage is specified to retrieve the self-similar images from the database. The problem of previous self-similar search is that the percentage of the angle differentiation among the objects in different part is different although their angle differentiation is exactly the same. The proposed model is introduced to improve the stability of the similarity percentage by reducing the number of face objects, object correlations, and the degree calculation. After testing over 100 samples, the proposed method illustrated that the stability of similarity percentage is improved especially for the left side objects of the face image.

1 INTRODUCTION

The face image is two dimensional, vertical and horizontal. For each image, there are 10 objects – Face, Right Eyebrow, Left Eyebrow, Right Eye, Left Eye, Right Ear, Left Ear, Nose, Mouth, and Scar that are identified and the size from the center toward the 0, 90, 180, and 270 degrees of each object are recorded in the database. The Face object is used as the reference object. There are 9 object correlations – Face against Right Eyebrow, Face against Left Eyebrow, Face against Right Eye, Face against Left Eye, Face against Right Ear, Face against Left Ear, Face against Nose, Face against Mouth, and Face against Scar – in which their distance and angle toward the Face object are recorded in the database as well. The self-similar images in which all the attribute values of objects and object correlations are not exceed the specified similarity percentage will be retrieved from the database by using the following formula (P. Porntrakoon, 1999; V. Srisarkun, 2001 & 2002).

$$angle\_similarity\_percentage \geq \frac{|q_i - r_i|}{\max(q, r)} \times 100 \qquad (1)$$

where q is an attribute value of the object of the key image and r is an attribute value of the object of stored image.

It is obvious that the degree calculation of each object – in different part of the face – toward the reference object is unstable. Therefore the percentage of the angle differentiation among the objects in different part will be different although their angle differentiation is exactly the same – e.g., 2 degrees on the face.

The proposed mothod reduces the number of objects to 8 objects, reduces number of object correlations to 7 correlations, and introduces the new calculations of the object correlations. The proposed method presents a more stable ratio of angle similarity among objects in different part of the face although their angle differentiation is exactly the same. Moreover, the proposed method requires less attributes to represent the content of the face image. The attribute number is adequate to retrieve the similar face images from the database. The space required to store the attribute values is less and the search time is much improved.

2 PROPOSED METHOD

2.1 Face Image Conversion

The face image is segmented into closed contours corresponding to the dominant image objects. Each object contains its object correlation and attributes (I. Kapouleas, 1990; S. Dellepiane, 1992; A.V. Ramen, 1993). In the proposed method, 8 objects – Nose, Right Eyebrow, Left Eyebrow, Right Eye, Left Eye, Right Ear, Left Ear, and Mouth are detected by specifying the top, bottom, leftmost, and rightmost positions of each object. Nose will be used as the reference object that has the correlation with the remaining objects.

Therefore, a face image has 7 object correlations and each correlation (Right Eyebrow versus Nose, Left Eyebrow versus Nose, Right Eye versus Nose, Left Eye versus Nose, Right Ear versus Nose, Left Ear versus Nose, Mouth versus Nose).

Each correlation has angular direction and distance to the center of the reference object. The distance is measured in pixel while the direction is measured in degrees.

In this paper, the object and the object correlation are estimated prior to the storing. The attributes include size, distance, and angle.

2.2 Specify the Positions of Objects on the Face Image and Calculate the Object's Center Coordinate (x, y)

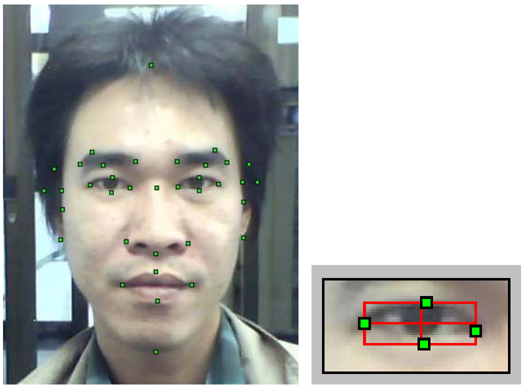


Figure 1: Specified positions of the top, bottom, leftmost, and rightmost of each object.

For each object, the coordinate (x, y) position of the top, bottom, leftmost, and rightmost are specified as shown in Figure 1. Then the coordinate (x, y) of each object's center is calculated as follows:

$$OCx = (OLx + ORx) / 2 \quad (2)$$

$$OCy = (OTx + OWx) / 2 \quad (3)$$

where *OCx,y* is the center, *OLx,y* is the leftmost, *ORx,y* is the rightmost, *OTx,y* is the top, and *OWx,y* is the bottom coordinate (x, y) of the object.

2.3 Calculate the Distance of Each Face Object based on the Reference Object

After the boundary and the center of each object are identified, the distances and angles from the center of reference object toward the remaining objects are calculated as follows:

$$\text{Distance} = (OCy - Oy) / \ TAN((OCy - Oy) / (OCx - Ox))^{-1} \quad (4)$$

Where *OCx* and *OCy* are the center coordinate (x, y) of the reference object, *Ox* and *Oy* are the center coordinate (x, y) of the correlated object.

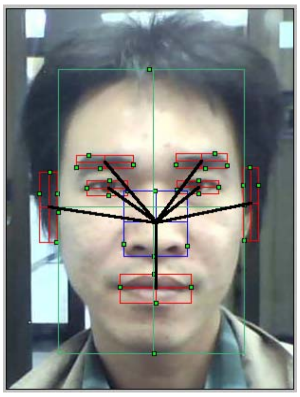


Figure 2: Distance of each object based on the reference object.

2.4 Calculate the Angle of Each Face Object based on the Reference Object

In the angle calculation, the direction of each object except the nose is measured – in degrees – from its center coordinated (x, y) to the center coordinated (x, y) of the reference object (Nose). This model considers the widest range of each object to the center object based on the location of that object toward the reference object as shown in Figure 3.

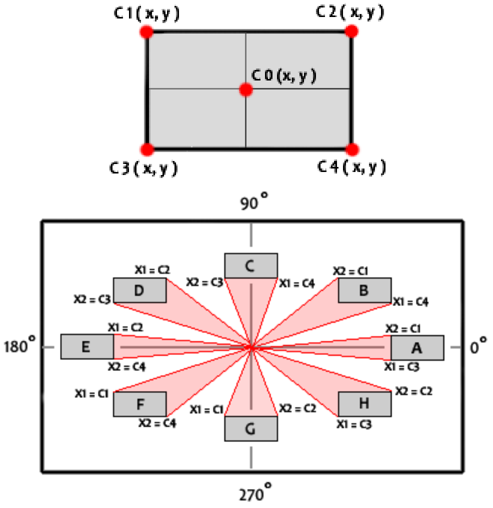


Figure 3: Distance of each object based on the reference object.

The object in different location from the center object will use the different coordinate positions (x, y) to calculate the Minimum and Maximum degrees toward the reference object.

Then, the widest position of each object toward the reference object is used to calculate the maximum and minimum degrees as shown in Figure 4 and is calculated as follows:

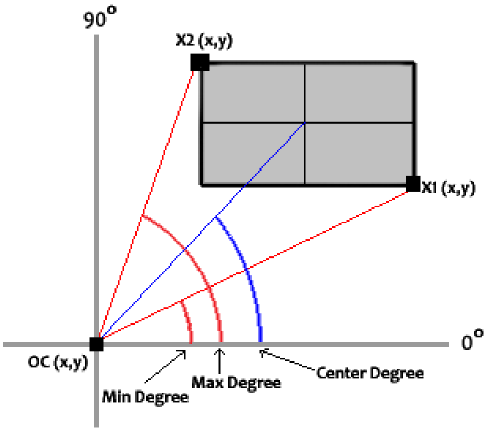


Figure 4: Maximum and Minimum degrees of each object toward the reference object.

$$\text{Min} = \text{ATAN}((OCy - X1y) / (OCx - X1x)) \quad (5)$$

$$\text{Max} = \text{ATAN}((OCy - X2y) / (OCx - X2x)) \quad (6)$$

Where X1x, X1y, X2x, and X2y are the widest positions – coordinate (x, y) – of the correlated object.

3 EXPERIMENTS

The experiments were performed to test the stability of similarity percentage by testing over 100 samples of front face image (640*480 resolution) which consider the object correlation one by one (Right Eyebrow, Left Eyebrow, Right Eye, Left Eye, Right Ear, Left Ear, and Mouth) toward the reference object (Nose). Then compare the similarity percentage of its objects by simulating that the object is compared with the same object size when it is simulated to locate at different degree (range from ± 1-10 degrees) from its own original position to prove that the percent of the similarity from the proposed method is more stable than the old method “A Model for Similarity Searching in 2D Face Image Data” (P. Porntrakoon, 1999), “A model for Self-Similar Searching in Face Image Data Processing” (V. Srisarkun, 2001), “Self-Similar Searching in Image Database for crime Investigation” (V. Srisarkun, 2001), “A model for Self-Similar Search in Image Database with Scar” (V. Srisarkun, 2002), and “Face Recognition Using a Similarity-based Distance Measure for Image Database” (V. Srisarkun, 2002).

Then the processes used to perform the experiments are as follows.

3.1 Resize the Position and Proportion of the Face Objects

To avoid the problem of the different object size caused by the distance of the captured images, the position and proportion of the objects are resized by adjusting the width of the reference object (Nose) in the captured images to have the same width. Then recalculate the top, bottom, leftmost, rightmost, and center coordinate (x,y) positions of each object based on the new proportion of the reference object as follows:

$$NOx = (iw + (100 / ((NLx - NRx) / iw * 100) * (dw - (NLx - NRx)))) / iw * Ox \quad (7)$$

$$NOy = (iw + (100 / ((NLx - NRx) / iw * 100) * (dw - (NLx - NRx)))) / iw * Oy \quad (8)$$

Where NOx is the new x coordinate after resizing, NOy is the new y coordinate after resizing, NLx is the leftmost, NRx is the rightmost x coordinate of the reference object (Nose), iw is the original image width in pixel, and dw is the default width value in pixel for resizing.

3.2 Calculate the Similarity Percentage of Angle of Each Object between Faces

According to the Maximum and Minimum degrees of each correlated object toward the reference object in each face image, this model will use the minimum and maximum degrees of the correlated object toward the reference object from the same object correlation number in different face images to calculate the similarity percentage of as shown in Figure 5 and is calculated as follows:

$$\text{Percent of similarity} = ((\text{Min1}, \text{Max1}) \cap (\text{Min2}, \text{Max2})) * 2 / ((\text{Max1} - \text{Min1}) + (\text{Max2} - \text{Min2})) *100 \qquad (9)$$

Where Min1, Max1, Min2, and Max2 are the Minimum and Maximum degrees of the same correlated object toward the reference object in different face images.

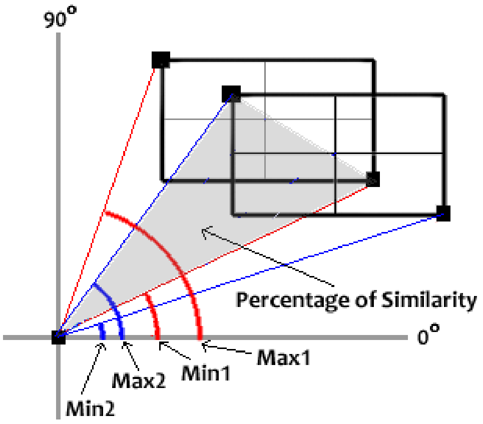


Figure 5: Percentage of Similarity.

4 EXPERIMENT RESULTS

From the experiment, we have summarized the results in average of percentage of angle similarity among object correlations on the face and standard deviations that compares the proposed method with the old one. The results are shown in Table 1, Figure 6 and Figure 7.

Table 1: Average Percentage of Angle Similarity result and standard deviation of the proposed method and the old method (P. Porntrakoon, 1999; V. Srisarkun, 2001&2002).

Degree Different	Average Similarity (%)		Average STD	
	Propose	Old	Propose	Old
1	97.7895	96.8704	0.43391	5.32916
2	95.5776	94.4426	0.86811	8.91565
3	93.3558	93.0225	1.30500	9.97539
4	91.1348	91.2665	1.74134	11.9929
5	88.9097	90.0068	2.17840	12.8165
6	86.6606	88.7411	2.62243	13.7332
7	84.2427	87.8864	3.15865	13.6906
8	81.9148	86.8308	3.65496	14.2223
9	79.6666	85.8518	4.08442	14.6272
10	77.6822	85.0452	4.38132	14.6733

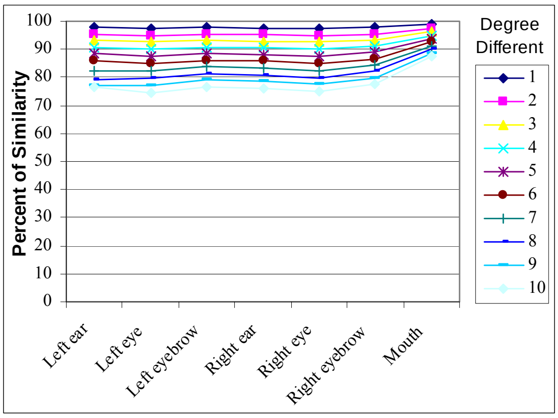


Figure 6: Average Similarity Percentage of the proposed method.

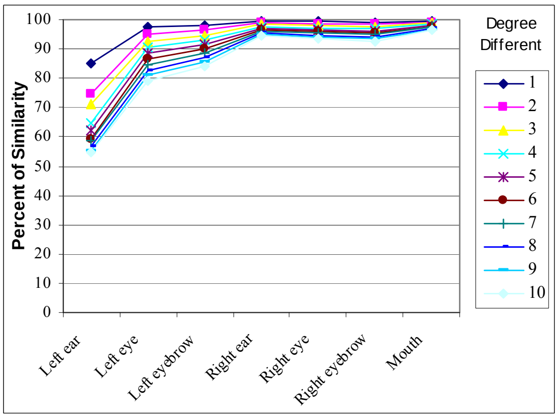


Figure 7: Average Similarity Percentage of the Old method (P. Porntrakoon, 1999; V. Srisarkun, 2001&2002).

We found that the proposed method provides more stable of the angle similarity percentage among object correlations compared to the old method that presents unstable (reference from the result of standard deviation value in Table 1) especially the left side object correlation that presents high deviation from the other correlated objects. Moreover, the old method still presents a little deviation when the actual different in degrees is increased which presents unstable result of the method.

5 CONCLUSIONS

In this paper, we proposed a method to handle approximate searching by image content in an image database. Older method, such as 2D string (S.-K. Change, 1987), giving binary answer is slow and not scaleable (S.-Y. Lee 1992). In addition, image content representation methods based on strings have been proven to be ineffective in capturing image content and may yield inaccurate retrieval (Petrakis, 1997). Our method allows querying the image database with the degree of similarity. And we do propose the method which considers the stability of the angle similarity percentage among object correlations. Older method, (P. Porntrakoon, 1999; V. Srisarkun, 2001&2002) also gave the unstable results.

The proposed method can reduce the instability in the angle similarity percentage for a better subsequent decision making process in similarity searching and reduce the number of object correlations which fasten the searching time.

6 FUTURE WORK

We plan to continue our research work by replace the proposed model which provided more stable result in percentage of angle similarity among object correlations over the full sequence reference from the old model (P. Porntrakoon, 1999; V. Srisarkun, 2001&2002) under the sample images of the same person which are taken at different time (approximately 2 -20 weeks). We believe that the front face photos that are taken from the same person at different time are not exactly the same .We will perform the experiments to prove the overall result of similarity between the future model and the old model.

ACKNOWLEDGEMENTS

We would like to thank Assumption University for this research funding.

REFERENCES

Petrakis, Euripides G.M., and Christos Faloutsos, 1997. In *IEEE trans. On Knowledge and data Engineering, "Similarity Searching in Medical Image Databases"*.

I. Kapouleas, 1990. In *Proc. 10th Int'l Conf. Pattern reconition, "Segmentation and Feature Extraction for Magnetic Resonance Brain Image Analysis"*.

S. Dellepiane, G. Venturi, and G. Vernazza, 1992. In *Pattern Recognition, "Model Generation and Model Matching of Real Images by Fuzzy Approach"*.

A.V. Ramen, S. Sarkar, and K.L. Boyer, 1993. In *CVGIP: Image Understanding, "Hypothesizing Structure in Edge-Focused Cerebral Magnetic Image Using Graph-Theoretic Cycle Enumeration"*.

S.-K. Change, Q.-Y. Shi, and C.-W. Yan, 1987. In *IEEE Trans. Pattern Analysis and Machine Intelligence, "Iconic Indexing by 2-D Strings"*.

S.-Y. Lee and F.-J. Hsu, 1992. In *Pattern Recognition, "Spatial Reasoning and Similarity Retrieval of Images Using 2D C-String Knowledge Representation"*.

P. Porntrakoon, and C. Jittawiriyanukoon, 1999. In *IASTED International Conference On Applied Modelling and Simulation, "A model for Similarity Searching in 2D Face Image Data"*.

V. Srisarkun, and C. Jittawiriyanukoon, 2001. In *the Sixth INFORMS Conference on Information System & Technology, "A model for Self-Similar Searching in Face Image Data Processing"*.

V. Srisarkun, J. Cooper and C. Jittawiriyanukoon, 2001. In *the Twentieth IASTED International Conference On Applied Informatics, "Self-Similar Searching in Image Database for crime Investigation"*.

V. Srisarkun, J. Cooper and C. Jittawiriyanukoon, 2002. In *ECIS International Conference On Applied Modelling and Simulation, "A model for Self-Similar Search in Image Database with Scar"*.

V. Srisarkun, J. Cooper and C. Jittawiriyanukoon, 2002. In *the Proceeding of the VIPromCom 2002-4th EURASIP-IEEE Region 8 International Symposium on Video Processing and Multimedia Communications, "Face Recognition Using a Similarity-based Distance Measure for Image Database"*.

COOPERATIVE TELEOPERATION TASK IN VIRTUAL ENVIRONMENT

Influence of Visual Aids and Oral Communication

Sehat Ullah, Samir Otmane and Malik Mallem
IBISC Laboratory, University of Evry, 40 rue de Pelvoux, Evry, France
{sehat.ullah, samir.otmane, malik.mallem}@ibisc.univ-evry.fr

Paul Richard
LISA Laboratory, University of Angers, 62 av Notre Dame du Lac, France
paul.richard@istia.univ-angers.fr

Keywords: CVEs, Cooperative teleoperation, Parallel robots, SPIDAR, Human performance, Multimodal feedback.

Abstract: Cooperative virtual environments, where users simultaneously manipulate objects, is one of the subfields of Collaborative virtual environments (CVEs). In this paper we simulate the use of two string based parallel robots in cooperative teleoperation task, Two users setting on separate machines connected through local network operate each robot. In addition, the article presents the use of sensory feedback (i.e shadow, arrows and oral communication) and investigates their effects on cooperation, presence and users performance. Ten volunteers subject had to cooperatively perform a peg-in-hole task. Results revealed that shadow has a significant effect on task execution while arrows and oral communication not only increase users performance but also enhance the sense of presence and awareness. Our investigations will help in the development of teleoperation systems for cooperative assembly, surgical training and rehabilitation systems.

1 INTRODUCTION

A CVE is a computer generated world that enables people in local/remote locations to interact with synthetic objects and representations of other participants within it. The applications of such environments are in military training, telepresence, collaborative design and engineering and entertainment. Interaction in CVE may take one of the following form (Otto et al., 2006): Asynchronous: It is the sequential manipulation of distinct or same attributes of an object, for example a person changes an object position, then another person paints it. Another example is, if a person moves an object to a place, then another person moves it further.
Synchronous: It is the concurrent manipulation of distinct or the same attributes of an object, for example a person is holding an object while another person is painting it, or when two or many people lift or displace a heavy object together.

In order to carry out a cooperative task efficiently, the participants need to feel the presence and actions of others and have means of communication with each other. The communication may be verbal or non verbal such as pointing to, looking at or even through gestures or facial expressions. We implement the VE designed for cooperative work in replicated architecture and seek solution to network load/latency and consistency in unique way. Similarly to make cooperative work easier and intuitive we augment the environment with audio and visual aids. Moreover we investigate the effect of these sensory feedback on user performance in a peg-in-hole task.

This section is followed by the related work, Section 3 describes the proposed system. Section 4 discusses the experiment and results analysis. Section 5 is dedicated to conclusion future work.

2 RELATED WORK

A lot of work has already been done in the field of CVE, for example MASSIVE provides a collaborative environment for teleconferencing (Greenhalgh and Benford, 1995). Most of this collaborative work is pertinent to the general software sketch and the underlying network architecture (Chastine et al., 2005; Shirmohammadi and Georganas, 2001). Basdogan et al. have investigated the role of force feedback in cooperative task. They connected two monitors and haptic devices to a single machine (Basdogan et al., 2001). Similarly, Eva-lotta et al. have reported the effect of force feedback over presence, awareness and task performance in a CVE. They connected two monitors and haptic devices to a single host (Sallnas et al., 2000). Other important works that support the cooperative manipulation of objects in a VE include (Jordan et al., 2002; Alhalabi and Horiguchi, 2001) but all theses systems require heavy data exchange between two nodes to keep them consistent.

Visual and auditory substitution has already been used both in single user VR and teleoperation systems to provide pseudohaptic feedback. The sensory substitution may be used as a redundant cue, due to lack of appropriate haptic device or to avoid the possible instabilities in case of real force feedback (Richard et al., 1996).

3 DESCRIPTION OF THE SYSTEM

We present our system that enables two remote users(connected via LAN), to cooperatively manipulate virtual objects using string based parallel robots in the VE. In addition we present the use of visual (shadow and arrows) aids and oral communication to facilitate the cooperative manipulation.

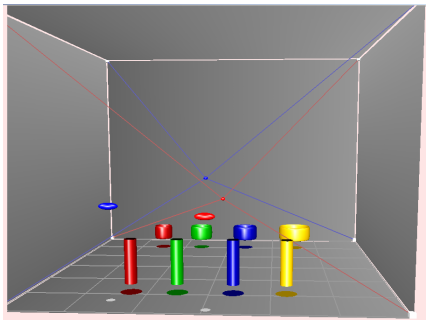


Figure 1: Illustration of the virtual environment.

The VE for cooperative manipulation has a simple cubic structure, consisting of three walls, floor and ceiling. Furthermore the VE contains four cylinders each with a distinct color and standing lengthwise in a line. In front of each cylinder at some distance there is a torus with same color. We have modeled two SPIDAR (3DOF) to be used as robots (Richard et al., 2006)(see figure 1). At each corners of the cube a motor for one of the SPIDAR has been placed. The end effectors of the SPIDARs have been represented by two spheres of distinct color. Each end effector uses 4 wires (same in color) for connection with its corresponding motors.

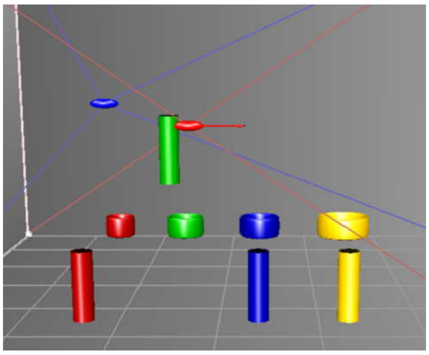


Figure 2: Illustration of the appearance of arrow.

We use two spheres which are identic in size but different in colors (one is red and the other is blue) to represent the two users. Each pointer controls the movements of an end effector. Once a pointer collides with its corresponding end effector, the later will follow the movements of the former. In order to lift and/or transport a cylinder the red end effector will always rest on right and blue on left of the cylinder.

3.1 Use of Visual Aid and Oral Communication in Cooperative Work

Cooperative work is really a challenging research area, for example the co-presence and awareness about collaborator's actions is essential. Similarly the cooperating persons should also have some feedback to know, when they can start together, or if there is some interruption during task. For this purpose we exploit visual(arrow and shadow) feedback and oral communication.

If any user moves to touch a cylinder on its proper side, an arrow appears pointing in the opposite direction of the force applied by the end effector. The arrow indicates the collision between an end effector and cylinder. Similarly during the transportation, if any user looses control of the cylinder, his/her arrow will disappear and the cylinder will stop moving. Here the second user will just wait for the first one to come back in contact with the cylinder. It means that the two users will be aware of each other's status via arrows during task accomplishment.(see figure 2)

In order to have the knowledge of perspective positions of various objects in the VE, we make use of shadow (see figure 1) for all objects in the environment. The shadows not only give information about the two end effector's contact with cylinder but also provide feedback about the cylinder's position with

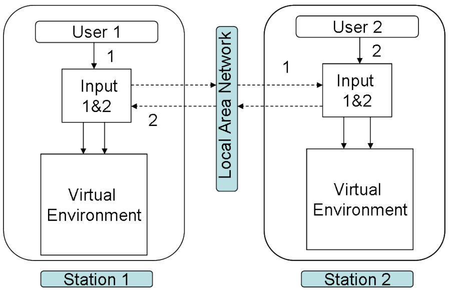


Figure 3: Illustration of the framework of cooperative virtual environment.

respect to its corresponding torus during transportation.

Normally human beings frequently make use of oral communication while performing a collaborative or/and cooperative task. In order to accomplish the cooperative work in a more natural manner, we include a module for oral communication in our system. For this purpose we use TeamSpeak software that allows the two users to communicate over the network using a headphone equipped with microphone (tea,).

3.2 Framework for Cooperative VE

The framework plays a very important role in the success of collaborative and/or cooperative VEs. We use a complete replicated approach and install the same copy of the VE on two different machines. As the figure 3 depicts each VR station has a module which acquires the input from the local user. This input is not only applied to the local copy of the VE, but is also sent to the remote station. It means that a single user simultaneously controls the movement of two pointers (in our case a sphere) at two different stations, so if this pointer triggers any event at one station, it is also simultaneously applied at other station. In order to have reliable and continuous bilateral streaming between the two stations, we use a peer-to-peer connection over TCP protocol. Here it is also worth mentioning that the frequently exchanged data between the two stations is the position of the two pointers where each is controlled by a user.

3.3 Experimental Setup

We installed the software on two pentium 4 type personal computers connected through Local network. Each machine had processor of 3GHZ and 1GB memory. Each system is equipped with standard graphic and sound cards. Both the systems used 24 inch plate LCD tv screen for display. Similarly each VR system is equipped with a patriot polhemus (pat,) as input device. The software was developed using C++ and OpenGL Library.

4 EXPERIMENTATION

4.1 Procedure

Ten volunteers including five male and five female participated in the experimentations. They were master and PhD students. All the participants performed the experiment with same person who was expert of the domain and also of proposed system. They were given a pre-trial in which they experienced all feedback. The users needed to start the application on their respective machines. After the successful network connection between the two computer the user could see the two spheres (red and blue) as well as the two end effector of SPIDARs on their screens. Seeing the two spheres they were required to bring their polhemus controlled spheres in contact with their respective end effectors (i.e red+red and blue+blue). The red sphere was assigned to the expert while the subjects were in charge of the blue one. In order to pickup the cylinder the expert needs to touch it from right while the subject should rest on its left. The experiment was carried out under the following four conditions. C1= only shadow, C2= shadow + arrows, C3= shadow + arrows + oral communication, C4= No aid All the ten groups performed the experiment using distinct counter balanced combinations of the four conditions. We recorded the task completion time for each cylinder. The time counter starts for a cylinder once the two end effectors have an initial contact with it, and stops when it is properly placed in the torus.

4.2 Task

The task was to cooperatively pick up a cylinder and put it into the torus. The users were required to place all the cylinder in their corresponding toruses in a single trial. Each group performed exactly four trials under each condition. The order of selection was also the same for all groups i.e to start from the red, go on sequentially and finish at yellow.

4.3 Task Completion Time

For task completion time the ANOVA (F(3,9)= 16.02, $p < 0.005$) is significant. Comparing the task completion time of C1 and C2, We have 30.07 sec (std 6.17) and 22.39 sec (std 3.10) respectively with a significant ANOVA. This result shows that arrow has an influence on task performance. Similarly comparing C4

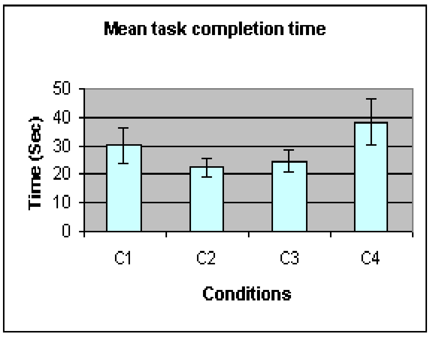


Figure 4: Task completion time under various conditions.

(mean 38.31 sec , std 7.94) with C1 also gives significant ANOVA. This indicates that only "shadow" as compare to "No aid" also increases user performance.

Now we compare the mean 22.39 sec (std 3.10) of C2 with that of C3 (24.48 sec std 3.93), the ANOVA result is not significative. It shows that users had almost the same level of performance under C2 and C3. On the other hand the comparison of C2, C3 with C4 (mean 38.31 sec , std 7.94) both have statistically significant results (see figure 4).

5 CONCLUSIONS

In this paper we simulate the use of two string based parallel robots in cooperative teleoperation task, two users setting on two separate machines connected through local network operated each robot. In addition the article proposed the use of sensory feedback (i.e shadow, arrows and oral communication) and investigated their effects on cooperation, co-presence and users performance. We observed that visual cues (arrows and shadow) and oral communication greatly helped users to cooperatively manipulate objects in the VE. These aids,specially arrows and oral communication also enabled the users to perceive each others actions. Our investigations will help in the development of teleoperation systems for cooperative assembly, surgical training and rehabilitation systems. Future work may be carried out to integrate the modality of force feedback.

REFERENCES

www.polhemus.com.

www.teamspeak.com/.

Alhalabi, M. O. and Horiguchi, S. (2001). Tele-handshake: A cooperative shared haptic virtual environment. *in Proceedings of EuroHaptics*, pages 60–64.

Basdogan, C., Ho, C.-H., Srinivasan, M. A., and Slater, M. (2001). Virtual training for a manual assembly task. In *Haptics-e*, volume 2.

Chastine, J. W., Brooks, J. C., Zhu, Y., Owen, G. S., Harrison, R. W., and Weber, I. T. (2005). Ammp-vis: a collaborative virtual environment for molecular modeling. In *VRST '05: Proceedings of the ACM symposium on Virtual reality software and technology*, pages 8–15, New York, NY, USA. ACM.

Greenhalgh, C. and Benford, S. (1995). Massive: a collaborative virtual environment for teleconferencing. *ACM Transactions on Computer Human Interaction*, 2(3):239–261.

Jordan, J., Mortensen, J., Oliveira, M., Slater, M., Tay, B. K., Kim, J., and Srinivasan, M. A. (2002). Collaboration in a mediated haptic environment. *The 5th Annual International Workshop on Presence*.

Otto, O., Roberts, D., and Wolff, R. (2006). A review on effective closely-coupled collaboration using immersive cve's. In *VRCIA '06: Proceedings of the 2006 ACM international conference on Virtual reality continuum and its applications*, pages 145–154. ACM.

Richard, P., Birebent, G., Coiffet, P., Burdea, G., Gomez, D., and Langrana, N. (1996). Effect of frame rate and force feedback on virtual object manipulation. *PRESENCE : Massachusetts Institute of Technology*, 5(1):95–108.

Richard, P., Chamaret, D., Inglese, F.-X., Lucidarme, P., and Ferrier, J.-L. (2006). Human-scale virtual environment for product design: Effect of sensory substitution. *International Journal of Virtual Reality*, 5(2):3744.

Sallnas, E.-L., Rassmus-Grohn, K., and Sjostrom, C. (2000). Supporting presence in collaborative environments by haptic force feedback. *ACM Trans. Comput.-Hum. Interact.*, 7(4):461–476.

Shirmohammadi, S. and Georganas, N. D. (2001). An end-to-end communication architecture for collaborative virtual environments. *Comput. Netw.*, 35(2-3):351–367.

COLLISION-MODEL BASED MOTION PLANNER FOR MULTI-AGENTS IN A FACTORY

S. H. Ji, W. H. Ko, K. T. Nam and S. M. Lee
Korea Institute of Industrial Technology, Sa-1-dong, Sangrok-gu, Ansan-si, KyungKi-do, South Korea
{Robot91, whko, robotnam, lsm}@kitech.re.kr

Keywords: Collision map, Collision Model, Multi-agents, Mobile-robot motion planning, Priority.

Abstract: It is well known that Mathematical solutions for multi-agent planning problems are very difficult to obtain due to the complexity of mutual interactions among multi-agents. We propose a practically applicable solution technique for multi-agent planning problems, which assures a reasonable computation time and a real world application for more than 3 multi-agents. First, based upon the collision map the collision features of multi agent is analyzed. The collision map is used for the collision avoidance of two industrial manipulators based upon the priority. Second, collision model ((M,D) network model) based upon the studied collision features is suggested in order to express the traveling features of multi agents. Finally, an interactive way to design the collision-free motion of multi agent on the network model is proposed.

1 INTRODUCTION

Multi-agent motion planning is one of the interesting and essential research fields in robotics. The demand for various specialized robots has been increasing rapidly with the advancement of robot technology.

Multi-agent motion planning has been studied for the last several decades. Multi-agent motion planning, however, is still a challenging field of research, having some technical difficulties in resolving conflict among agents. The centralized approaches have been faced with problems such as the curse of dimensionality, complexity, computational difficulty, and NP-hard problem (Canny, 1988; Akella, 2002).

To overcome these problems in the approach, we proposed the extended collision map method (Ji, 2007). We modified the collision map such that the method enables N agents to proceed with the collision-free operation according to the priority by going on the collision avoidance process one after another from the highest priority agent.

Yet, in this method, the mutual relation regarding the collision region among agents was not analyzed.

In this regard, in this paper the mutual relation regarding the collision region is analyzed, and based upon the studied collision features, (M,D) network model which can express the traveling features of multi agent is shown. (M,D) network model can express not only the collision features between two agents but also the complicated mutual interference among more than three agents. Likewise, the collision-free operation of multi agent can be designed and the operating finish time of agents can be figured by using (M,D) network model.

The remainder of the paper is organized as follows: Section 2 defines our research and the detailed approach conceptually. Section 3 presents the concept of the key technique of this paper – Collision model. Section 4 provides the way how to plan collision-free motion of multi-agents based on the (M,D) network model. Finally, this paper is concluded in Section 5.

2 PROBLEM STATEMENTS

2.1 Assumptions

To overcome the drawbacks of the centralized approach, the extended collision map method applies several concepts as follows:

The intelligent space can provide a central planner with essential and necessary information for motion planning and motion monitoring. This information includes all the agents' motion status and all the static and moving obstacles' positions (Lee, 2000; Norihiro, 2003).

Global off-line path planner (Central planner) can give the safe paths to all agents. In this paper, ‘safe path’ is the meaning that no agent will not crossover any other agent’s starting point or destination if it keeping on its own safe path. Therefore there can be intersection points among agents’ paths.

2.2 Collision Map

The concept of the original collision map was presented in the previous study (Lee, 1987). The original concept is as follows: An agent with a higher priority is called 'agent 1', and an agent with a lower priority is called 'agent 2'. The radii of the two agents are r_1 and r_2 respectively. Using the obstacle space scheme, agent 1 can be represented as the agent with a radius of r_1+r_2, and agent 2 can be considered as a point agent. The original trajectory of agent 1 is assumed not to be changed. On the contrary, agent 2 must modify its trajectory if a collision is anticipated.

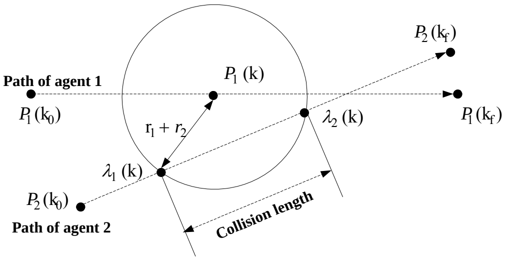


Figure 1: Paths of two agents and collision.

If the path of agent 2 meets agent 1 with radius of r_1+r_2, the two agents will collide with each other. At this instant, the part of agent 2's path that overlaps with agent 1's path, is called the 'collision length', which is denoted by the portion between $\lambda_1(k)$ and $\lambda_2(k)$ in Fig. 1. These overlapped parts are examined at every instant of the sampling time k to construct a 'collision region.' If the TLVSTC (traveled length versus servo time curve, simply trajectory) of agent 2 arrives at the region, the two agents will collide with each other under the original trajectories. This colliding case is shown in Fig. 2. In this figure, the vertical axis represents the traveled length of agent 2 and the horizontal axis represents the elapsed time.

Because it is difficult to mathematically represent the boundary line of the collision region, the concept of ‘collision box’ was introduced. This concept can be explained in Fig. 2. In this figure, k_s is the time when agent 1 starts overlapping agent 2’s path. Also k_e is the time when agent 1 leaves agent 2’s path. l_s and l_e are the minimum and maximum values of the collision length in the collision region, respectively.

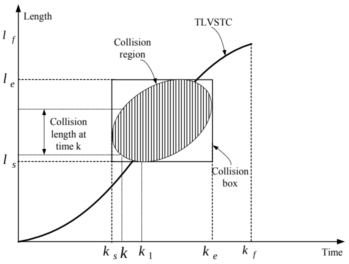


Figure 2: TLVSTC and collision region.

2.3 Extended Collision Map

The extended collision map method considers more than two agents which have many intersections in workspace. Thus, the intersection and its corresponding collision region should be described. An intersection is denoted by the symbol

$$I_{ij}^{k}\ ; i > j \qquad (1)$$

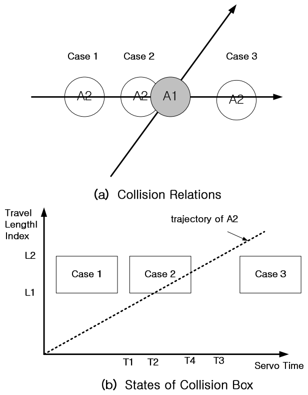
where i and j represent the identifying number of the agent, and k is the ordering number denoting intersections along the path of the agent i from the starting point. The corresponding collision region of the intersection is expressed as R^{k}_{ij}.

3 COLLISION MODEL

3.1 Collision Characteristics

We assume A1 has an intersection point with A2 which is less important than A1 in Fig.3(a). The possible position relations between two agents around the intersection point are as followed; First, A1 passes through the intersection region before A2 enters the region(Case1). Second, the agents collide with each other(Case2). Third, A1 reach the region only after A2 exits the region. The states of collision box related the agents in Fig.3(a) as shown in Fig.3(b), where L1 and L2 are the minimum traveled length and maximum length from start position to the intersection region along A2’s path.

Time characteristics related to collision region including T_k(k=1,2,3,4) in Fig. 3(b) are shown on Table I, and we define two variables, ‘M’ and ‘D’, in order to describe the collision states among agents.



(a) Collision Relations

(b) States of Collision Box

Figure 3: Collision-States of two agents.

Table 1: Characteristics related to collision region.

Variables	Meaning
T_1	Time when A1 reaches the collision region
T_2	Time when A2 reaches the collision region
T_3	Time when A2 exits the collision region
T_4	Time when A1 exits the collision region
T_{1d}	A2's delayed start time
$\mathrm{T}_{2\mathrm{d}}$	A2's delayed start time
M	T_3-T_1
D	T_4-T_2,

We can predict whether the agents collide with each other by the variables, M and D, related to the collision region and define the collision-free navigation condition of an agent as followed:

Collision-Free Navigation Condition. When an agent has more than one intersection with other agents which have higher priorities than the agent, it should not have any collision region of which collision characteristics are positive.

3.2 Impact of Time Delay on Characteristics

When A2, the agent with lower priority, is delayed in departure by T_{2d} without change in path shape nor velocity profile in order to avoid a collision with A1, the time variables are changed as followed:

Because the agents keep up their own path shape and A1 keeps up its velocity profile, neither T_1 nor T_4 is affected by A2's delayed departure. T_2 and T_4 which are related to the agents' path shape and A2's TLVSTC are exchanged with $T_2 + T_{2d}$ and T_3+T_{2d}, because A2's TLVSTC is shifted to the right by T_{2d} in Fig. 3(b). Thus, impact of time delay on collision characteristics is define as shown in Eq. (2).

$$\begin{aligned} M' &= M + T_{2d} \\ D' &= D - T_{2d} \end{aligned} \tag{2}$$

where K_0 is a constant which is determined initially by the agents' paths shapes and velocity profiles. According to Eq.(2) M increases and D decreases when A2 is delayed in departure.

3.3 Collision Model

We present the collision model which express collision relations and predict possibility of collisions among the agents. And all of the agent's minimum delayed departure time for collision-free navigation can be extracted from the model. The elements of collision model are defined in Table 2.

Now, we express the collision model from the case in Fig. 4 as the network model shown in Fig. 5. There are three agents (agent 1, agent 2, and agent 3) with path shapes as shown in Fig.4. We assume that all of agent's radii are 5m and there velocities are 1m/sec, 2m/sec, and 1m/sec. We assume also that it takes no time for them to accelerate, decelerate, or turn around. And we assume their priority order is 1-2-3.

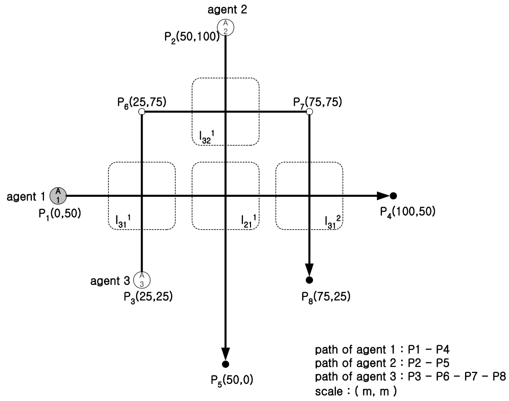


Figure 4: Three agents with intersection points.

The collision network model is as followed: V = {1,2,3}, P=(1,2,3), E={(2,1,1), (3,1,1), (3,1,2), (3,2,1)}. L and T are shown in Fig. 5.

When an agent(A_i) is delayed by T_i^d, the collision model is changed related the agent node. For inlet links from the higher priority agents, M's increase and D's decrease by delayed departure time(T_i^d). In the other, for outlet links to lower priority agents, M's decrease and D's increase by the same amount.

Table 2: Elements of collision model.

Symbols	Meaning
V	Node space(V) = { 1, …, N}. This is a set of agent identified numbers.
E	Link space(E) = { (i, j, k) ∈ V^2 x N | i ∈ P^+_j , k=1,…, k(i,j) }. This is a set of collision regions among agents. P^+_j is explained in priority order space, and the links go from the agent with higher priority to the other agent. k(i,j) is the number of collision regions between agent j and agent i . So some agent can have more than two links with other agent if they have several collision regions
C	Link relation space(C) = { (M_{ij}^k, D_{ij}^k) ∈ R^2 | (i,j,k) ∈ E }. This is a set of collision characteristics, M and D in the Table I.
T	Node navigation characteristic space(T) = { ($T_i^{delayed}$, $T_i^{traveled}$) ∈ R^2}. This is a set of agents' delayed departure times and pure traveled time from the start point to the destination.
P	Priority order space(P) = {(N^1, …, N^N) ∈ V^N | N^i is the identified number of the agent with the ith highest priory} This is a set of agent orders in which each agents are placed from an agent with the highest priority to an agent with the lowest priority. P^+_j is the set of agents which have higher priorities than agent j in P and P^-_j is the set of agents which have lower priorities than agent j in P, the space of priority order space

4 COLLISION MODEL BASED MULTI-AGENT MOTION PLANNER

As a result of the time delay, the safe inlet link may be dangerous. So in this paper we propose an iterative approach to find the minimum delayed departure time for collision avoidance as followed:

Collision-Free Motion Planner for an Agent on Collision Model

Step1. We extract the links on which the agent is expected to collide with higher priority agents(Inlet Links) by use of collision characteristics.

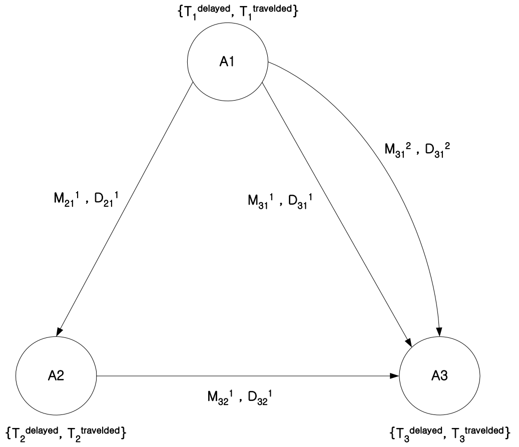

{$T_1^{delayed}$, $T_1^{travelded}$}
A1
M_{31}^2 , D_{31}^2
M_{21}^1 , D_{21}^1
M_{31}^1 , D_{31}^1
A2
A3
M_{32}^1 , D_{32}^1
{$T_2^{delayed}$, $T_2^{travelded}$}
{$T_3^{delayed}$, $T_3^{travelded}$}

Figure 5: Collision model for three agents in Figure4.

Step2. We define an instantaneous delayed departure time (T_i^d) as the maximum of the Ds' in the selected links.

$$T_i^d = \max (\{D_{ij}^k \mid j \in P+(i), (i, j, k)\in E \text{ s.t. } M_{ij}^k > 0 \text{ and } D_{ij}^k > 0\}) \quad (3)$$

Step3. We modify node variables, link parameters by T_i^d.

Step4. If there is no inlet links to the agent which is dangerous, the agent can go to its destination safely. Otherwise, we execute above actions from the first stage.

Collision-Free Motion Planner for Multi-Agents on Collision Model

First, we select an agent from the priority order space (P) by use of priority index.

Second, if the agent has the highest priority, go to first stage. Otherwise, we apply the collision-free motion planner on collision model to the agents so that the agent can navigate safely.

Third, if the selected agent has the lowest priority, the all of the agents can navigate safely, and finish up this algorithm. Otherwise, increase priority index by 1 and go to first stage.

The procedure of this algorithm for the three agents in Fig. 4 is shown in Fig. 6. Because the all agents' links is in a safe state in Fig. 6(d), we can predict that the agents can navigate without collision among them.

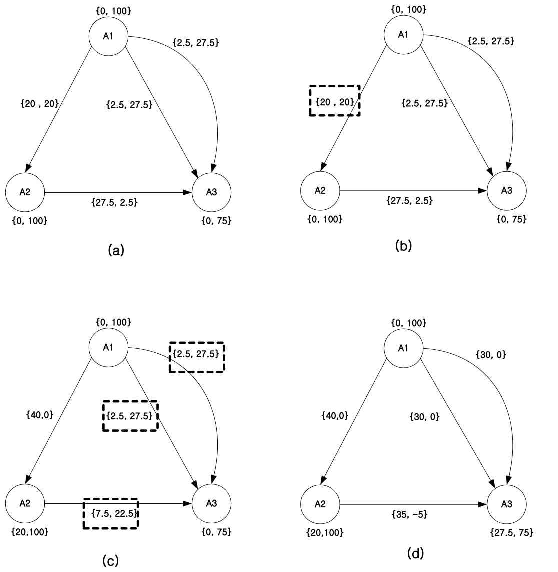


Figure 6: Procedure of collision-free motion planner on collision model for the agents in Figure 4.

5 CONCLUSIONS

In this paper, we present a systematic approach to the multi-agents motion planning problem. In this regard, in this paper the mutual relation regarding the collision region is analyzed, and based upon the studied collision features, (M,D) network model, collision model, which can express the traveling features of multi agent is shown. Collision model can express not only the collision features between two agents but also the complicated mutual interference among more than three agents. Likewise, the collision-free operation of multi agent can be designed and the operating finish time of agents can be figured by using collision network model.

Because our method is fast and scalable, complete, so our method can be used practically to multi-AGVs in factories, airports, and big buildings where there are sensor networks obtaining global position information.

ACKNOWLEDGEMENTS

This work was supported by the Next-Generation New Technology Development Programs from the Ministry of Knowledge Economy(MKE).

REFERENCES

J.C.Latombe, 1991. *Robot Motion Planning*, Kluwer academic publishers.

M.M.Quottrup, T.Bak, R.Izadi-Zamanabadi, 2004. *Multi-Robot Planning : A Timed Automata Approach*, Proc. of IEEE Int. Conf. on Robotics and Automation.

J.F.Canny, 1988. *The Complexity of Robot Motion Planning*, MIT Press.

K.Azarm and G.Schmit, 1997. *Conflict-free Motion of Multiple Mobile Robots Based on Decentralized Motion Planning and Negotiation*, Proc. of IEEE Int. Conf. on Robotics and Automation.

B.H.Lee, C.S.G.Lee, 1987. *Collision-Free Motion Planning of Two Robots*, IEEE Transactions on Systems, Main, and Cybernetics, vol. 17, no 1, pp. 21-31.

K.S.Barber, T.H.Liu, and S.Ramaswamy, 2001. *Conflict Detection During Plan Integration for Multi-Agent Systems*, IEEE Transactions on Systems, Man, and Cybernetics, vol. 31, no. 4, pp. 616-627.

J.H.Lee and H.Hashimoto, 2000. *Intelligent space*, Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, vol. 2, pp. 1358-1363.

S.H.Ji, J.S.Choi, and B.H.Lee, 2007. *A Computational Interactive Approach to Multi-agent Motion Planning*, International Journal of Control, Automation, and Systems, vol. 5, no. 3, pp. 295-306.

S. Akella and S. Hutchinson, 2002. *Coordinating the Motions of Multiple Robots with Specified Trajectories*, Proc. Of IEEE Int. Conf. on Robotics and Automation.

H.Norihiro, K.Kiyoshi, M.Kehji and S.Yasuyuki, 2003. *Collaborative Capturing of Experiences with Ubiquitous Sensors and Communication Robots*, Proc. of IEEE Int. Conf. on Robotics & Automation, pp. 4166-4171.

QUANTIFIED ONTOLOGIES FOR REAL LIFE APPLICATIONS

Lucia Vacariu[1], George Fodor[2]
[1] *Department of Computer Science, Technical University of Cluj Napoca, 26 Baritiu str, Cluj Napoca, Romania*
[2] *ABB AB Process Automation, Vasteras, Sweden*
Lucia.Vacariu@cs.utcluj.ro, george.a.fodor@se.abb.com

Gheorghe Lazea, Octavian Cret
Department of Automation, Department of Computer Science, Technical University of Cluj Napoca, Cluj Napoca, Romania
Gheorghe.Lazea@aut.utcluj.ro, Octavian.Cret@cs.utcluj.ro

Keywords: Merging Ontologies, Symbolic Approaches to Control, Semantic Services, Mobile Robots.

Abstract: Industrial applications are using run-time symbolic approaches only when formal methods can assign useful meaning to symbols by computationally inexpensive algorithms. However, most reasoning methods are either computationally prohibitive or may compute indefinitely; thus such methods have limited use in industrial applications. In many practical situations, the uncertain environment in which an "intelligent" control system acts consists of the symbolic space of some other "intelligent" control system, both networked in the same name space. The result of such interaction is to establish relations between heterogeneous vocabularies and reasoning agents, and between symbols and the physical environment in which the connected systems act. This paper introduces and motivates the necessity for on-line quantification of the degree to which symbols in a system have their intended meaning.

1 INTRODUCTION

The theory presented here has relevance for distributed real-time systems such as those used in multi-robot applications or in distributed manufacturing industries. These systems have a large set of symbols in the form of names for components, signals, process states or configuration parameters. Heterogeneity manifests by units being of different specialization and of different make. Having the right meaning of each symbol is essential for a correct operation of the system. The costs for matching all signals, communication protocols and sub-products during a tender process for a complex system turn out to be a significant part of the total cost of the system. Moreover, after delivery, ensuring that the final system behaves according to specifications can be a lengthy and highly qualified process. The solution to this problem is to establish an ontology for the given industrial domain. These tools need to reduce symbol complexity by automatic information processing, such as via semantic web and ontological languages.

It is an uncommon situation today that such configuration tools work across dissimilar firms or markets, though many core technologies and standards are available. On a theoretical level, the operations needed, such as ontology merging, alignment composition, union and intersection are still under research (Furst, 2008).

We stress here that formal design verification cannot replace the ontological compliance presented in this paper: even a perfectly designed system that is formally proven to follow a design might encounter a complex environment that does not follow the assumptions in the specification.

Seen as software architecture, ontologies are implemented at the current level of technology as services. These can be organized as local services in each unit or as a combination of a hierarchic set of services – local and specialized - with indirections provided by name servers. Without specifying details, we call in this paper a generic ontology service as the "Industrial Ontology Server" (IOS) (Figure 1).

Practically, an IOS should be able to infer the structure of any type of distributed industrial application. Of course, this is a very ambitious claim, well beyond the forefront of what is available today in academia or industrial research institutes.

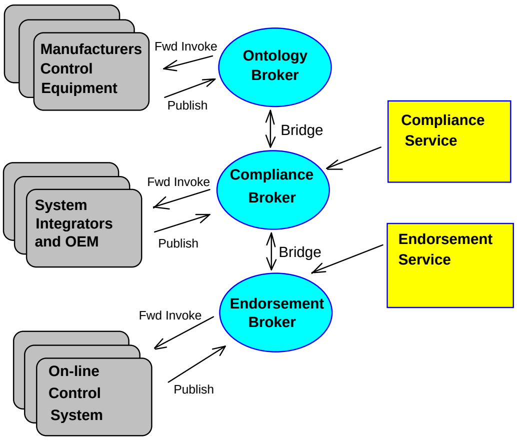


Figure 1: Federated Service Architecture IOS.

The solution proposed here is to use a minimal, uniform ontology associated to products and that each actual systems to update automatically the required operational and dynamic information into IOS-es. With this approach, biding costs would be substantially reduced and market participation would guarantee more objectivity and flexibility.

Clearly a multi-agent system architecture could implement such requirements. Tools such as JADE (Java Agent DEvelopment framework) with Protégé (Ontology editor and knowledge-base framework) could handle the design of systems based on a common design ontology (Tomaiuolo et al., 2005).

2 SOME RELEVANT TECHNOLOGIES AND THEORIES

An IOS is relevant if it considers all levels of an industrial unit included in a system: from signals and actuators, up to overall goals, including states, alarms, resource allocation, synchronization with other units, etc.

2.1 Ontologies in Process Control

Ontologies for automation and process control applications have several specific layers. At the *design phase*, there is available a formal or informal description of the domain and of the constraints of the design (*Design Model*). The standard for IEC61499 prescribes the Engineering Support System (ESS) that can perform certain syntactic and semantic verifications. Valuable research is being conducted for improving ESS tools (Thamboulidis, Koumoutsos, and Doukas, 2007).

The framework (middleware) in which the program, agents or components execute has own ontology and semantics that limits what the application program can execute (*Execution Model*).

Application Ontology. This is the ontology that effectively decides on goals and actuation. The domain of discourse is not necessarily the same as the domain used for the design ontology.

Visualized Ontology. The ontology is typically visualized on a human-readable interface. The domain of discourse, taxonomy among objects and object properties are represented graphically. Automatic generation of visualization using as input ontologies expressed in XML/RDF would be an important advance in technology.

Communicated Ontology. From the execution model and design data, system designers extract an ontology used for communicating among cooperating systems. This ontology may not have the domain of the Design Model, nor of the Execution Model.

Ontology does not enter in a formal, verified way in the design of large control products, as tools are not mature enough. Another conclusion is that a system has several ontologies that should be aligned. There are no commercial tools that can align ontologies. Interesting research results are reported using category theory (Zimmermann et al., 2006).

3 METRICS FOR ON-LINE ONTOLOGIES

3.1 The Decision-Control Space

Essential for taxonomy of process control systems is the type of actuators used. Actuators are performing changes in the real world; their semantics is determined by physical laws.

Independently of the software architecture type, a control system has two essential parts: (a) a decision (information) level and (b) a physical, energy-related level of actuator and plant changes. All the relevant information from sensed signals used for decision forms a hyperspace with each coordinate being one kind of decision information. Let this space be H with N dimensions, $H \in R^N$. Chains of decisions generate chains of action trajectories $T_i(i = 1,...,M)$ in this space.

Trajectories may not be continuous as disturbances create 'jumps' from one possible trajectory to another. A valid place on this trajectory is often not a point but a hyper-sphere (a topology) since usually control decisions are taken within intervals and not on discrete points.

In traditional control, the space of all relevant signals is called a *state space* and the trajectory a *goal path.* Each point reached during control in the state space is a *state*. For each state space, a controller (or agent or component) has mapped a decision procedure that result in some action being taken. We are not concerned here with what kind of decisions or reasoning a controller is doing, but only with the mapping between actions $u_{i,j}$ and spheres $h_{i,j}$. Here the variable *u* denotes an action, and the variable *h* denotes a sphere in *H*. We call the pair $S_{\mathrm{i,j}} = (h_{i,j}, u_{i,j})$ as a state and use first index to denote the sphere *i* and the second index to denote a goal path *j*.

3.2 State Semantics

The *active state* is the one currently materialized by sensors. Even if the system consists of many agents or components, there is always a unique active state since the hyperspace covers the whole possible space. However, there may be multiple actions corresponding to each active state. We represent here all actions for one point in the hyperspace as a single action. Some typical state transitions we are interested in are the following: case (1) - normal control with no disturbances, case (2) - control with disturbances and case (3) - lockout.

Performing no action may be a legal, correct operation of the controller, however if inaction is due to decision lockout, then this case is distinct and should be detected.

We seek here moreover to quantify the level of true semantics states have.

3.3 Quantifying State Semantics

3.3.1 Intra-state Distance

The degree for how "strong" is a state with a current place $h_{i,j}$ in the space ***H*** is the inverse of the distance from $h_{i,j}$ to the centre of the sphere intended for that state. Therefore, closer $h_{i,j}$ is to the state sphere boundary, less correlation it has with the current state and its action. This is the typical situation when the true state is somewhere in between two states, none fully reached; fuzzy logic can quantify and correct this situation (Grantner and Fodor, 2002). For a measured probability distribution Q, the Kullback Leibler divergence of Q from P is:

$$D_{KL}(P \,||\, Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)} \qquad (1)$$

3.3.2 Inter-state Endorsement

The following levels of endorsement for a state are defined to characterize how well predictions are built into the semantics of a state materialize.

Void-Endorsement. A state is void-endorsed if it is materialized. That means for a state $S_{\mathrm{i,j}} = (h_{i,j}, u_{i,j})$

$$\mathrm{ve}(\mathrm{S}_{\mathrm{i,j}}) \text{ iff } (\mathrm{h}_{\mathrm{i,j}} \wedge \mathrm{u}_{\mathrm{i,j}}) \qquad (2)$$

This is the simplest form of endorsement, but it tells an important think: that the program semantics about the environment matches at least once a real instance of the environment. The properties matched are those in $h_{i,j}$.

Weak State Endorsement. A state is weakly endorsed if a consecutive state of a void-endorsed state placed on the same goal path is weakly endorsed at the next instance of time.

$$\mathrm{we}(S_{i,k}) \text{ iff } \mathrm{ve}(S_{i,k}) \wedge \mathbf{O}\, \mathrm{ve}(S_{j,k}) \qquad (3)$$

Here '**O**' is the 'next time' logical operator; both states are on the same path T_k as the second index *k* shows. A we() state is not a goal state. Weak endorsement means that if a state has materialized and the controller has executed an action at that state, then the expected outcome really turned out to be true in the environment.

Strong State Endorsement. A state is strongly endorsed if a consecutive state on the same path materializes and both states are weakly endorsed.

$$\mathrm{se}(S_{i,k}) \text{ iff } \mathrm{we}(S_{i,k}) \wedge \mathbf{O}\, \mathrm{we}(S_{j,k}) \qquad (4)$$

A se() state is again not a goal state. This state endorsement tells that after a state materialize and the action executed, a next expected state indeed materializes as well and moreover the action from that second state has the expected effects.

3.3.3 Goal Path Endorsement

Goal paths are endorsed in similar way as states.

Void Endorsed Goal Path. A goal path T_k is void endorsed if there exists a state that is not the goal state of the path and which is weakly endorsed:

$$\mathrm{vpe}(T_k) \text{ iff } \exists i \; \mathrm{ve}(S_{i,k}) \qquad (5)$$

A void-endorsed goal path has some $h_{i,j}$ of some state that materializes in the environment, moreover the corresponding action is being executed, but there is no evidence that any of the following expected states on the same goal path have been materialized. Note that there may be states that are not on any goal path, so a void endorsed state may not necessarily mean a void endorsed goal path.

Weakly Endorsed Goal Path. A goal path T_k is weakly endorsed if there exists some state on the goal path that is weakly endorsed and which is not the goal state of the path.

$$\mathrm{wpe}(T_k) \text{ iff } \exists i \; \mathrm{we}(S_{i,k}) \qquad (6)$$

A weakly endorsed goal path has at least one state that when acting on the path, get expected effects on the same goal path. However, it is not sure that the expected state has the required quality that even its action will get expected results and thus the semantics of the second reached states is not entirely sure.

Strongly Endorsed Goal Path. A goal path T_k is strongly endorsed if there exists a state that is strongly endorsed on the goal path.

$$\mathrm{spe}(T_k) \text{ iff } \exists i \; \mathrm{se}(S_{i,k}) \qquad (7)$$

More generally, a goal path is n-strongly endorsed if there are n states which are strongly endorsed on the path. N-strong endorsement tells that many states on the goal path are semantically right, but there may be disturbances that materialize states interleaved with disturbances, on some other goal paths. The condition that one full goal path is traversed without interruption is given by the full-goal path endorsement: a goal path is full-goal endorsed if all the states of the goal path materialize in expected order up to the goal state. Clearly all states of a path that has full-goal endorsement are strongly endorsed, except the goal state and the state immediately before the goal state that is weakly endorsed.

3.3.4 Global Semantic Norms

Many types of norms can be conceived to quantify the level of true semantics using the endorsements given above. For example if $|h_{i,j}|$ is a normalized distance from the center of a state hyper-sphere to $h_{i,j}$ so that $|h_{i,j}| \leq 1$ and the norm $|\mathrm{se}(S_{i,k})|$ gives the number of states on the current goal path from the state i to the goal state, then a measure of the semantics of the current goal path, SM, is:

$$\mathrm{SM}(T_i)=|\mathrm{h}_{i,j}| +|\mathrm{se}(S_{i,k})| \qquad (8)$$

SM is a continuous, real valued function that shows how much of the current goal path has been completed.

4 CONCLUSIONS

Complex systems such as mobile robots systems, or distributed industrial control systems need to communicate and use ontological information about their environments and about the tasks they perform. Symbolic operations using formal methods are as yet prohibitive due to computational reasons while manual work raises substantially the costs of such systems. This paper presents a method that combines ontological operations defined formally with automatic updates for control ontology based on on-line direct sensory and actuation data.

REFERENCES

Furst, F., 2008. Ontology Matching with Axioms and Conceptual Graphs. In *IEEE Intelligent Systems*. Vol. 23, No. 6, pp. 73-75.

Grantner, J.L., and Fodor, G.A., 2002. Fuzzy Automaton for Intelligent Hybrid Control Systems. In *Proceedings of the 5-th Hybrid Systems Symposium*. 2002, Stanford, USA.

Thamboulidis, K.C., Koumoutsos, G.V., and Doukas, G.S., 2007. Semantic Web Services in the Development of Distributed Control and Automation Systems. In *Proceedings of the IEEE International Conference on Robotics and Automation*. April 10-14, 2007, Roma, Italy.

Tomaiuolo, M., Turci, P., Bergenti, F., Poggi, A., 2005. A Two-Level Approach for Ontology Management in Multi-Agent Systems. In *WETICE 2005, Proceedings of the 14th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprise.* June 13-15, 2005, pp. 21-26.

Zimmermann, A., Krötzsch, M., Euzenat, J., and Hitzler, P., 2006. Formalizing ontology alignment and its operations with category theory. In *FOIS 2006, Proceedings of the International Conference on Formal Ontology in Information Systems.* November 9-11, 2006, Baltimore, Maryland, USA.

ONLINE CALIBRATION OF ONE-DIMENSIONAL SENSORS FOR ROBOT MANIPULATION TASKS

Jan Deiterding and Dominik Henrich
Lehrstuhl für Angewandte Informatik III
Universität Bayreuth, D-95440 Bayreuth, Germany
{jan.deiterding, dominik.henrich}@uni-bayreuth.de, http://www.ai3.uni-bayreuth.de

Keywords: Learning and adaptive systems, Architectures and programming, Compliant assembly.

Abstract: The purpose of this paper is to enable a developer to easily employ external sensors emitting a one-dimensional signal for flexible robot manipulation. To achieve this, the sensor must be calibrated using data tuples describing the relation between the positional change of the supervised object and the resulting sensor value. This information is used for adaptation methods, thus enabling robots to react flexibly to changes such as workspace variations or object drifts. We present a sensor-independent method to incrementally generate new data tuples describing this relation during multiple task executions. This method is based on the Secant method and is the only generally applicable solution to this problem. The method can be integrated easily into robot programs without detailed knowledge about its functionality.

1 INTRODUCTION

Industrial robots are able to perform complex tasks with utmost precision and at high speed without exhibiting symptoms of fatigue. However, these tasks are nearly always executed in a fixed environment, i.e. the precision is achieved by ensuring that all objects are placed in exactly the same position every time. All parts must have the same dimension, position, orientation, etc. Only by employing external sensors such as vision or force/torque sensors, a robot can deal with imprecisions and variations in objects and the environment. When designing such programs for more flexible robots, a developer faces the problem of determining the relation between the sensor value obtained and the actual physical variation of the supervised object.

The task is to find a *change function* that transforms sensor values into Cartesian descriptions of the change in order to successfully deal with these. The classical approach is to analytically determine a function describing this mapping. However, for complex sensors this task quickly becomes difficult and it is sometimes simply not possible to find an analytical solution if the underlying physical principles are unknown to the developer. In these cases data tuples describing the relation between the positional change of an object and the resulting sensor value are recorded and a selected type of function is fitted to these tuples. These approaches require a large amount of analysis and programming before the robot executes the task for the very first time. Another downside is, that this pre-calculated solution is fixed and prevents the robot from adapting to changes of the environment. For example, the robot must be stopped and re-calibrated if a drift in the workspace or the sensor system occurs. The advantage in the use of change functions is that an additional layer of abstraction is introduced. The program can be designed independent from the actual sensor because all workspace changes are described in Cartesian coordinates. Now, we may replace the sensor with a different one using another measuring principle and – as long as the change function is correct – no alterations have to be made to the program. General features of change functions are described in (Deiterding, 08) and a general outline to determine these functions is given, but no generally applicable method is presented to calibrate sensors iteratively during the execution of a robot manipulation task.

In this paper, we focus on sensors emitting one-dimensional signals, such as distance or force/torque sensors. We do not deal with imaging sensors as this class of sensors usually requires an upstream pattern matching algorithm to distinguish the relevant information from background data. We show how calibration data for a change function can be computed iteratively during the first executions of the task and how these methods can be integrated easily into the programming environment, only requiring the developer to specify a minimum of task-dependent parameters. Additionally, we show how the robot adaptively optimizes the task with respect to execution time based on a steadily improving approximation of the function.

The rest of this paper is organized as follows: In Section 2, we give a short overview of related work concerning this topic. In Section 3, we will outline a framework with which a developer can create sensor based robot programs that automatically acquire calibration data during execution. In Section 4, we describe which algorithms are encapsulated into this framework and compare them with other approaches. Section 5 describes how a typical robot task can be solved using our approach. In the last section, we give a short summary of our work and discuss further steps.

2 RELATED WORK

The task of inferring information from noisy sensor data is covered thoroughly by various books on pattern classification, e.g. (Duda, 00). But all of these describe methods for extracting the relevant information from sensor values, assuming that this information is present in the data. Multiple papers dealing with the planning of sensing strategies for robots exist, e.g. (Leonhard, 98), (Rui, 06). Most of these involve a specific task (Adams, 98), (Hager, 90) or are aimed at employing multi-sensor strategies (Bolles, 98), (Dong, 04). Various papers deal with the use of sensors in the work cell to allow for information retrieval (Hutchinson, 88). In (Kriesten, 06), a general platform for sensor data processing is proposed, but once more it is assumed that the sensors are already capable of detecting changes. More general discussions of employing sensors for robot tasks can be found in (Firby, 89), (Pfeifer, 94).

Two types of sensors are typically used for manipulation tasks: Force/torque and vision sensors. When force/torque sensors are employed, maps may be created describing the measured forces with respect to the offset to the goal position. (Chhatpar, 03) describes possibilities to either analytically compute these maps or create them from samples. Based on this, (Thomas, 06) shows how these maps can be computed using CAD data of the parts involved in the task. In both cases, the maps must be created before the actual execution of the task and are only valid if the parts involved are not subject to dimensional variations. If the information is acquired using cameras, the first step is to perform some kind of pre-processing of the data to extract the relevant information. To determine how this information relates to the positional variation is once again the task of the developer and highly dependent on the nature of the task. Examples for information retrieval using vision sensors are given in (Dudek, 96), (Paragios, 99) and (Wheeler, 96).

In summary, all of the papers mentioned above either propose specific solutions for specific types of sensors and tasks or algorithms to extract the relevant information from the sensor signal. A problem is that these solutions do not outline a general approach which can be used regardless of the type of sensor. Additionally, all papers assume that the developer is capable of integrating the methods into his own robot program. Unfortunately, this is usually not the case for developers in small and medium sized enterprises, which often possess only basic knowledge about robot programming.

Here, we are interested in determining the relationship between the sensor signal and the Cartesian deviation iteratively during multiple task executions. We want to integrate this algorithm into an easy-to-use interface that will enable developers having no special knowledge in robot programming to create adaptive robot programs. We only focus on one-dimensional data, such as distance sensors or force/torque sensors. Vision sensors always require some kind of pre-processing that is highly dependent on the task.

3 INTEGRATION INTO THE PROGRAMMING ENVIRONMENT

In this chapter, we will explain how a developer with minimal knowledge about sensor data processing can easily create robot programs that employ external sensors. We will explain which considerations must be made by the developer, how the program must be structured in general, and which parameters are mandatory.

3.1 Setting Up the Workspace

The first thing a developer has to do is to decide in which way a change can occur between consecutive executions of the task. Based on this, a suitable sensor must be chosen that is capable of recognizing this change and that satisfies the requirements imposed on change functions. Here we will only provide a short summary, see (Deiterding, 08) for a detailed explanation: A change function f describes the alteration of a sensor signal when the object supervised by the sensor has moved. It is a function that relates a Cartesian position to a sensor value. Using the inverse f^{-1} gives us the position p_{est} for the current sensor value. Note that all functions are defined in relation to a pre-set reference position p_{ref} and a corresponding reference sensor value s_{ref}. Only the difference of the current sensor signal to s_{ref} is taken into account. This is not a limitation, but rather a standardization of the function, so the only root of this function is (0,0) because there is only one reference position.

3.2 Online Computation of Change Functions

The central idea of this paper is that the change function f_{real}, which is defined by the task and the sensor, is unknown and cannot be calculated analytically or approximated beforehand. Instead, the robot will compute an approximation f_{est} of f_{real} online during the first executions of the task. Instead of two separate phases – the calibration of the sensor and the actual execution of the task – the calibration process is encapsulated in the execution (see Figure 1). The calibration may take longer now, nonetheless the program will work correctly right from the very first execution. In addition the developer will spend less time setting up the sensor and the program is capable of adapting to changes both in the workspace and in the sensor data, e.g. due to a warm-up of the sensor, without the need for a manual recalibration. The robot starts with a very rough approximation f_{est} of f_{real} and refines this approximation gradually with each execution by incorporating newly gained information.

During execution, the robot uses f_{est}^{-1} to react to Cartesian changes of the supervised object. If the object has moved away from p_{ref} by x_{change} to p_{change}, this is detected through the sensor value s_{act}:

$$s_{act} = f_{real}(x_{change}) \tag{1}$$

Thus, the robot must modify its movement by calculating:

$$x_{est} = f_{est}^{-1}(s_{act}) = f_{est}^{-1}(f_{real}(x_{change})) \tag{2}$$

The stored reference position is then modified accordingly:

$$p_{est} = p_{ref} + x_{est} \tag{3}$$

Now, the robot moves to p_{est}. If f_{est} is close enough to f_{real} then:

$$x_{est} = x_{change} \tag{4}$$

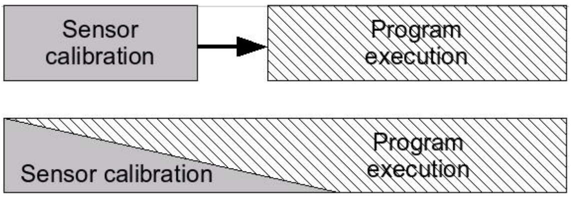


Figure 1: In the classical approach to sensor-based robot programming, the sensor is calibrated before the actual program is executed (top). In our approach, the calibration process is integrated into the execution cycle (bottom).

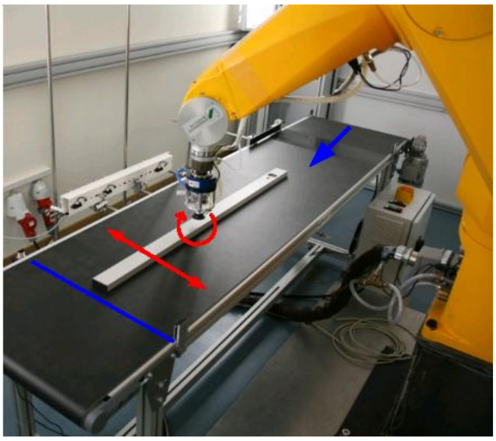


Figure 2: Experimental setup. A steel rod is delivered along a conveyor belt (blue arrow) until it reaches a light barrier (blue line). The rod can be in any position on the belt (red). Shown in this picture is the reference position of the rod in order to be picked up.

If the change was estimated correctly, this knowledge is incorporated into the change function. If the estimate was wrong, then there is not enough information stored in S to perform a reasonable correction using the current sensor value s_{act}. Thus, the correct position must be determined and f_{est} must be modified in such a way that the next estimate will be correct for the current sensor value. Initially this will often be the case since early values of f_{est} are quite inadequate.

When the robot has performed the motion defined by x_{est}, the new position is either correct or it is skewed because f_{est} was not accurate enough. In the latter case, two possibilities arise. The key point is to decide whether the robot motion will modify the sensor signal or not. This is best illustrated by an example. Consider the following task: A steel rod is delivered to the robot via a conveyor belt. The belt stops when the rod passes a light barrier (see Figure 2). The robot shall pick up the rod using a vacuum gripper and place it in a box for transport. To solve this task, we could construct a feeding mechanism ensuring that the rod is aligned the same way every time. However, we want to allow the rod to be in any position as long as it faces upwards. So we have translational changes along the x-axis and rotational changes around the z-axis of the coordinate system of the conveyor belt. To sense this misalignments, we employ two distance sensors that are placed parallel to the y-axis of the conveyor belt. (Figure 3) The developer faced with the task to design this robot program now has to plan how the position and orientation of the rod can be recognized and how the robot should react. So, there are two cases:

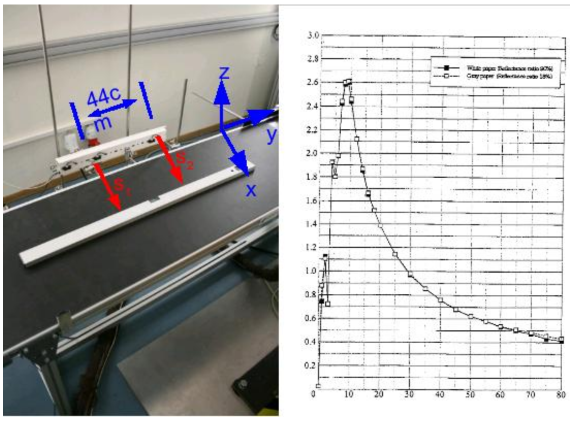


Figure 3: Left: Reference position of the rod and placement of the distance sensors to recognize the position and rotation of the rod on the conveyor belt. The distance is determined using s_1. The rotation is determined using the difference between s_1 and s_2. Right: Scan of the data sheet provided by the manufacturer describing the sensor signal for given distances (x-axis: distance, y-axis: sensor signal). The resolution of the sensor is in the range [10; 80] cm.

Case 1) When the robot moves onto the belt to pick up the rod, this motion does not alter the sensor signal because the rod itself has not moved. In this case the correct position must be searched for. This is usually the case when preparatory sensors are used. The developer can either manually guide the robot to the correct position or use a second sensor to perform an automated search, but it is up to the developer to define a valid search algorithm, because this depends strongly on the task. The search should be kept as simple as possible. When the sensor is calibrated adequately well, the change function's estimate is accurate and always locates the object correctly. So this search is only executed in the very first iterations. Because of this it is not necessary to implement a fast, efficient search strategy, since this represents only a backup strategy in case the change function is still inadequate for a given sensor value. Once the correct position p_{change} has been reached, x_{change} is calculated as

$$x_{change} = p_{change} - p_{ref} \tag{5}$$

and the data tuple (x_{change}, s_{act}) describes a valid data point of f_{real}, because the sensor value has not changed during the search. This tuple is added to a set S describing the current knowledge about f_{real}. With increasing size of S more and more knowledge about f_{real} is collected and the more precise the next estimations will be.

Case 2) This case occurs, when the robot has located the rod and grasped it. Now, a robot motion will alter the sensor signal. In this case a corrective motion can be performed instead of a search. This is usually the case if the sensor is used concurrently. We can employ an automated search; the direction of the search is defined by the Cartesian coordinates that are altered by the sensor. The search terminates when $s_{act} = s_{ref}$. If this value has been reached, the robot has corrected the change. A detailed solution describing the motions involved is described in Section 4.

3.3 Defining the General Program Structure

When defining the program structure, the developer must decide how the adaptation strategy for the change can be integrated into the robot program. This is done at the point when robot movements are executed based on the sensor signal. The robot uses the current sensor value s_{act} and current estimate of the change function f_{est} using S to determine p_{est}.

The key point is to decide whether the robot motion will modify the sensor signal or not. This leads to the following basic program structure:

If a motion does not change the signal, the source code will look similar to this:

```
1 pos = changeFunction();
2 MOVE pos;
3 IF NOT isOkay() THEN {
4       performSearch();
5       pos = getCurrentPosition();  }
6 updateChangeFunction(pos,sensor.value());
```

The robot will calculate and move to the estimated position using the change function by calling the function `changeFunction` (Lines 1 and 2). At this point, a decision must be made if the position is correct, which is either accomplished using a second sensor or by asking the developer to check (Line 3).

If this is not the case, a search is initiated, guiding the robot to the correct position (Lines 4 and 5). Then a new data tuple is added to S improving f_{est} (Line 6) by calling `updateChangeFunction`. This must be called explicitly by the developer to update S with the new, correct position p_{real} for s_{act}.

On the other hand, if a motion does change the signal, the source code will look similar to this:

```
1 DO {
2  pos = getCorrection();
3  MOVE pos;
4 } WHILE sensor.value()!=s_ref;
```

Here, the search is realized using a do/while-loop. We estimate the current change (Line 2) and move the robot accordingly (Line 3) until we have reached the reference position (Line 4). We will describe a suitable method to calculate reasonable correction values in Section 4. Here, it is important that these methods are encapsulated in the function `getCorrection`, so they remain hidden from the developer.

All the developer must do to use these methods is to specify the following parameters:

1) The taskframe and the coordinate(s) in which the change occurs. p_{ref} and s_{ref} are calculated within this taskframe. The default sensor values are recorded when p_{ref} is stored.
2) The sensor used to supervise p_{ref}. This includes a specification of the sensor's signal-to-noise ratio (SNR).
3) A Boolean value specifying if a robot motion will alter the sensor signal. The function `getCorrection` uses this value to determine which estimation method is executed.
4) Furthermore, it makes sense to require all estimates x_{est} to be within a specific range to prevent the robot from leaving the workspace in case of an extreme estimate. However, this may increase the number of corrections necessary to reach p_{ref}.

These four parameters enable the robot to learn a change function adaptively during task execution. All other functionality is independent from the task and is integrated into the function `getCorrection`.

The actual implementation of f_{est} is interchangeable. The calibration data gained by the adaptation is stored in S. It is up to the developer to determine how the tuples in S are used to approximate the function. Any interpolation method can be employed, because no additional knowledge about the function type of f_{est} is necessary. Curve-fitting methods may be used as well, which will lead to a reasonable approximation of f_{est} after fewer executions compared to interpolation methods. But, as is the case with all adaptation and learning methods in general, the more information one has available right from the start, the faster the methods will work adequately.

4 SUPERVISING AND ADAPTING TO CHANGES DURING EXECUTION

In this section, we describe how corrective motions can be executed by the robot using sensor information gained during a movement. All corrective motions are used to supplement the existing knowledge about the change function. We explain how this method can be integrated into a programming environment and kept hidden from the developer.

4.1 Using the Secant Method for Corrective Motions

In principle, it is possible to use a search motion pre-defined by the developer even if the correction has changed the sensor signal, but this discards the information gained by the alteration of the sensor signal during the search. We can use this information to our advantage and generate corrective motions which locate p_{est} faster than a standard search motion.

Since this correction alters the sensor signal, we use it to judge the performed correction and compute subsequent corrections accordingly. Suppose we knew x$_{change}$, the first tuple for S would be (x_{change}, s_{act}). Here, we only know s_{act}, not x_{change}. But x_{change} is simultaneously the offset along the x-axis of (x_{change}, s_{act}) from the root, due to the monotonicity of f_{est}. If we perform multiple corrections until we reach the root, we can compute x_{change} as the sum of all corrections the robot has made. From a mathematical point of view, this is equivalent to finding the root of an unknown function.

The Secant method (Press, 92) is defined by the recurrence relation

$$x_{n+1} = x_n - \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} f(x_n) \qquad (6)$$

where f is an unknown function. As can be seen from the recurrence relation, the Secant method requires two initial values, x_0 and x_1. The values x_n of the Secant method converge to a root of f if the initial values x_0 and x_1 are sufficiently close to the root. The order of convergence is φ, where $\varphi = (1+\sqrt{5})/2 \approx 1.62$ is the golden ratio. In particular, the convergence is superlinear. This result only holds true under some conditions, namely that f is twice continuously differentiable and the root in question is simple and may not be a repeated root. Change functions, as we have defined them, fulfill these conditions.

In our case, the f is the real change function f_{real} and the first value x_0 is simply the change we wish to calculate, x_{change}, while the second value x_1 is the first corrective motion the robot has performed, x_{est}, which is based on the current estimate of the change function f_{est}. Note that f_{est} is used only once for the initial correction, all subsequent corrections are based on the Secant method (see Figure 4) only using the current sensor values provided by f_{real}. Since the convergence of this method is superlinear, we will not need many additional corrections x_n, $n > 1$, should x_1 prove to be poor.

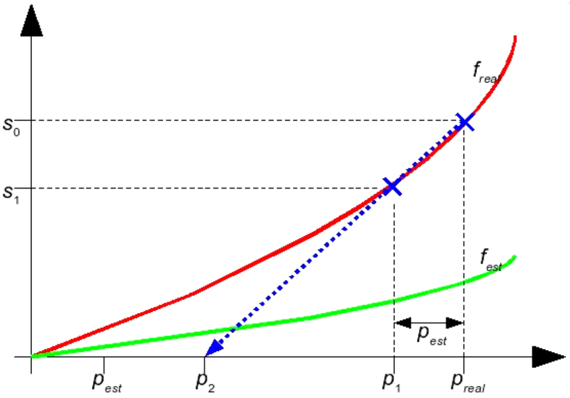


Figure 4: Illustration of the first two steps of the correction algorithm: For a given variation p_{real} we perform an estimated correction p_{est} based on the corresponding sensor value s_0, the real change function f_{real} (red) and our current estimate f_{est} (green). We move the robot to position p_1 and retrieve a new sensor value s_1. We then use the Secant method to grade the last correction and move the robot accordingly to p_2. All subsequent corrections are performed using the Secant method only.

It is important to consider the following: When the next value x_{i+1} is calculated, it must be kept in mind that we have already performed correction x_i before we could measure s_{i+1} to rate x_i. So we must subtract the impact of x_i from x_{i+1}.

Another advantage of this approach is that all corrections x_i and corresponding sensor values $s_i = f_{real}(x_{i-1})$ are known. We can store these as pairs (x_i, s_i) in a temporary stack. When we have reached p_{ref}, we can use this information to create multiple new data tuples for S. If we have performed i corrections until the robot reaches p_{ref}, the topmostpair (x_i, s_i) on the stack already describes a valid data tuple for S. The next pair on the stack (x_{i-1}, s_{i-1}) describes a correction to p_{ref} altered by x_i. So ($x_i + x_{i-1}$, s_{i-1}) is another valid data tuple for the set. Subsequent processing of the stack provides us with a valid data tuple for every correction performed, so we add i new data tuples to S. This leads to an accurate approximation of f_{est} after fewer executions compared to the addition of only one tuple to S in every execution.

The Secant method only works for one-dimensional functions. It is possible to combine multiple sensors to obtain an n-dimensional signal. In this case, the Broyden method (Broyden, 65) can be used, which is similar to the Secant method.

This method is only applicable if a robot motion alters the sensor signal, as is described in Case 2 in Section 3.2. In the first case of that section, there is no other option as to use either a manual guidance method or an automated search.

4.2 Possible Utilization of other Approaches

The Secant method is not the only method to determine the root of a function. Some other methods are Newton's method, fixed point iteration, and the bisection method. We will now compare the Secant method with these and show why the Secant method is the best choice for this task.

Newton's method and fixed point iteration both use the derivative of the function to calculate the next correction. But, as we have explained in Section 1, it is not always possible to find an analytical solution. Additionally, if this solution was known, it would be more sensible to record a number of examples before setting up the main program and use the examples to determine the function parameters.

The bisection method does not rely on the function's derivative, but has another drawback: To find the root of a function f in an interval [a, b], both f(a) < 0 and f(b) > 0 must hold, or vice versa. If both values are negative or positive, this method cannot be employed. This is a serious drawback for this case, since we cannot ensure that the first correction we have performed will result in a new sensor value which has the inverse sign of the first value.

In summary, we can say that to our knowledge the Secant method is the only applicable method that enables a robot to perform a series of corrective motions without any need for backtracking until the root of an unknown change function is reached.

5 EXPERIMENTS

In this section, we show the validity of our approach and explain the interaction of all components described in Sections 3 and 4.

We have implemented the task described in Section 3.2. The sensors used are distance sensors GP2D12 made by SHARP with a measurement range of [10; 80] cm. The first sensor supervises the position where the robot is supposed to pick up the rod and measures the translation along the x-axis. The second is located 44 cm away from the first along the y-axis of the belt (Figure 3, left). The difference between the two sensor values describes the rotation around the z-axis.

The data sheet for the sensors shows that the sensor signal is not linear with respect to the physical distance (Figure 3, right), so it is not possible to use a simple linear conversion to determine the translation or the rotation of the rod. In theory, the change function describing the rotation can be derived as an Arcus-Tangens function, but the parameters for this function are unknown. Therefore, the robot shall learn both functions adaptively during task execution. A reference position pref is set up (Figure 3, left), describing the ideal position and orientation the rod should have. This position would be identical with the position of the rod in case a feeding mechanism is employed. It is important to measure the sensor values for p_{ref} as well. Later on, all measurements are compared against these values and if the difference exceeds the SNR of the sensor in question, a change is recognized. The developer now sets up two mappings describing the changes (Table 1).

The robot program for a single task execution is now short and relatively simple:

```
1 PROGRAM pickupRod() {
2   offset$_{est}$ = getCorrection(Distance);
3   MOVE offset$_{est}$;
4   IF (force$_{z-axis}$() < force$_{contact}$) THEN
5      searchRod();
6   update(Distance, HERE);
7   graspRod();
8   MOVE $p_{ref}$;
9   DO
10     rotation$_{est}$ = getCorrection(Rotation);
11     MOVE rotation$_{est}$;
12  WHILE (rotation$_{est}$ != $p_{ref}$)
13  MOVE $p_{dropoff}$;
14  releaseRod();
15 }
```

Table 1: Change function mappings used for the experiment.

	Distance	**Rotation**
Position	p_{pickup}	p_{pickup}
Dimension	Translation along *x*	Rotation around *z*
Sensors	Sensor1	Sensor1 - Sensor2
SNR of Sensor	5.0	10.0
Movement modifies sensor signal	FALSE	TRUE
Range of Correction	[-240; 240] mm	[-10; 10] mm

In Lines 2 and 3 the function `getCorrection` receives a reference to a mapping structure defined in Table 1 as parameter and moves the robot to the estimated position of the rod. We use a force/torque sensor to check whether the rod was grasped correctly (Line 4). If this is not the case, we employ a basic search motion probing the conveyor belt in fixed intervals for the rod (Line 5). When the rod is located, we manually update *S*, grasp the rod and move it to the reference position (Lines 6 to 8). At this point the rod may still be rotated by an unknown amount. In Lines 9 to 12 we correct this rotation by repeatedly calling `getCorrection` until the reference position is reached. Then we move the rod to $p_{dropoff}$ and release it (Lines 13 and 14). Note that the program itself does not contain any sensor data processing. Additionally, it is neither necessary for the developer to determine the type of the change functions nor any parameters for these functions. To calculate the Cartesian change for an unknown sensor value, we use a simple linear interpolation over all data tuples in *S*.

We executed the program 100 times. Every time the translation and rotation of the rod was chosen randomly. The initial estimate of both change functions was deliberately chosen badly as a bisecting line (Figure 5). For the change function describing the distance of the rod, we could have also created data tuples using the data sheet of the sensor (Figure 4, right). We have chosen not to do this, for two reasons: Firstly, the data tuples would have to be measured manually by the developer in the figure and modified by the distance of the rod's default position, which is a cumbersome task. Secondly, the data sheet is rather small and the resolution is low so it is difficult to determine exact values. Here, it is easier to just use a bad approximation for the very first executions, because this will change after a few executions. Because of this, the robot was unable to grasp the rod correctly during the very first executions and also needed multiple corrections to compensate the rod's rotation. After 10 executions the estimations of the change function look similar to the one in Figure 3, and an Arcus-Tangens function respectively (Figure 6). After 100 executions we obtained a precise interpolation of both change functions (Figure 7), allowing the robot to grasp the rod 20 out of 20 times (100%) without the need for a search motion. The rotation was corrected successfully with just one rotation in 14 out of 20 cases (75%). In the other cases, the robot had to perform more than one rotation to align the rod correctly.

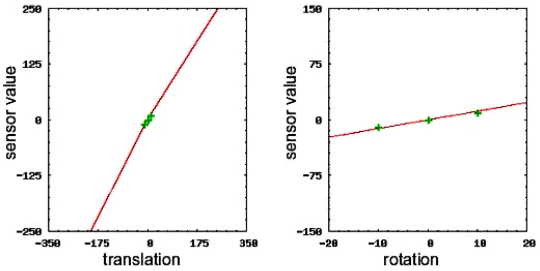


Figure 5: Initial estimates of the change functions used to compute the translation (left) and rotation of the rod (right).

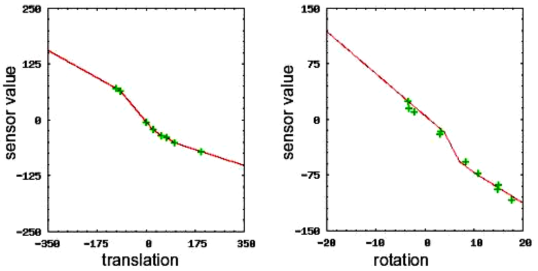


Figure 6: Estimates of the change functions used to compute the translation (left) and rotation of the rod (right) after 10 executions.

The accuracy of the estimated change functions in locating and rotating the rod during the adaptation process is shown in Figures 8 and 9. In both figures we show whether the robot was able to grasp the rod and rotate it correctly using the estimates of the change functions (red). A value of 0 means that the robot had to search for the rod or perform multiple rotational corrections, respectively, while a value of 1 means that the estimate was correct. The green lines show the overall accuracy of the robot over all task executions up to that point, while the blue lines show the accuracy over the last 20 executions. We can see that the robot was capable of grasping the rod correctly nearly all the time after 50 executions,

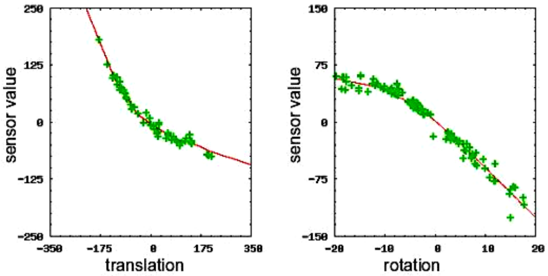


Figure 7: Estimates of the change functions used to compute the translation (left) and rotation of the rod (right) after 100 executions.

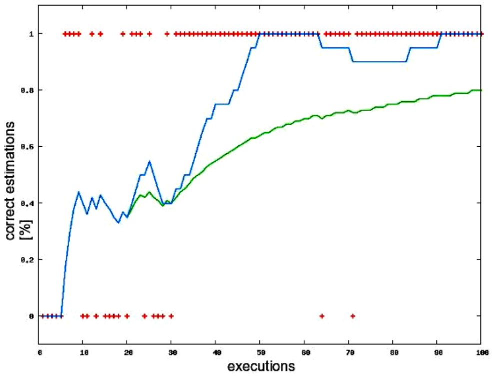


Figure 8: Overall (green) and averaged (blue) percentage of correct estimations of the rod's translation on the conveyor belt using the change function for 100 executions. A red dot with a value of 0 indicates that the robot could not locate the rod with the given change-function, but had to perform a search instead. A value of 1 indicates that the rod was found without the need for a search motion.

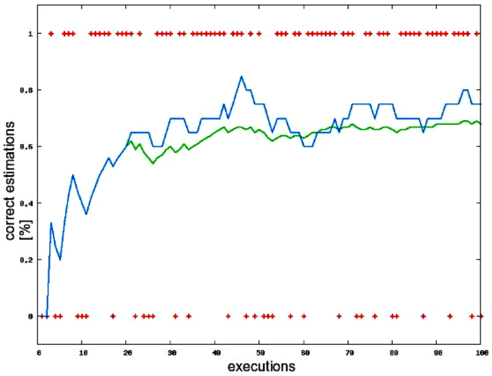


Figure 9: Overall (green) and averaged (blue) percentage of correct estimations of the rod's rotation on the conveyor belt using the change function for 100 executions. A red dot with a value of 0 indicates that the first correction of the rotation did not align the rod perfectly and further corrections were necessary. A red dot with a value of 1 indicates that the rod was aligned correctly with only one motion.

and had an overall accuracy of 80%. Due to the fact that two sensors are necessary to measure the rotation, the SNR of this combined sensor is relatively high, so the correction could not be performed in one motion every time. In spite of this, the robot was still capable of performing a perfect correction in 75% of all cases.

6 CONCLUSIONS

The aim of this work is to enable a developer to easily employ external sensors for flexible robot programs. The focus of this study was to show that data tuples describing the connection between sensory data and positional variations can be acquired automatically by the robot independent of the task and without the need for intricate calculations by the developer. We have presented a method to determine this data online during multiple executions of the task. The intention was to keep the requirements and methods independent from the type of sensor and make them universally applicable so they can be easily incorporated into a robot program. Finally, we presented an experiment to validate our research. We showed that it is possible to employ the proposed methods to successfully determine two change functions for a pick-and-place task.

In the next step our aim is to integrate time stamps into the data set *S*. Then we are able to deal with drifts in the sensor data due to heating processes of the sensor itself by discarding the older data tuples which do not reflect the current state of the system any more.

REFERENCES

Adams, M., “Sensor Modelling, Design and Data Processing for Autonomous Navigation”, World Scientific Publishing, 1998, ISBN 9810234961.

Bolles, B., Bunke, H., Christensen, H., Noltemeier, H., “Modelling and Planning for Sensor-Based Intelligent Robot Systems”, Seminar on, Schloß Dagstuhl, 1998, http://www.dagstuhl.de/Reports/98391.pdf.

Broyden, C.G., “A Class of Methods for Solving nonlinear Simultaneous Equations”, Mathematics of Computation, Vol. 19, No. 92. (Oct., 1965), pp. 577-593, Jstor.

Chhatpar, S.R., Branicky, M.S. “Localization for robotic assemblies with position uncertainty”. Proc. IEEE/RSJ Intl. Conf. Intelligent Robots and Systems, Las Vegas, NV, October, 2003.

Deiterding, J., Henrich, D. “Acquiring Change Models for Sensor-Based Robot Manipulation”, Int. Conf. o. Robotics and Automation 2008.

Dong, M., Tong, L., Sadler, B.M., “Information retrieval and processing in sensor networks: deterministic scheduling vs. random access”, Proc. o.t. Int. Symp. on Information Theory, 2004. ISIT, pages 79 – 85.

Duda, R., Hart, P. and Stork, D., “Pattern Classification”, Wiley & Sons, 2000, ISBN 0471056693.

Dudek, G., Zhang, C. “Vision-based robot localization without explicit object models” Int. Conf. On Robotics and Automation, 22-28 Apr 1996, ISBN 0-7803-2988-0, pages 76-82 vol.1.

Firby, R.J. “Adaptive execution in complex dynamic worlds”, Dissertation, Yale university, 1989, www.uchicago.edu/users/firby/thesis/thesis.pdf.

Hager, G. “Task-Directed Sensor Fusion and Planning: A Computational Approach”, Springer, 1990, ISBN 079239108X

Hutchinson, S.A., Cromwell, R.L. and Kak, A.C., “Planning sensing strategies in a robot work cell with multi-sensor capabilities”, in. Proc. IEEE Int. Conf. On Robotics and Automation, 1988, pages 1068-1075.

Kriesten, D., Rößler, M., et al., “Generalisierte Plattform zur Sensordatenverarbeitung”, Dresdner Arbeitstagung Schaltungs- und Systementwurf, 2006, http:// www.eas.iis.fhg.de/events/workshops/dass/2006/dassprog/pdf12_kriesten.pdf.

Leonhardt, U., Magee, J., “Multi-sensor location tracking”, Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking, Dallas, USA, 1998, ISBN 1-58113-035-X, pages: 203 – 214.

Paragios, N., Tziritas, G.. “Adaptive Detection and Localization of Moving Objects in Image Sequences” Signal Processing: Image Communication, 14:277-296, 1999.

Pfeifer, R., Scheier, C., “From perception to action: The right direction”, Proc. “From Perception to Action”Conference, IEEE Computer Society Press, Los Alamitos, 1994, pages = "1-11".

Press, W.H., Flannery, B.P., Teukolsky, S.A., Vetterling W.T. "Secant Method, False Position Method, and Ridders' Method." §9.2 in Numerical Recipes in FORTRAN: The Art of Scientific Computing, 2nd ed. Cambridge, England: Cambridge University Press, pp.347-352, 1992.

Rui, K., Yoshifumi, M., Satoshi, M., “Information Retrieval Platform on Sensor Network Environment”, IPSJ SIG Technical Reports, 2006, No. 26, ISSN 0919-6072, pages 37-42.

Thomas, U., Movshyn, A., Wahl, F., “Autonomous Execution of Robot Tasks based on Force Torque Maps”, Proc. o. t. Jnt. Conf. on Robotics. 2006, Munich, Germany, May 2006.

Wheeler, M. “Automatic modeling and localization for object recognition”, Carnegie Mellon University, Computer Science Technical Report CMU-CS-96-118, 1996.

HUMANOID REALISTIC SIMULATOR
The Servomotor Joint Modeling

José L. Lima[1], José A. Gonçalves[1], Paulo G. Costa[2] and A. Paulo Moreira[2]
[1]*Polytechnic Institute of Bragança, Bragança, Portugal*
[2]*Faculty of Engineering of University of Porto, Porto, Portugal*
jllima@ipb.pt, goncalves@ipb.pt, paco@fe.up.pt, amoreira@fe.up.pt

Keywords: Humanoid, Servomotor, Modeling, Simulation.

Abstract: This paper presents a humanoid servomotor model that can be used in simulations. Once simulation is a tool that reduces the software production time, it was developed a realistic simulator that own the humanoid features. Based on a real platform, the simulator is validated when compared with the reality.

1 INTRODUCTION

Recent research in biped robots has resulted in a variety of prototypes that resemble their biological counterparts. Legged robots have several advantages, they can move in rugged terrains, they have the ability to choose optional landing points, and two legged robots are more suitable to move in human environment (Suzuki and Ohnishi, 2006).

The simulator should capture the essential characteristics of the real system. In this paper, the servomotor model that powers the real humanoid joints is addressed. The model of a Dynamixel AX-12 servomotor and its characteristics are found by an iterative method based on a realistic simulator, the *SimTwo* (Costa, 2009).

There are several simulators with humanoid simulation capability, like Simspark, Webots, MURoSimF, Microsoft Robotics Studio, YARP: Yet Another Robot Platform (Wang et al., 2006) and OpenHRP3 (Ope, 2009), meanwhile, the *SimTwo*, as a generic simulator, allows to simulate different types of robots and allows the access to the low level behaviour, such as dynamical model, friction model and servomotor model in a way that can be mapped to the real robot, with a minimal overhead. This simulator deals with robot dynamics and how it reacts for several controller strategies and styles. Using a realistic simulator can be the key for reducing the development time of robot control, localization and navigation software. It is not an easy task to develop such simulator due to the inherent complexity of building realistic models for the robot, its sensors and actuators and their interaction with the world (Browning and Tryzelaar, 2003).

The purpose of developing such simulator is to produce a personalized and versatile tool that will allow the development and validation of the robot's software thereby reducing considerably the development time.

The paper is organized as follows: Initially, the real robot (which is the basis of the simulator) and its main control architecture are presented. Then, section 3 presents the developed simulator where the servomotor model was developed. Further, section 4 presents the validation of the simulator by comparing its results with the real robot. Finally, section 5 rounds up with the conclusions and future work.

2 REAL HUMANOID

There are several humanoid robots kits available. The commercially available Bioloid robot kit, from Robotis, is the basis of the used humanoid robot and the overview of the proposed biped robot is shown in Figure 1.

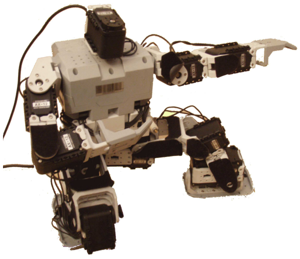
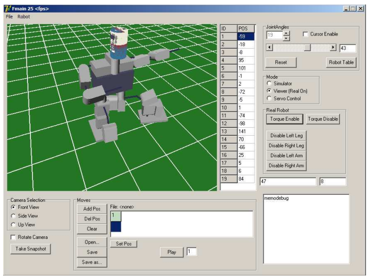


Figure 1: Real humanoid robot (from Bioloid).

The servo motors are connected to the central processing unit through a serial 1Mbps network. Next subsection presents the physical robot in which the developed humanoid simulator was based.

2.1 Main Architecture

The presented humanoid robot is driven by 19 servo motors (AX-12): six per leg, three in each arm and one in the head. Three orthogonal servos set up the 3DOF (degree of freedom) hip joint. Two orthogonal servos form the 2DOF ankle joint. One servo drives the head (a vision camera holder). The shoulder is based on two orthogonal servos allowing a 2DOF joint and elbow has one servo allowing 1DOF. The total weight of the robot (without camera and on board computer) is about 2 kg and its height is 38 cm.



ID	POS
1	-59
2	-18
3	-8
4	95
5	101
6	-1
7	2
8	-72
9	-5
10	1
11	-74
12	-98
13	141
14	70
15	-66
16	25
17	5
18	6
19	84

Mode: ● Viewer (Real On)

Camera Selection: ● Front View

Figure 2: Real humanoid robot control application.

To control the real humanoid robot a high level application (presented in Figure 2) was developed. This application is independent from the simulator, although it allows to acquire and share some real robot data with simulator.

3 SIMULATION MODEL

Designing the robot's behaviour without real hardware is possible due to a physics-based simulator implementation. The physics engine is the key to make simulation useful in terms of high performance robot control (Browning and Tryzelaar, 2003). The dynamic behaviour of robot (or multiple robots) is computed by the ODE (Open Dynamics Engine), a free library for simulating rigid body dynamics.

3.1 Simulator Architecture

The simulator architecture is based on the real humanoid robot. The body masses and dimensions are used to build a humanoid simulator similar to the real one. The communication architecture in real robot brings some limitations to control loop such as lag time. The developed simulator enhances these properties. The same architecture levels of the real robot are implemented in the simulator. At the lowest level, the servo motor model includes the control loop, just like the real servomotors. At the highest level, some predefined joint states are created based on several methods presented on literature (Kajita et al., 2006) and (Zhang et al., 2008). At the middle level, an optimized trajectory controller that allows to minimize the energy consumption is introduced as presented in Figure 3 (Lima et al., 2008b).

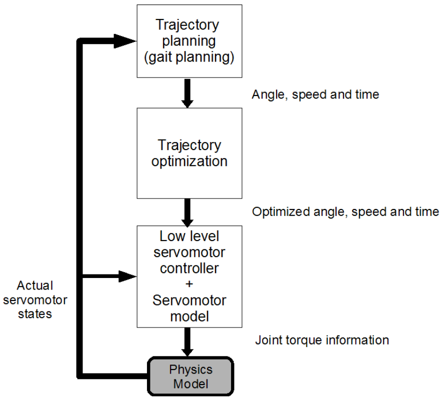

Trajectory planning (gait planning)
Angle, speed and time
Trajectory optimization
Optimized angle, speed and time
Low level servomotor controller + Servomotor model
Actual servomotor states
Joint torque information
Physics Model

Figure 3: Simulator main architecture.

Next subsections describe the servomotor model that is used in the simulator where the electrical, friction and controller models are presented.

3.2 DC Motor Model

The servomotor can be modeled by a DC motor model, presented in Figure 4, where U_a is the converter output, R_a is the equivalent resistor, L_a is the equivalent inductance and e is the back emf voltage as expressed by equation 1 (Conceição et al., 2006).

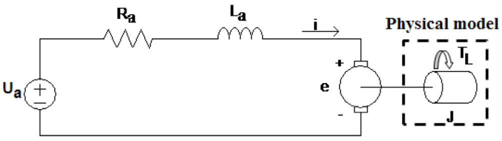


Figure 4: DC motor electric model.

$$U_a = e + R_a i_a + L_a \frac{\partial i_a}{\partial t} \qquad (1)$$

The motor can deliver a T_S torque and its load has a J moment of inertia that will be presented by the physical model ODE. Current i_a can be related with developed torque T_D through equation 2 and the back emf voltage can be related with angular speed through equation 3, where K_s is a motor parameter that can be found by an experimental setup as presented in subsection 3.3 (Bishop, 2002).

$$T_D(t) = K_s i(t) \tag{2}$$

$$e(t) = K_s \omega(t) \tag{3}$$

In fact, the real developed torque (useful) that will be applied to the load (T_S) is the motor torque subtracted by the friction torque (T_F) as presented in equation 4. The friction torque is discussed in subsection 3.5.

$$T_S = T_D - T_F \tag{4}$$

3.3 DC Motor Model Measurements

It was used the AX-12 servomotor from Dynamixel as the base of the humanoid simulator articulations. The R_a and L_a values can be directly measured (R_a=8 Ω and L_a= 5 mH). The K_s motor parameter can be found by an indirect estimation. For several angular speeds, it can be measured the emf voltage while motor is open circuit.

Figure 5 shows the graphical data of the K_s line and its trend line. The average value of 13 measures for K_s (line slope) is about 0.006810 $V.s/rad$.

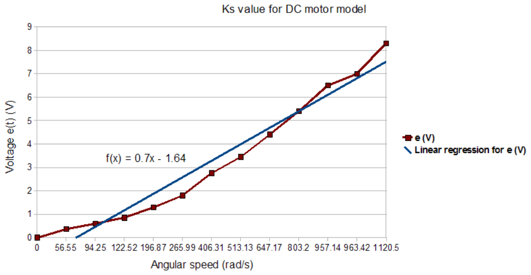


Figure 5: K_s value for DC motor model.

3.4 DC Motor Nonlinearities

In a way to better model the real system, where variables cannot assume all values, there must be some limits applied to some quantities. The first one, is the voltage applied to the supply terminals U_a. This voltage is limited by the batteries voltage. Further, current i is limited by the drive electronics once it is related to the torque through equation 2 (torque limit is programmed in the real servomotor). Current gradient is also implicitly limited by the presence of L_a. Figure 6 shows the block diagram computed by the simulator. There is also a internal gradient limitation to ensure some numerical stability specially when the integration step period is significantly slower than electric dynamic.

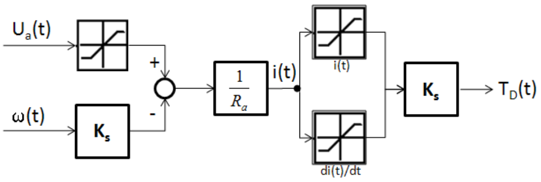


Figure 6: Servomotor model block diagram computation.

3.5 Friction Model

The friction model has two terms: the static and dynamic friction as presented in equation 5.

$$T_F = F_c sign(\omega) + B_v \omega \tag{5}$$

The first one can be modeled as the sign function (with Fc constant) and the second one can be modeled as a linear function with slope Bv.

The sum of this two components (T_F), the final friction model, is shown in figure 7.

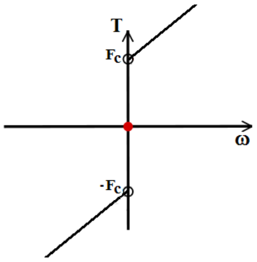


Figure 7: Servomotor model nonlinearities.

Figure 8 shows the friction model implemented in the servomotor model.

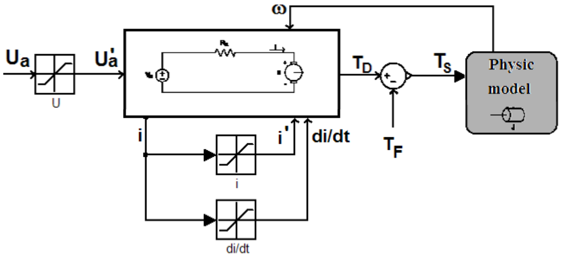


Figure 8: Servomotor model with friction model.

The F_c and B_v constants are found using simulator scanning several possible values minimizing the error with the real system during an arm fall from 90 to 0 degrees. The error surface function can be minimized.

As result, Bv=0.01278 $N.m.s/rad$ and Fc=0.0000171 $N.m$ where found as the best values. These constants allow the simulator to follow reality very closely as presented in figure 9 where an arm falls from 45, 90 and 135 degrees for both robots.

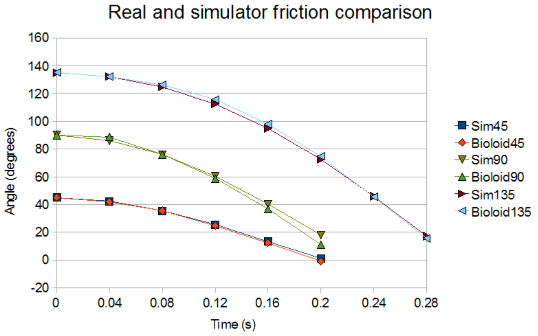


Figure 9: Real robot and simulator friction comparison.

3.6 Low-level Controller

The low level controller resembles the closed loop controller in the real robot implemented by the servomotor manufacturer. This controller accepts, from the higher level, the angle and angular speed references that define a trajectory. The controller type present in the Dynamixel isn't specified. However, there are some possible models for the controller, such as PID or state feedback. Considering a state feedback controller and assuming K_θ^i the position error gain and K_ω^i the speed error gain for each i joint, the equation 6 defines the servomotor input (U_a) that keeps the desired reference conditions for each i joint. The output torque is computed by the physic model as presented in Figure 10.

$$U_a^i(t) = K_\theta^i(\theta_{ref}^i(t) - \theta^i(t)) + K_\omega^i(\omega_{ref}^i(t) - \omega^i(t)) \tag{6}$$

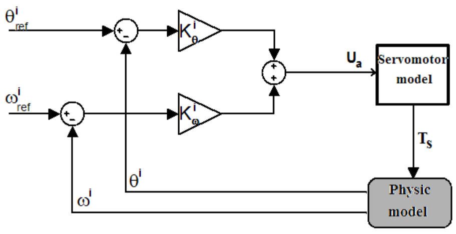


Figure 10: Low level controller.

The position error and the speed error gains can be computed resorting to a least squares approach. Having the real robot arm step response (from 0 to 90 degrees), it is possible to scan several gains and to determine the quadratic error between the real servo and the humanoid simulator joint for each solution. The one that fulfill the lowest quadratic error is the chosen gains to the controller. Figure 11 shows the quadratic error for the several solutions.

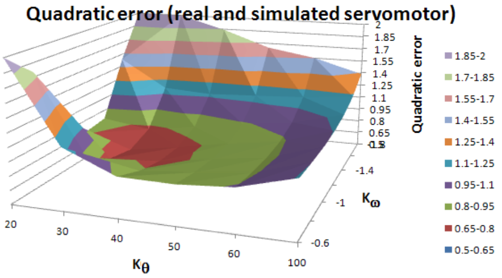


Figure 11: Quadratic error for real and simulator servomotor.

The lowest quadratic error (0.66 $degrees^2$) occurs for K_θ^i equals 30 and for K_ω^i equals -1.2. These gains allow to obtain the step response very close to the real servo. Figure 12 shows both step responses.

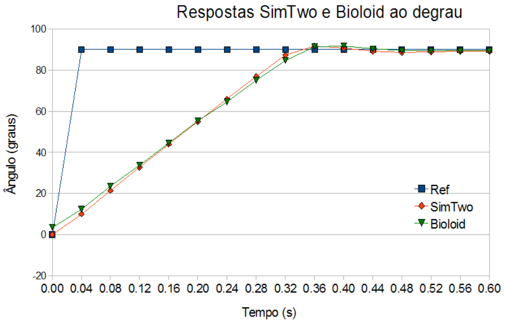


Figure 12: Real and simulator servomotor step response.

3.7 Servomotor Model

With the previous results, it is possible to create the servomotor model including the DC motor, the friction and the controller models (with its nonlinearities) that will represent the real servo motor in the simulator. The final model is presented in figure 13.

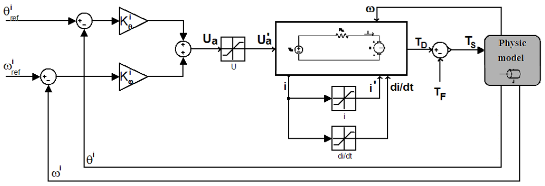


Figure 13: Simulator servomotor model.

4 SIMULATOR VALIDATION

To validate the humanoid simulator model it is required to implement the same control signal to both robots and to analyze the behaviors. Predefined trajectory states, that allow robot to walk, are based on the Zero Moment Point (ZMP) method. Figure 14 shows the sequence during walk movements for both robots (real at left and simulator at right). It is possible to observe that both robots exhibit very similar behaviours.

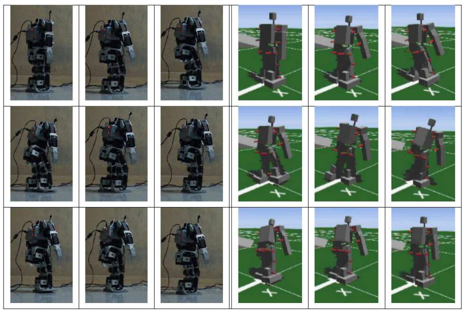


Figure 14: Real and simulator robots walking with the same predefined gaits.

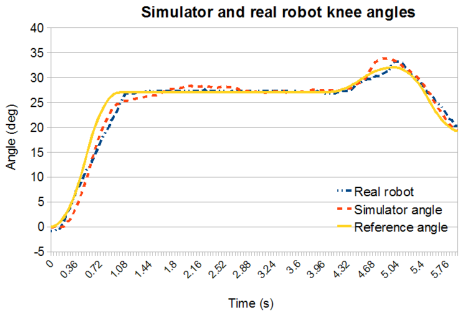


Figure 15: Real and simulator robots knee angles during a walk movement.

With this walking movement, it can be acquired all joint angles for both robots. Figure 15 shows a knee angle for real and simulated robots, in a walk movement, that shows simulator behaves as real robot. Moreover, the power consumption comparison between real and simulated robot is presented in (Lima et al., 2008a).

5 CONCLUSIONS AND FUTURE WORK

A simulator that allows a humanoid robot simulation capability is addressed and validated. The joints that emulate the real articulations are based on a realistic servomotor model. The proposed servomotor model was implemented in the developed simulator, *SimTwo*. This simulator is based in a real platform. The friction model and closed loop controller gains are found based on the real robot behaviour. It allows to search the optimal values for friction and controller gains based on a heuristic approach. The validation with the real humanoid robot allows to confirm the proposed servomotor model. As future work, the simulator can be useful to find several parameters that optimize a desired condition such as energy consumption in the walk movement and further applied to the real robot.

REFERENCES

(2009). Open architecture humanoid robotics platform. http:// www.openrtp.jp/openhrp3/en/index.html.

Bishop, R. (2002). *The Mechatronics Handbook*. CRC Press, New York.

Browning, B. and Tryzelaar, E. (2003). Ubersim: A realistic simulation engine for robotsoccer. In *Proceedings of Autonomous Agents and Multi-Agent Systems, AAMAS'03*.

Conceição, A., Moreira, A., and Costa, P. (2006). Dynamic parameters identification of an omni-directional mobile robot.

Costa, P. (2009). Simtwo webpage. http://www.fe.up.pt/˜paco/wiki/.

Kajita, S., Morisawa, M., Harada, K., Kaneko, K., Kanehiro, F., Fujiwara, K., and Hirukawa, H. (2006). Biped walking pattern generator allowing auxiliary zmp control. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2994–2999.

Lima, J., Gonçalves, J., Costa, P., and Moreira, A. (2008a). Realistic behaviour simulation of a humanoid robot. In *8th Conference on Autonomous Robot Systems and Competitions*.

Lima, J., Gonçalves, J., Costa, P., and Moreira, A. (2008b). Realistic humanoid robot simulation with an optimized controller: a power consumption minimization approach. In *11th. International Conference on Climbing and Walking Robots*, pages 1242–1248.

Suzuki, T. and Ohnishi, K. (2006). Trajectory planning of biped robot with two kinds of inverted pendulums. In *Proceedings of 12th International Power Electronics and Motion Control Conference*, pages 396–401.

Wang, X., Lu, T., and Zhang, P. (2006). Yarp: Yet another robot platform. *International Journal of Advanced Robotic Systems*.

Zhang, L., Zhou, C., and Xiong, R. (2008). A lie group formulation for realtime zmp detection using force/torque sensor. In *Proceedings of the 11th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines*, pages 1250–1257.

DOUBLE WELL POTENTIAL AS DIFFUSIVE FUNCTION FOR PDE-BASED SCALAR IMAGE RESTORATION METHOD

A. Histace
ETIS UMR CNRS 8051, ENSEA-UCP, 6 avenue du Ponceau, 95014 Cergy, France
aymeric.histace@ensea.fr

M. Ménard
L3i, University of La Rochelle, Pole Sciences et Technologie, 17000 La Rochelle, France
michel.menard@univ-lr.fr

Keywords: Image Diffusion, Double well potential, Directional diffusion, Selectivity.

Abstract: Anisotropic regularization PDE's (Partial Differential Equation) raised a strong interest in the field of image processing. The benefit of PDE-based regularization methods lies in the ability to smooth data in a nonlinear way, allowing the preservation of important image features (contours, corners or other discontinuities). In this article, we propose a PDE-based method restoration approach integrating a double-well potential as diffusive function. It is shown that this particular potential leads to a particular regularization PDE which makes it possible integration of prior knowledge about the gradients intensity level to restore. As a proof a feasibility, results of restoration are presented both on ad hoc and natural images to show potentialities of the proposed method.

1 INTRODUCTION

Since the pioneering work of Perona-Malik (Perona and Malik, 1990), anisotropic regularization PDE's raised a strong interest in the field of image processing. Many regularization schemes have been presented so far in the literature, particularly for the problem of scalar image restoration (see (Histace and Ménard, 2008) for a complete review). In (Deriche and Faugeras, 1996) authors propose a synthetic formulation to express the global scheme of PDE-based restoration approaches. More precisely, if we denote $\psi(\mathbf{r},t) : \mathbb{R}^2 \times \mathbb{R}^+ \rightarrow \mathbb{R}$ the time intensity function of a corrupted image $\psi_0 = \psi(\mathbf{r},0)$, the corresponding regularization problem of ψ_0 is equivalent to the minimization problem described by the following PDE:

$$\frac{\partial\psi}{\partial t} = c_\xi(\|\nabla\psi\|)\frac{\partial^2\psi}{\partial\xi^2} + c_\eta(\|\nabla\psi\|)\frac{\partial^2\psi}{\partial\eta^2} \quad , \quad (1)$$

where $\eta = \nabla\psi/\|\nabla\psi\|$, $\xi \perp \eta$ and c_ξ and c_η are two weighting functions (also called diffusive functions). This PDE can be interpreted as the superposition of two monodimensional heat equations, respectively oriented in the orthogonal direction of the gradient and in the tangential direction: It is characterized by an anisotropic diffusive effect in the privileged directions ξ and η allowing a non-linear denoising of scalar image. Eq. (1) is of primary importance, for all classical methods can be expressed in that global scheme: For instance, if we consider the former anisotropic diffusive equation of Perona-Malik's (Perona and Malik, 1990) given by

$$\frac{\partial\psi}{\partial t} = div\left(c(\|\nabla\psi\|)\nabla\psi\right) \ , \quad (2)$$

with $\psi(\mathbf{r},0) = \psi_0$ and $c(.)$ a monotonic decreasing function, it is possible to express it in the global scheme of Eq. (1) with

$$\begin{cases} c_\xi = c(\|\nabla\psi\|) \\ c_\eta = c'(\|\nabla\psi\|).|\nabla\psi| + c(\|\nabla\psi\|) \end{cases} \quad (3)$$

Formulation of Eq. (1) is also interested, for it makes stability study of classical proposed methods possible. What we proposed in this article is a prospective study for the integration of a double well potential as a diffusive function $c(.)$ in Eq. (2). Our aim and motivation for such a study are mainly to show that, firstly, such a choice can lead to a stable PDE-based approach for scalar image denoising that can overpass classical approach of Perona-Malik's from which it is derived and which presents instability problems as formerly shown in (Catté et al., 1992), and, secondly, that this approach overcomes some drawbacks of the classical methods like corner smoothing or pinhole effect. Layout of this article is the following one:

In section two, we introduce the double well function and derive the corresponding PDE in the global scheme of Eq. (1). We also proposed in that section a study of the stability of the derived PDE compared to the stability of Perona-Malik's approach. Third section is dedicated to experimental and quantitative results. Fourth and last section contains conclusion and discussion.

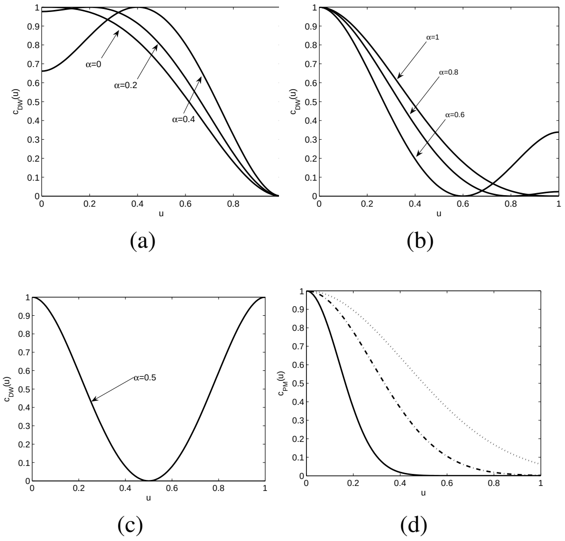
2 DOUBLE WELL POTENTIAL AND CORRESPONDING PDE

2.1 Diffusive Function

The double well potential considered in this article is defined by the following function:

$$c_{DW}(u) = 1 - \phi(u) = 1 - \int_0^u v(\alpha - v)(v-1)dv\ . \quad (4)$$

Some graphical representations of Eq. (4) for different values of α are proposed Fig. 1.



(a) (b)

(c) (d)

Figure 1: (a), (b), (c) Plots of function $c_{DW}(.)$ of Eq. (4) for different values of α: (a) $0 < \alpha < 0.5$, (b) $0.5 < \alpha < 1$, and (c) $\alpha = 0.5$. (d) Plots of function $c_{PM}(.)$ of Eq. (5) for different values of k and α. Solid lines stand for $k = 0.2$, dash-dotted lines for $k = 0.4$, and dotted lines for $k = 0.6$.

This function has to be compared with the classical Perona-Malik's function $c_{PM}(.)$ given by:

$$c_{PM}(u) = e^{-\frac{u^2}{k^2}}\ , \quad (5)$$

with k a soft threshold defining selectivity of $c_{PM}(.)$ regarding values of image gradients. Fig. 1.(d) shows graphical representations of $c_{PM}(.)$ defined by Eq. (5) for different values of k. As one can notice on Fig. 1.(d), for $\|\nabla\psi\| \to 0$, $c_{PM}(\|\nabla\psi\|) \to 1$, whereas for $\|\nabla\psi\| \to 1$, $c_{PM}(\|\nabla\psi\|) \to 0$. As a consequence, boundaries within images which are on a threshold, function of k, are preserved from the smoothing effect of Eq. (2). One can notice on Fig. 1 that $\phi(.)$ has been normalized. As a consequence, we are able to ensure that $0 \leq c_{DW}(u) \leq 1$ for all values of u like classical PM's function of Eq. (2). Global variations of c_{DW} can be compared to those of c_{PM} for $\alpha = 0$ and $\alpha = 1$. For $0 \leq \alpha < 1$, since c_{DW} is issued from a double well potential, selectivity of Eq. (2) is more important and centered on a particular gradient value function of α. For instance, for $\alpha = 0.5$, only gradients of value 0.5 are totally preserved from the diffusive effect that can be interpreted as an integration of directional constrains within the restoration process. Moreover, we are now going to show, that integration of c_{DW} as diffusive function leads to interesting stability property of corresponding PDE.

2.2 Study of Stability

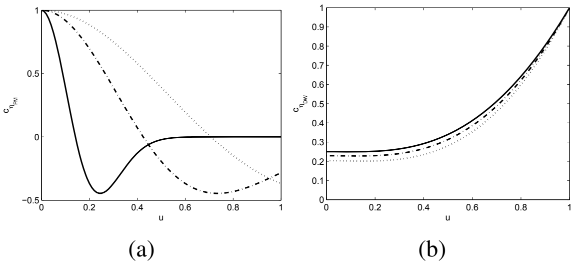
As mentioned in first section, classical Perona-Malik's PDE presents instability problems. More precisely, as shown in (Catté et al., 1992), sometimes noise can be enhanced instead of being removed. This can be explained considering Eq. (3). If we consider $c_{PM}(.)$ function, it appears that corresponding c_η function of Eq. (3), in the global scheme of Eq. (1), can sometimes takes negative values (see Fig. 2.(a) for illustrations). This leads to local instabilities of the Perona-Malik's PDE which degrades the processed image instead of denoising it. Now, if we calculate mathematical expression of c_η with $c(.) = c_{DW}(.)$ of Eq. (4), one can obtain that:

$$c_\eta(\|\nabla\psi\|) = c'_{DW}(\|\nabla\psi\|).|\nabla\psi| + c_{DW}(\|\nabla\psi\|)\ , \quad (6)$$

Considering Eq.(6), if we plot this function, one can notice that corresponding c_η function never takes negative values (see Fig. 2.(b) for illustrations): Diffusive process remains stable for all gradient values of processed image which is of primary importance.

3 EXPERIMENTAL RESULTS

We propose in this section to make a visual and quantitative comparison between classical Perona-Malik's PDE of Eq. (2) with diffusive function $c(.) = c_{PM}(.)$ of Eq. (5), and proposed derived PDE with $c(.) = c_{DW}(.)$ of Eq. (4) as diffusive function. For quantitative comparisons, we will consider adapted measure of similarity between non corrupted initial image and



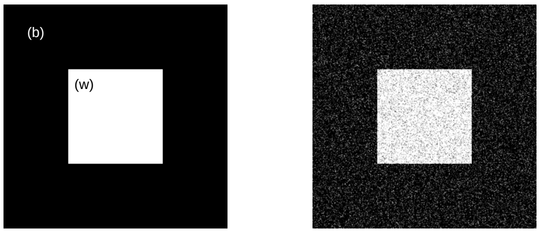
(a) (b)

Figure 2: Plots of function $c_{\eta_{PM}}$ and $c_{\eta_{DW}}$ for different values of k and α. Solid lines stand for $k = 0.2$ and $\alpha = 0.5$, dash-dotted lines for $k = 0.4$ and $\alpha = 0.7$ and dotted lines for $k = 0.6$ and $\alpha = 1$.

restored one. This measure will depend on the nature of original image. For practical numerical implementations, the process of Eq. (2) is sampled with a time step τ. The restored images $\psi(t_n)$ are calculated at discrete instant $t_n = n\tau$ with n the number of iterations.

3.1 Synthetic Image

The first proposed image is the binary image of Fig. 3.(a) corrupted by a white gaussian noise of mean zero and standard deviation σ.



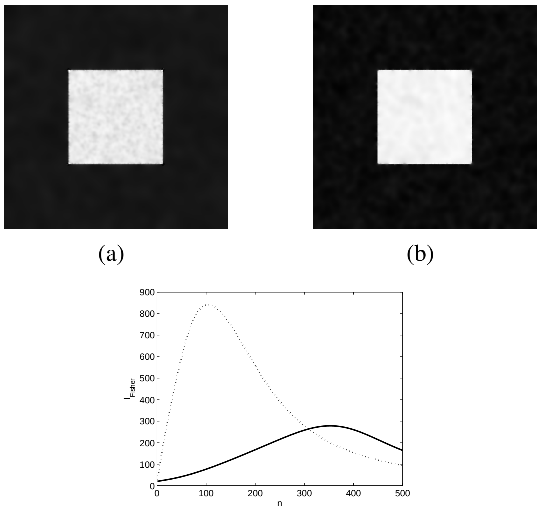
(a) (b)

Figure 3: (a) Original synthetic image and (b) its corrupted version ψ_0 . Corrupting noise is a white Gaussian one of mean zero and standard deviation $\sigma = 0.05$.

Considering binary nature of non corrupted image (Fig. 3.(a)), quantification of the denoising effect of Eq. (2) with $c(.) = c_{PM}(.)$ and $c(.) = c_{DW}(.)$, will be estimated with Fisher's index given by

$$I_{Fisher} = \frac{(m_w - m_b)^2}{\sigma_w^2 + \sigma_b^2} \ , \qquad (7)$$

with $m_{w,b}$ the average value of the pixels of the restored image $\psi(t_n)$ being originally in the white (w) or black (b) part of original image (Fig. 3.(a)) and $\sigma_{w,b}$ the corresponding standard deviation. Because aim of this article is to show potentiality of the described restoration method, only optimal results for both compared approaches are presented Fig. 4: Values of k and α parameters are empirically chosen and strategy for optimal choice is not describe here.



(a) (b)

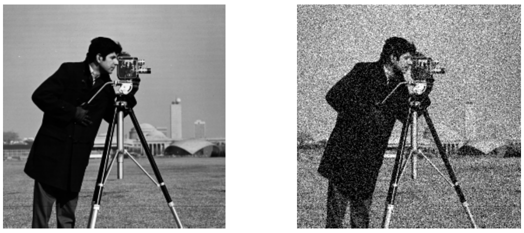
(c)

Figure 4: (a) Restored image with $c(.) = c_{PM}(.)$ (classical Perona-Malik's approach), (b) Restored image with $c(.) = c_{DW}(.)$ (proposed approach), (c) Fisher index function of iteration number n, solid lines stands for classical Perona-Malik's approach, dotted line stands for proposed method. k is equal to 0.4, α is equal to 0.5 (these values have been empirically tuned).

As one can notice on Fig. 4, both visually and quantitatively, restoration of binary image of Fig. 3.(a) is better with the diffusive function of Eq. (4). More precisely, stability property of the double well function prevents restoration process from possible enhancement of corrupting Gaussian noise. Homogenous areas of Fig. 4.(b) does not visually shows oscillations, nor corners of the white square as in Fig. 4.(a). This visual impression is confirmed by variations of Fisher's index in Fig. 4.(c) that reaches a level third times more important than with classical approach of Perona-Malik's. The value of α parameter corresponding to best result is 0.5: this is not surprising, for it is also the value of the gradient intensity characterizing the boundaries of the with square. As a consequence, this experiment also confirmed the possible gradient intensity selectivity of the proposed approach interpreted as a directional diffusion process. We shall now experiment the proposed approach in the context of restoration of real scalar images.

3.2 Real Images

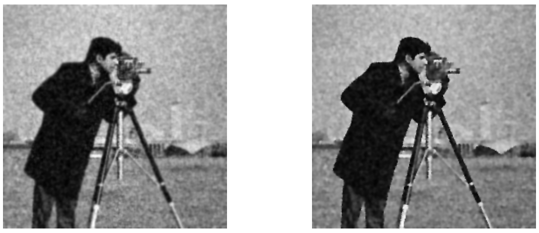
In this section, we propose to compare both our proposed method with PM's approach on the classical "cameraman" image. For our purpose, this latter has been corrupted by a white gaussian noise of mean zero and standard deviation σ (see Fig. 5).



(a) (b)

Figure 5: (a) Original image "cameraman" and (b) its corrupted version ψ_0. Corrupting noise is a white Gaussian one of mean zero and standard deviation $\sigma = 0.05$.

Considering nature of non corrupted image (Fig. 5.(a)), quantification of the denoising effect of Eq. (2) with $c(.) = c_{PM}(.)$ and $c(.) = c_{DW}(.)$, will be estimated with a classical PSNR measurement. Once again, because aim of this article is to show potentiality of the described restoration method, only optimal results for both compared approaches are presented Figs. 6 and 7.



(a) (b)

Figure 6: (a) Restored image with $c(.) = c_{PM}(.)$ (classical Perona-Malik's approach), (b) Restored image with $c(.) = c_{DW}(.)$ (proposed approach). k is equal to 1 for PM's restoration approach, α is equal to 0.2 for proposed approach (these values have been empirically tuned).

One can notice on Figs. 6 and 7 that both visually and quantitatively, it is possible to find a value of α that can outperform results of classical PM's approach. Quantitatively speaking PSNR is around 2dB higher and visually speaking, boundaries on Fig. 6 are preserved in a better way from the diffusion effect.

4 CONCLUSIONS

In this article, we have proposed an alternative diffusive function for restoration of scalar images within the framework of PDE-based restoration approaches. The proposed diffusive function allows integrating prior knowledge on the gradient level to restore thanks parameter α of Eq. (4) and remains always stable on the contrary of classical PM's approach. Proposed method also remains fast and easy to compute. Quantitatively speaking, better restoration results have been obtained. Concerning possible outlooks, this proposed method could be associated to the orientation selectivity of a PDE-based method already presented in the framework of that conference in (Histace et al., 2007). Association of both properties should lead to a restoration method with interesting directional properties for "vision in robotics" area for example.

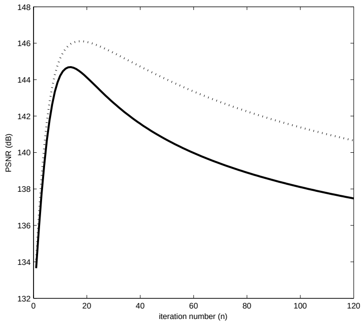


Figure 7: PSNR function of iteration number n, solid lines stands for classical Perona-Malik's approach, dotted line stands for proposed method. k is equal to 1, α is equal to 0.2 (these values have been empirically tuned). These two curves have been computed by calculation of the mean results obtained for one hundred different realizations of the gaussian corrupting noise.

REFERENCES

Catté, F., Coll, T., Lions, P., and Morel, J. (1992). Image selective smoothing and edge detection by nonlinear diffusion. *SIAM Journal of Applied Mathematics*, 29(1):182–193.

Deriche, R. and Faugeras, O. (1996). Les edp en traitements des images et visions par ordinateur. *Traitement du Signal*, 13(6):551–578.

Histace, A., Courboulay, V., and Ménard, M. (2007). Selective image diffusion for oriented pattern extraction. In *4th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, pages 270–274.

Histace, A. and Ménard, M. (2008). *Robotics, Automation and Contol Book: PDE based approach for oriented pattern segmentation*, chapter 22, pages 207–218. Intech Education and Publishing.

Perona, P. and Malik, J. (1990). Scale-space and edge detection using anistropic diffusion. *IEEE Transcations on Pattern Analysis and Machine Intelligence*, 12(7):629–639.

CONTROL AND SUPERVISION FOR AN INDUSTRIAL GRAIN DRYER

Clemente Cárdenas, Eduardo J. Moya, David García and Oscar Calvo
Fundación CARTIF, Parque Tecnológico de Boecillo. Parcela 205. 47151 Boecillo, Valladolid, Spain
clecar@cartif.es, edumoy@cartif.es, davgar@cartif.es, osccal@cartif.es

Keywords: Supervision, Control, Industrial Process, PLC, SCADA, Profibus, PID.

Abstract: Automation and control of processes in a food industry is a very important aim. The main reasons are: guaranteeing a better quality of the final product, reducing cost time and improving the use of the raw materials. Specifically, drying and storage grain industries have plants which, in many cases, are out of phase. Besides, they are controlled by machine operators. Our work has consisted in developing a total and supervision automated system to control most of the processes. A first step has been to automate four cereal dryers in order to collect data. Subsequently, a control has been designed to get a constant value of moisture of the grain. At the same time, these data have been used to obtain a total traceability of the process.

1 INTRODUCTION

In most cereal drying industries it is very important to store the final product in optimal conditions along time in order to achieve a good preservation.
Combination of several measures is necessary:

- Grain cleaning and sorting, avoiding any undesired product or seed.
- Drying until a moisture level is reached, to guarantee the correct preservation.
- Storing temperature Control during all the time that the product remains in the facilities.

In general, once recollected, grains don't have a suitable degree of humidity and temperature to be stored in silos for a long period of time. That is why it is necessary to increase the temperature in order to reduce humidity, making the drying a process of great relevance. Therefore, supervision and automation offer the operator the necessary tools to control the drying process accurately, using historical and real-time process performance information.

Improving control enhances consistency and saves energy by ensuring key process variables are more stable. Processes may also be operated closer to optimum values or constraints.

Process automation is not innovative, but if supervision and control solutions are customized, as in this case, we can deduce, then, that we are innovating.

In the following sections we describe an example of control of such processes.

2 DRYING AND STORAGE PROCESSES

Basically, the cereal drying process consists in passing a hot air current through the product, in order to reduce the moisture inside the grain.
There are several factors to take into account from the point of view of the process and also from that of the product:

- The product can have different humidity percentages.
- Moisture reduction depends on each type of grain.
- Each product has a temperature upper limit and a humidity lower limit to consider .
- The goal is to achieve a maximum performance in Tons/hour, as well as a minimum energetic consumption.

A horizontal grain dryer consists of a perforated metal sheet connected to a source of heated forced air supplied by a diesel or gas burner. The grain conduit has upper and lower ends to receive and discharge, respectively, a quantity of grain to be dried by heat conveyed to the grain through the perforated sheet. Rollers with an agitator keep grain moving downward into the dryer. It is also necessary that several extractors ease the ejection of humid air out of dryer.

The inside of a horizontal dryer and a detail of the rollers are showed in figure 1.

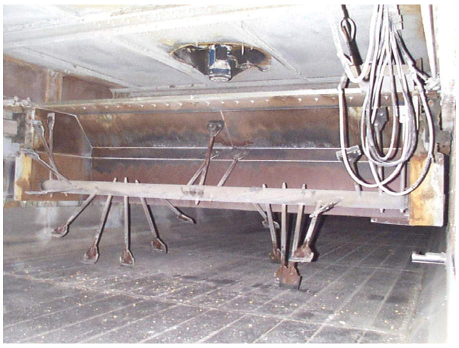


Figure 1: A typical horizontal dryer.

Before the final storage into silos, the grain is cooled.

Periodically, once the product is stored, it is advisable to control the grain temperature to avoid a lost of quality or possible explosions due to high temperatures. Implantation of a temperature and humidity control and supervision system in silos, guarantee a quality for the final customer. When the temperature rises above a reference signal, fans are switched on to introduce cool air. At the same time, the warm air is put out of the silos by means of extractors. More details of these processes are showed in (de Dios, 1996).

3 SUPERVISION, AUTOMATION AND CONTROL PROCESSES

It is necessary to integrate inside processes with many interacting elements, automation and control systems. Next, each of these systems will be explained.

3.1 Automation of the Process

Automation operation is a first step to control and supervise any process. It is used to carry out sequential motors start and stop, processes stop due to failures, motor speed control or temporisation actions. Automation architecture consists of:

- A first PLC´s to control four dryers and a second PLC to control silos fans.
- Two PC to install Silos and Dryers SCADA systems.
- Two Grain moisture measuring instruments.
- A PC-PLC PROFIBUS Communication Card.

The PLC is connected with decentralized periphery devices using PROFIBUS DP. This has been possible by letting one unique PLC control four dryers. The moisture equipment is connected to the Dryer SCADA PC through a serial RS-232C port. In figure 2 an automation scheme is depicted.

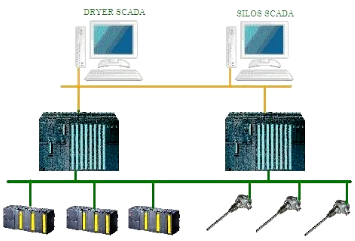


Figure 2: Automation scheme for silos and dryers control.

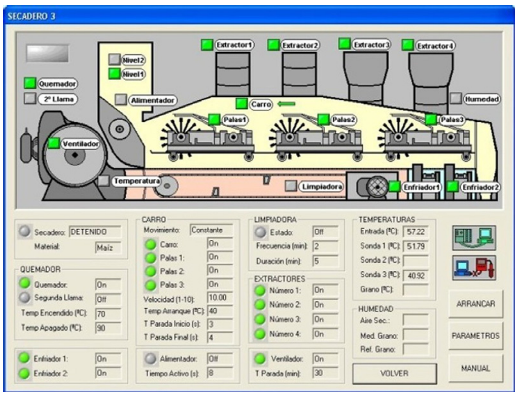
Each silo contains twenty four temperature sensors located in six levels and one outside the silo. There are also two humidity sensors (one inside and one outside). These sensors are connected to a ADC through four channels, which are connected to the Silos SCADA PC . (see figure 4 for a detailed view).

3.2 Supervision

One of the most important tasks has been the design of the supervision system. Instead of acquiring a commercial SCADA, a supervision programs have been developed with the Visual C++ tools.

According to the type of product, the supervision system makes it possible to change some parameters in order to control the drying conditions. It is possible to activate the number of extractors, to time the grain feeder, to change the discharge time, even to switch on a second flame in the burner.

In the stored grain process a complex supervision and control system has been developed. Not only can we supervise the temperature of each silo to six levels, but we can control the temperature based in different choices. For example, depending on the external air humidity and temperature, a time period or a temperatures difference can activate extractors and fans.



Secadero: DETENIDO
Material: Maiz

QUEMADOR
Quemador: On
Segunda Llama: Off
Temp Encendido (ºC): 70
Temp Apagado (ºC): 90

Enfriador 1: On
Enfriador 2: On

CARRO
Movimiento: Constante
Carro: On
Palas 1: On
Palas 2: On
Palas 3: On
Velocidad (1-10): 10.00
Temp Arranque (ºC): 40
T Parada Inicio (s): 3
T Parada Final (s): 4

Alimentador: Off
Tiempo Activo (s): 8

LIMPIADORA
Estado: Off
Frecuencia (min): 2
Duración (min): 5

EXTRACTORES
Número 1: On
Número 2: On
Número 3: On
Número 4: On

Ventilador: On
T Parada (min): 30

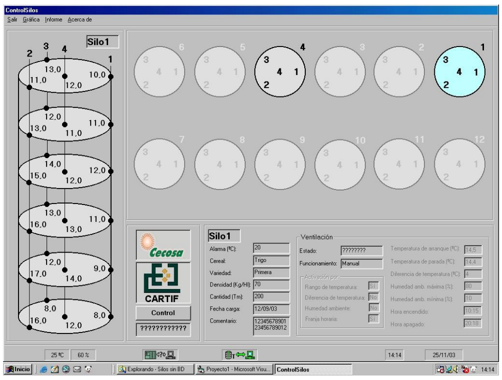
TEMPERATURAS
Entrada (ºC): 57.22
Sonda 1 (ºC): 51.79
Sonda 2 (ºC):
Sonda 3 (ºC): 40.92
Grano (ºC):

HUMEDAD
Aire Sec.:
Med. Grano:
Ref. Grano:

Figure 3: A window of dryers SCADA application.

In every reading, the control compares the actual values with the ones defined by user. If for instance the actual temperature is higher that the limit set , the system turns on the aerations system. Once the temperature below that set point the PLC shuts off the aeration system. Examples of such controls are showed in (Silva, 2003) and (Srzednicki, 2005).
A larger and more complex system totally controllable and observable through the internet, by user our SCADA Web (Janeiro, 2006) is being developed.

The parameters of both supervision systems can be changed from a window as depicted in Figure 3 and figure 4 .



Silo1
Alarma (ºC): 20
Cereal: Trigo
Variedad: Primera
Densidad (Kg/Hl): 70
Cantidad (Tm): 200
Fecha carga: 12/09/03
Comentario: 12345678901 23456789012

Ventilación
Estado: ????????
Funcionamiento: Manual

Activación por:
Rango de temperatura: Sí
Diferencia de temperatura: No
Humedad ambiente: No
Franja horaria: Sí

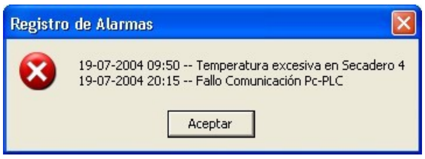
Temperatura de arranque (ºC): 14,5
Temperatura de parada (ºC): 14,4
Diferencia de temperatura (ºC): 4
Humedad amb. máxima (%): 80
Humedad amb. mínima (%): 10
Hora encendido: 10:15
Hora apagado: 20:18

14:14
25/11/03

Figure 4: Temperature Silos SCADA application.

The prime emphasis in these processes has been on improving the security of existing machines. Due to different controls implemented it is possible to generate a important number of alarms. The most important one is the temperature control, used to avoid possible fires inside of the dryers.

The communication between all equipments; computers, humidity measuring instruments and PLCs, is continuously being checked.
All these alarms and other parameters are registered in a database, to be analysed in order to verify the stored and drying conditions, and improve the processes.


Registro de Alarmas

19-07-2004 09:50 -- Temperatura excesiva en Secadero 4
19-07-2004 20:15 -- Fallo Comunicación Pc-PLC

Figure 5: Alarms registration.

3.3 Moisture Control System

Most development work has centered on the optimization of machine design and capacity, and the application of these machines to existing processing strategies. The result has been the development of more compact dryers in recent years. The manner in which processing itself is carried out, must be considered. There are works related to to the engineering aspects of the process, but researchs carried out on dryers control are not quite extensive. Recent studies are based on the use control techniques principally PID (Guofang, 2006),predictived (Qiang, 2001) and fuzzy control, (Zhang and Litchfiled, 1993),(Bremmer, 1997) and (Chunyu et al., 2007).

Control techniques principally consist of a computer program and a number of sensors measuring process properties. It is also necessary to have some forms of SCADA/PLC systems.

The goal is to design a feedback controller for the plant shown by the block diagram in figure 6, which includes the feedback interconnection of the plant and controller, and elements associated with the performance objectives.

The moisture error of the discharged grains was used as input parameter of the controller. The output parameter of the controller was the speed of the rolls.

Continuous-flow grain drying is a non-linear process with a long delay; it is often subjected to large disturbances and therefore is difficult to control. The ON/OFF and PID designed controllers have been an adequate control method in this type of machines.

The grain humidity has to be controlled by changing the temperature inside the dryers.

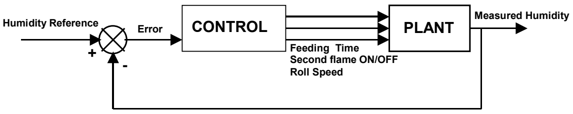


Figure 6: A dryer control diagram.

It has been verified that it is more effective to switch the two flames of the burner than modifying the speed of the rolls. The combination of an ON/OFF control of burner with a PID control actuating in roll speed makes it possible to achieve an optimal control of humidity grain in these dryers.

In spite of this good behaviour of this type of controls, we are developing robust control due to long delays and disturbances in some cases. A work of an implemented robust control in a similar process can be viewed in (Cárdenas, 2003).

Simulations test show that the robust controller performed well over a wide range of drying conditions.

4 CONCLUSIONS

Process automation and supervision seem to promise significant potential for development in the future.

The efficency of dryers has been increased significantly. This has been achieved by making them larger, more space efficient and by increasing control and supervision systems. The incorporation of these controls has also made itpossible to reduce the grain humidity before it is stored into silos. In addition, data collection and analysis, as well as product traceability, ensures optimum quality for customers and tools to enhance profitability.

The control method provide a new solution for grain drying process.

Advanced controllers are being simulated with good results and we expect to implement them in the factory in the future for a better optimal energy consumption.

ACKNOWLEDGEMENTS

This work was supported in part by “Programa Nacional de Recursos y Tecnologías Agroalimentarias”, (PROFIT) from the Spanish Technology and Science Ministry.

REFERENCES

Bremner, H., 1997. An application of model-based fuzzy control to an industrial grain dryer. *Transactions of the Institute of Measurement and Control*, volume 19, nº4, pages 185-191.

Balas, G., J. C. Doyle, K. Glover, A. Packard and R. Smith., 1994. *μ-Anaysis and Synthesis Toolbox for use with MATLAB*. The Mathworks.

Cárdenas, C., Garcia, J.D., Baeyens E., Olmos J.D., 2003. Robust controllers for granulation processes in the food industry. *AFOT 03 Congress*.

Chunyu, Z., Qinglei, C. et al., 2007. A model predictive of a grain dryer with four stages based on recurrent fuzzy neural network. *In Springer Berlin, editor, Lectures Notes in Computer Science*, volume 4491, pages 29-37.

De Dios, C. A. (1996). Secado de granos y secaderos. *Organización de las Naciones Unidas para la agricultura y la Alimentación*. Santiago, Chile

Guofang, L., Zhihuai, M., 2006. An intelligent controller for grain dryer. *American Society of Agricultural and Biological Engineers*. Paper nº 066197.

Janeiro, J., Moya, E., Garcia, D., Calvo, O., Cardenas, C., 2006. Remote Supervisión and Maintenance of Industrial Proceses. *ICINCO´07 Congress*.

Qiang, L., Bakker-Arkema, F.-W., 2001. Automatic control of crossflow grain dryers, Part2: Design a model-predictive controller. *Journal of agricultural engineering reseach.*volume 80. nº2, pp 173-181.

Silva, J., Alves, G.R, Alves, B.M., 2003. A scale model of a grain storage tower equipped with micro web servers. *International Journal of online engineering*

Srzednicki, G., Hou, R., Driscoll, H. R., 2005. Development of a control system for in-store drying of paddy in Northeast China. *Journal of Food Engineering*, volume 77, Issue 2, pages 386-377.

Zhang, Q., Litchfiled. 1993. Fuzzy logic control for a continuos crossflow grain dryer. *Journal of Food Process Engineering.* Volume 16, pages 59-57.

Cadder., 1992. Computer control system for continuous and semi-continuous grain dryers. Result 106. *Building and industry publications*.

PROSPECTIVE ELASTO-PLASTIC PRESSURE SENSORS

All-Elasto-Plastic Polyisoprene/Nanostructured Carbon Pressure Sensing Element

Maris Knite, Juris Zavickis, Gatis Podins, Raimonds Orlovs and Kaspars Ozols
Institute of Technical Physics, Riga Technical University, Azenes str. 14/24, Riga, Latvia
knite@latnet.lv, juriszavickis@inbox.lv, gatis.podins@pa.lv, raimonds@ktf.rtu.lv, kozols@ktf.tu.lv

Keywords: Completely Flexible Pressure Sensor, Polyisoprene, High-Structured Carbon Black, Hybrid Composite.

Abstract: Our further achievements in the design, processing and studies of physical properties of elastomer – nano-structured carbon composites as prospective compressive strain sensor materials for robotic tactile elements as well as for other automatic systems are presented. Composites made of polyisoprene matrix and high-structure carbon nanoparticle filler have been designed and manufactured to develop polymer nano-composites for flexible, entirely polymeric pressure sensing elements. Electrical resistance of the composites as a function of mechanical strain and pressure is studied. SEM pictures of cross-section surface of sensing elements are analyzed.

1 INTRODUCTION

In our previous study we have already shown the possibility to manufacture entirely flexible PNC sensing element with glued conductive rubber electrodes (Knite, 2008). Such elements show good mechano-electrical properties but they have one drawback – possible delamination of flexible electrodes during operation. In this study we further developed the technology of preparing all-elasto-plastic (AEP) strain sensing element with vulcanized flexible electrodes made of polyisoprene-nanostructured carbon (PNC) composite. Recently, some promising results have been presented regarding the application of polymer/conductive filler composites as strain and pressure sensors as well as selective gas sensors (Knite, 2002; Knite, 2004; Qu, 2007; Li, 2008; Knite, 2007; Sakale, 2009). Interesting and excellent properties have been obtained in case the composite contains dispersed nano-size conducting particles. If the size of carbon particle and specific surface area of carbon black are between 60 to 200 nm and 16-24 m^2/g, respectively (low-structure carbon nano-particles (LSNP)), the electrical resistance of natural rubber composites slowly decreases with applied pressure (Job, 2003). The effect is explained by the increasing number of conductive channels due to the increase of external pressure. Resistance of natural polyisoprene-carbon nanocomposites grows very rapidly and reversibly for both – tensile and compressive strain when high-structure carbon nano-particles (HSNP) (specific surface area 950 m^2/g, mean diameter 25 nm) are used as the filler (Knite, 2007). The sensing elements described in all mentioned papers contain metallic electrodes that reduce the flexibility of the whole element as well as delamination of electrodes can be possible due to bending. In this paper our recent success in the design, processing and studies of properties of vulcanized foliated composite sensor element is reported.

2 PREPARATION OF SAMPLES AND THE EXPERIMENT

The polyisoprene – nano-structured carbon black composite was made (see Figure 2) by rolling high-structure PRINTEX XE2 (DEGUSSA AG) nano-size carbon black (CB) and necessary additional ingredients (sulphur and zinc oxide) into a Thick Pale Crepe No9 Extra polyisoprene (MARDEC, Inc.) matrix and vulcanizing under 3 MPa pressure at 155 ˚C for 20 min. The mean particle size of PRINTEX XE2 is 30 nm, DBP absorption – 380 ml/100 g, and the BET surface area – 950 m^2/g.

The sensor element was made as follows. Two blends of polyisoprene accordingly with 30 and 10 phr (parts per hundred rubber) carbon black have been mixed. Initially 30 phr of PRINTEX have been used for obtaining PNC composite electrodes, but the tests of mechanical and electrical properties showed, that electrodes made from PNC composites with 20 phr of PRITEX were as much conductive as 30 phr carbon black/polyisoprene electrodes but had better elasticity as well as superior adhesion to active element. Three semi-finished rounded sheets made from mentioned above two PNC composite blends have been formed and fitted onto special steel die. Those are two sheets for conductive electrodes (30 phr CB) and one sensitive sheet (10 phr CB) for pressure-sensing part. Each of these three sheets were separately pre-vulcanized under 3 MPa pressure and 110°C temperature to obtain flat surfaces. This operation lasts 10 minutes. After that the components were cooled and cleaned with ethanol. Further, all three parts were joined together in one sensor element and were placed into the steel die and vulcanized under pressure of 30 MPa and 155° C temperature for 20 minutes vulcanization (previous attempts (Knite, 2008) to create sensor element with conductive glue were shown to be relatively ineffective). To study mechano-electrical properties small brass foil electrodes were added before vulcanization. Finally, disc shape sensor 50 mm in diameter and 3 mm thick was obtained. From this preparation we cut out useful sensor elements for testing (Figure 1).

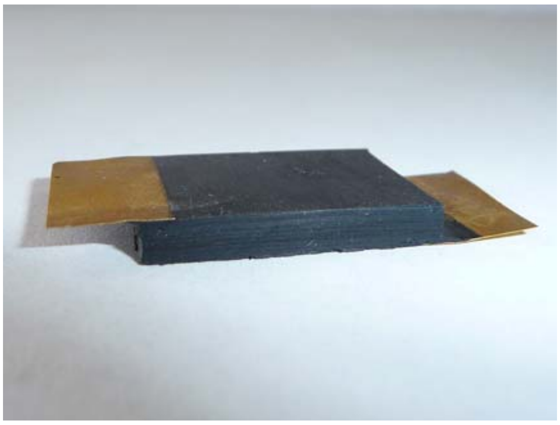


Figure 1: The accomplished all-elasto-plastic sensor element with brass foil extensions.

A modified Zwick/Roell Z2.5 universal testing machine, HQ stabilized power supply and a KEITHLEY Model 6487 Picoammeter/Voltage Source was used for testing mechano-electrical properties of sensor elements. All devices were synchronized with the HBM Spider 8 data acquisition logger. Resistance R versus compressive force *F* was examined. Uniaxial pressure was calculated respectively.

3 RESULTS AND DISCUSSION

Before testing the accomplished sensor element, we measured the electrical properties of separate vulcanized electrode layers. We also separately tested the mechano-electrical properties of vulcanized active element layer to see whether it has expected sensing capabilities. The active element of the sensor (nano-structured carbon black composite with 10 phr) belongs to the region of the percolation threshold (specific electrical resistance ρ = 12 Ω·m). The specific resistance for flexible electrodes is in the order of 0.1 Ω·m, which is noticeably above the percolation threshold.

Let's look closer at the conductivity properties of sensors. Measurement results for electrical resistance versus pressure for small pressure range are given in Figure 2.

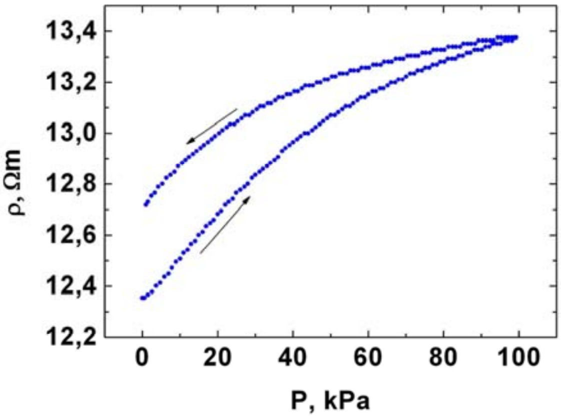


Figure 2: Electrical resistance of the all-elasto-plastic sensor element as function of pressure (lower pressure range, T = 294 ^{0}K).

Measurement results for relatively large pressure range are shown in Figure 3. The observed positive piezoresistance effect can be explained by transverse slip of nano-particles caused by external pressure leading to disarrangement of the conductive channels. The volume concentration of conductor particles V_C at which the transition proceeds is called the percolation threshold or the critical point. According to the statistical model, conductor particles, in the vicinity of V_C, assemble in clusters. Upon approaching V_C, the correlation radius ξ (the average distance between two opposite particles of a cluster) diverges as

$$\xi \sim /V\text{-}V_C/^{-\nu} \qquad (1)$$

where ν is the critical index (Roldughin, 2000).

In the vicinity of the percolation threshold, electrical conductivity of the composite changes as:

$$\sigma \sim /V\text{-}V_C/^t \qquad (2)$$

where t is the critical index (Roldughin, 2000). Under mechanical deformation of composites ξ and, consequently, σ change. This is the reason causing the piezoresistive effect.

Because of higher mobility of HSNP compared to LSNP the electro-conductive network in the elastomer matrix is easily disarranged by very small tensile, compressive or shear strain. We suppose this feature makes the elastomer–HSNP composite an option for flexible sensitive tactile elements for robots and automatics.

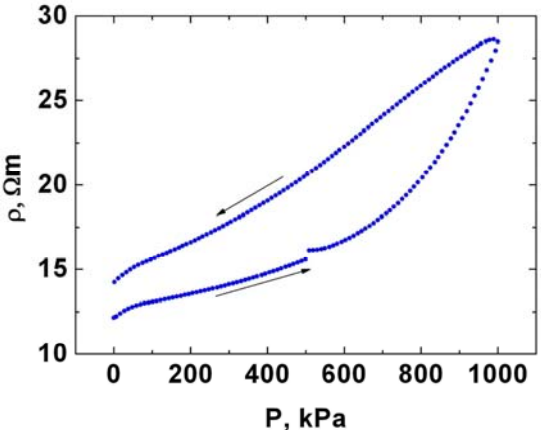


Figure 3: Electrical resistance of the all-elasto-plastic sensor element as function of pressure (higher pressure range, T = 294 ^{0}K).

The scanning electron microscopy (SEM) was used to check the quality of joined regions of three PNC sheets of the AEP sensor element. SEM micrographs of breaking surface of the sensor element are shown in Figure 4. To prepare the sample for SEM investigations the sensor element was frozen in liquid nitrogen and then broken. Good quality of joining of all three PNC sheets can be clearly visible in SEM images with different scale (Figure 4). Pale regions correspond to electrically more conductive PNC composite with 30 phr CB and dark regions cover the PNC composite with 10 phr CB. The pale particles, which are visible in the bottom picture (Figure 4), are carbon nano-particles.

A functioning model of low-pressure-sensitive indicator was made. The block diagram of pressure indication circuit is shown on Figure 5. The sensor is connected to power supply (PS) via resistor (R) and to the input of amplifier (Amp). Transistor-based two-stage amplifier includes integrating elements.

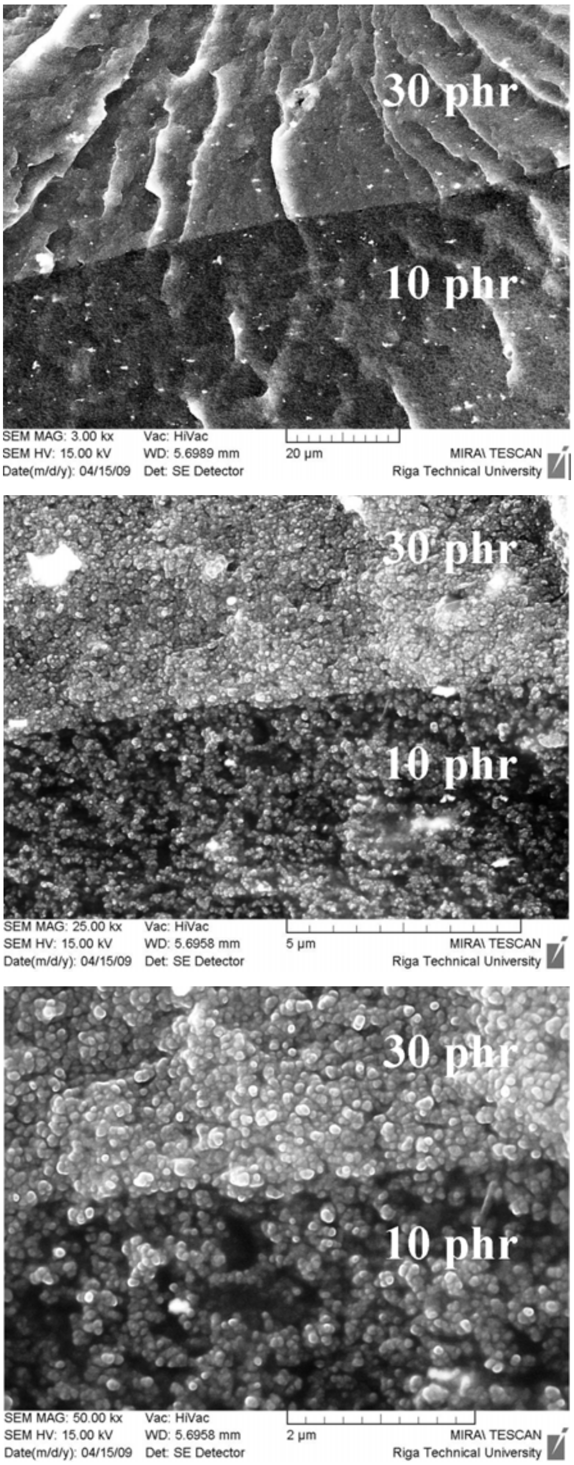


Figure 4: SEM micrographs of sensor element. Sensor element was frozen in liquid nitrogen and then broken in two. One of the broken sides is shown in different scales: 20 µm, 5 µm and 2 µm. Boundary between two PNC composite layers with 10 and 30 phr (parts per hundred rubber) carbon black are shown.

These elements are necessary to avoid noise from induced currents and to flatten the wavefronts. The first stage amplifies the signal in linear mode. The second stage works in saturation mode. The output of the amplifier is connected to the comparator (Comp), which forms sharp wavefronts.

These signals are passed to the differential circuit and they form a sharp pulse, which is passed further to the one-shot multivibrator (OSM).

The duration of the pulse of the OSM is adjustable. The OSM is necessary to form the determined length of pulse which is independent from AEP sensor element deformation time. The output of OSM is connected to performing device PD (indicator/counter or actuator).

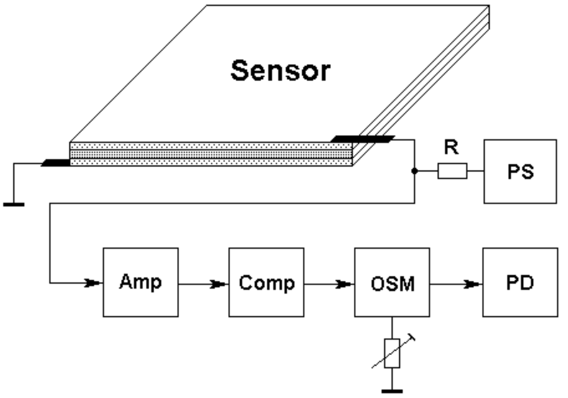


Figure 5: Block diagram of pressure-sensitive indication circuit with completely elasto-plastic sensing element.

4 CONCLUSIONS

Completely flexible polyisoprene – high-structured carbon black all-elasto-plastic sensing element has been designed, prepared and examined.

The sensor element was composed of two electrically conductive composite layers (electrodes) and piezoresistive PNC layer (active element) between them. A method for curing three-layer hybrid composite for pressure sensing application was developed. The joining in-between conductive flexible electrodes and sensitive sensor material was remarkably improved.

Hybrid three-layer polyisoprene/high-structure carbon black composite has shown good pressure sensing properties. Functioning model of low-pressure-sensitive indication circuit which can turn on suitable actuator has been made.

ACKNOWLEDGEMENTS

The research has been supported by Latvian National Research program in Materials Science. The authors are thankful to Mr. Dmitrij Jakovlew from the Institute of Biomaterials and Biomechanics of the Riga Technical University for the TEM investigations.

REFERENCES

Knite, M., Podins, G., Zike, S., Zavickis, J., Tupureina, V., 2008. Elastomer – carbon nanostructure composites as prospective materials for flexible robotic tactile sensors. In *Proc. of 5th International Conference on Informatics in Control, Automation and Robotic,* 1: 234-238.

Knite, M., Teteris, V., Polyakov, B., Erts, D., 2002. Electric and elastic properties of conductive polymeric nanocomposites on macro- and nanoscales. Materials Science & Engineering C, 19: 5-19.

Knite, M., Teteris, V., Kiploka, A., Klemenoks, I., 2004. Reversible tenso-resistance and piezo-rezistance effects in conductive polymer-carbon nanocomposites. *Advanced Engineering Materials,* 6: 742-746.

Qu, S., Wong, S., C., 2007. Piezoresistive behaviour of polymer reinforced by expanded graphite. *Composites Science and Technology,* 67, 231-237.

Li, X., Levy, C., Elaadil, L., 2008. Multiwalled carbon nanotube film for strain sensing. *Nanotechnology*, 19: 045501 (7pp).

Sakale, G., Knite, M., Teteris, V., Tupureina, V., 2009. Polyisoprene – nanostructured carbon composite (PNCC) material for volatile organic compound detection, *Proc. of the International Scientific Conference on Biomedical electronics and Devices (BIODEVICES 2009),* Porto, Portugal, 117.

Job, A.E., Oliveira, F.A., Alves, N., Giacometti, J.A., Mattoso, L.H.C., 2003. Conductive composites of natural ruber and carbon black for pressure sensors. Syntetic metals, 135-136: 99-100.

Knite, M., Klemenok, I., Shakale, G., Teteris, V., Zicans, J., 2007. Polyisoprene-carbon nano-composites for application in multifunctional sensors, *Journal of Alloys and Compounds,* 434-435: 850-853.

Roldughin, V., I., Vysotskii, V., V., 2000. Percolation properties of metal-filled films, structure and mechanisms of conductivity, *Progres in Organic Coatings*, 39: 81-100.

COOPERATIVE UAVS MAPPING COMPLEX ENVIRONMENT USING 2D SPLINEGON

Samuel B. Lazarus, Antonios Tsourdos, Brian A. White, Rafał Żbikowski and Peter Silson
Department of Informatics & Sensors, Cranfield University
Defence Academy of the United Kingdom, Shrivenham, U.K.
a.tsourdos@cranfield.ac.uk

Keywords: Multiple sensor fusion, Circle Packing, Path planning, Integrated GPS/INS navigation, Obstacle and collision avoidance, Splinegon, Boundary estimation and Data association.

Abstract: This paper presents a novel approach which enables multiple UAVs to efficiently explore an unknown environment and incrementally build the map of the area and its complex shaped obstacles, represented here as concave and convex in shape. The task is achieved by a improved performance of sensor based searching, navigation and mapping of these complex shaped obstacles in an unknown environment. The improved performance is quantified by explicit bounds of navigating the UAVs using an extended Kalman filter and to build the map of the complex shaped obstacles using the 2-D Splinegon. The circle packing search algorithm is used for the completeness of coverage in searching the unknown obstacles regions and the UAVs trajectories are generated by the Dubins path planning algorithm. This novel proposed algorithm results in a robust approach to search and map the obstacles using multiple UAVs that is also computational attractive.

1 INTRODUCTION

This paper focusses on swarm of UAVs deployed for a mission of searching an unknown region to detect obstacles and to extract their shape. The circle packing search algorithm (Kershner, 1939), (Guo and Qu, 2005) is implemented where the search is carried out by a sequence of “looks” each of which covers a circle corresponding to the footprint of the sensor on board. This circle pacing algorithm covers the plane by packing each circles into the unknown environment. The centers of each of this packed circles represent the way points to be used on path planning of the UAVs. A mission planning algorithm is described which enables the UAVs to switch between the searching mode and the mapping of the detected obstacle. Since the swarm of UAVs fly around an unknown environment, a sense and avoid system is developed so the UAVs autonomously replan their paths when they approach an obstacle or predict intersection of air traffic. Thus the system presented here provides a safe surveillance of unknown areas by swarm of UAVs. Furthermore the proposed mission planning not only enables the swarm of UAVs to switch from searching mode into mapping mode, but also ensures the allocation of the required number of UAVs to map that obstacle with in searched region.

In the fulfillment part of the mapping task, the measurements from laser sensor that are mounted on the UAVs is the only source to construct the map of the detected obstacle. This strongly suggests that the most efficient way in modelling approach should be to define these measurement points as vertices that can form a polygon with line segments. This raises an issue as to how to represent the curved nature of these obstacles. One such promising approach is introduced in this paper that uses a generalization of polygons that produces a set of vertices that are connected by line segments of constant curvature. This is a subset of a class of objects named as splinegons (Dobkin et al., 1988), (Dobkin and Souvaine, 1990).

In the mapping process the fused EKF estimated positions are used with the limited number of measurements (i.e. required number of the the interpolation points) from the laser sensors to build the map. As the vehicles fly around the obstacle, sensors such as laser sensors are used to measure the distance to the obstacles. Out of all these measurements, only a carefully selected number of measurements are chosen which represent the required vertices to construct a simple polygon. The data association algorithm is implemented to select a limited number of vertices and to uniformly distribute these vertices around the obstacle in a computationally efficient way. This selection of maximum number of vertices that can form a polygon is limited from sixteen to twenty vertices. This selection of vertices is accomplished by calculating the length and the curvatures between each of the vertices and to eliminate the vertex that has a minimum length and curvature. The selection process is based on the size and the curved nature of the obstacle. Thus, the constructed polygon is generalised to produce a set of vertices that are connected by line segments of constant curvature called splinegon. This splinegon is a set of vertices that have constant curvature line segments defined with C^2 contact at the vertex points.

1. The prediction of next way points for the search algorithm and prediction of a new way points when obstacle/collision avoidance algorithm is activated.

2. Switching between the search mode into the mapping the unknown obstacle and predict the way points and its orientation to get the measurements.

3. The prediction of the required number of vehicle to accomplish the mapping task (How many vehicle is needed to complete the mission?)

4. If more then one UAV is used for the mapping task,

   (a) Find a shortest way points for the other UAVs to reach the mapping region.

   (b) Whether the UAVs needs to exchange the information with one to another, and

   (c) Identify the state of the obstacle whether it is single or multiple, depending upon the intersection detection between the local updates. Eventually, the global update is performed between the intersection detected UAVs.

5. The prediction of required number of measurement for the mapping from each of the detected obstacles.

This newly proposed algorithm gives a solution that has a highly richness in building the curved nature of the unknown obstacle in an unknown environment. This proposed approach is a computationally attractive one resulting in information driven mission planning and the mapping of unknown environments with limited measurements.

The development of this algorithm is shown in the functional block diagram in figure 1.

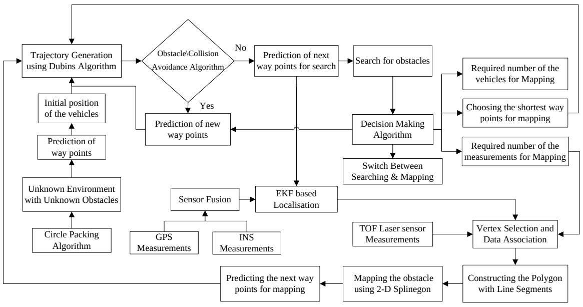

Trajectory Generation using Dubins Algorithm
Obstacle\Collision Avoidance Algorithm
No
Yes
Prediction of next way points for search
Search for obstacles
Required number of the vehicles for Mapping
Choosing the shortest way points for mapping
Required number of the measurements for Mapping
Initial position of the vehicles
Prediction of new way points
Decision Making Algorithm
Prediction of way points
Switch Between Searching & Mapping
Unknown Environment with Unknown Obstacles
Sensor Fusion
EKF based Localisation
Circle Packing Algorithm
GPS Measurements
INS Measurements
TOF Laser sensor Measurements
Vertex Selection and Data Association
Predicting the next way points for mapping
Mapping the obstacle using 2-D Splinegon
Constructing the Polygon with Line Segments

Figure 1: The functional block diagram.

2 PATH PLANNING AND CIRCLE PACKING ALGORITHM

The path planning algorithm generates a flyable and safe path to the UAVs to fly from one location to another. The locations are predefined by the way-points. The starting location is called base and the final location to be reached is called target. The Dubins set of paths is used to connect the base and target by a sequence of successive way-points. The base and the target are characterized by the poses, in a set of position coordinates (x,y) and orientation θ of the UAVs. Considered that an initial pose $P_s(x_s,y_s,\theta_s)$ at the base and a final pose at the target $P_f(x_f,y_f,\theta_f)$ are given, the path $r(t)$ connecting the two poses can be a single or a composite curve. A path represented by a curve in $2D$ is completely determined by its curvature (Kreyszig, 1991) and the maximum curvature bound of the UAV by κ_{max} and other constraints by $\prod$; then the path planning can be mathematically represented as:

$$P_s(x_s,y_s,\theta_s) \xrightarrow{r(t)} P_f(x_f,y_f,\theta_f),\ \kappa(t) < \kappa_{max},\ \prod \quad (1)$$

Extending the equation (1) for a group of UAVs, this changes into:

$$\begin{aligned} P_{si}(x_{si},y_{si},\theta_{si}) &\xrightarrow{r_i(t)} P_{fi}(x_{fi},y_{fi},\theta_{fi}) \\ \kappa_i(t) &< \kappa_{i,max} \quad \text{and} \quad \prod \end{aligned} \quad (2)$$

where $\kappa_i(t)$ is the curvature, $\kappa_{i,max}$ is the maximum curvature bound of i^{th} path, and $i = 1 \ldots N$, N is the number of UAVs. This path planning algorithm is integrated with an obstacle and Collision avoidance system to generated the trajectories for each UAVs and to protect the flying UAVs from any collision with the surrounded obstacles or with the on coming UAVs. This algorithm is applied online while the UAVs are in motion.

The circle packing algorithm (Washburn, 1981) is to attempt to cover the given plane by packing the circles of radius R. In other words finding a minimum number of circles with the radius of R to completely cover in the given area of search. This is accomplished by fixing the coverage range or the sensor range are represented as a circle. The key problem is to determine the required number of circles with a radius R to cover the given area. This in turn produced many solutions to this problem. But the objective is to find an optimal solution to minimise the repeated search. One such algorithm that was reported in (Kershner, 1939) and (Guo and Qu, 2005) is implemented here so as to covert the whole area to perform the searching task.

The solution to optimally place the minimum number of circles can be described as the circle has the radius of R_c and the ares to covered with this circle is denoted as W. A pattern is composed of a string of circles with radius R_c that has to placed along the vertical line, and the distance between the centers of any of the two adjacent circles is $\sqrt{3}R_c$. The m column of circles are place that are oriented parallel to the $Y-axis$ and the in the same way the distance between the centers of any of the two adjacent circles is $1.5R_c$. The origin $[x_o, y_o]$ is chosen at the left bottom of the given area W. This in turn enable to place the m circles that are parallel to the $y-axis$ which contains the n number of circles to completely cover the given area. So the center $[x_c^{kl}, y_c^{kl}]$ of the k^{th} row $(1 \leq k \leq n)$ and the l^{th} column $(1 \leq l \leq m)$ can be defined viz:

$$[x_c^{kl}, y_c^{kl}] = \begin{cases} [x_o + (l-1)3/2R_c,\ y_o + (k-1)\sqrt{3}R_c] \\ \text{if } l \text{ is an odd integer} \\ [x_o + (l-1)3/2R_c,\ y_o + \sqrt{3}/2R_c \\ +(k-1)\sqrt{3}R_c] \\ \text{if } l \text{ is an even integer} \end{cases} \quad (3)$$

So, the required number of circles needed in each of the column and row m and n can be defined as follows:

$$m = \begin{cases} \text{Int } \left(\frac{x_w}{1.5R_c}\right) + 1, & \text{if Rem } \left(\frac{x_w}{1.5R_c}\right) \leq \frac{2}{3} \\ \text{Int } \left(\frac{x_w}{1.5R_c}\right) + 2, & \text{if Rem } \left(\frac{x_w}{1.5R_c}\right) > \frac{2}{3} \end{cases} \quad (4)$$

$$n = \begin{cases} \text{Int } \left(\frac{y_w}{\sqrt{3}R_c}\right) + 1, & \text{if Rem } \left(\frac{x_w}{1.5R_c}\right) \leq \frac{1}{2} \\ \text{Int } \left(\frac{y_w}{\sqrt{3}R_c}\right) + 2, & \text{if Rem } \left(\frac{x_w}{1.5R_c}\right) > \frac{1}{2} \end{cases} \quad (5)$$

Where, x_w and y_w is the length of the environment along $X-axis$ and $Y-axis$ respectively. I is an integer number and Rem is the reminder of the number, in which $Rem = x - Int(x)$. So by applying the above equations the required number of circles for each of the row and column is obtained (Kershner, 1939). The prediction of a set next way-points to start the mapping task is performed that would generate a set of way points for each UAVs from its current location to reach the starting point of the mapping task.

3 DEFINITION OF 2D SPLINEGON

A Splinegon with constant curvature line segments can be defined with C^2 contact at the vertices. This implies that the line segments share both a common vertex and that the tangents at the vertices are also the same. In order to ensure C^2 contact between vertices, the line segments must meet both position and tangent end point constraints. A single arc segment between vertices only has one degree of freedom: the arc curvature. This is not enough to be able to match the tangent constraint at both end vertices, as at least two degrees of freedom are necessary. Extra degrees of freedom are thus required to ensure the C^2 constraints were both line segments end vertices can be met. One solution to increasing the degrees of freedom is to introduce an intermediate vertex such that the line segment is replaced by two arc segments of different curvature, as shown in figure 2. Hence two arcs of differing curvatures will connect the UAV vertices via the intermediate vertex. In order to develop the defining equations for such a solution, the intersection of two constant curvature arcs at a point with C^2 contact is considered.

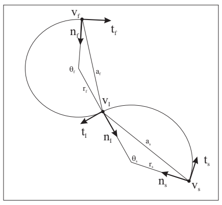


Figure 2: Arc segment with C^2 contact intermediate vertex.

4 IMPLEMENTATION AND RESULTS

The primary objective of the current work is to be able to search the given unknown environment with a swarm of UAVs to detect the region of the unknown obstacles and to extract the shape of the obstacle using 2-D Splinegon technique. Initially the circle packing algorithm is implemented that would pack the required number of circles to fit in the given environment. The search algorithm is employed with the swarm of UAVs that would predict the next way points in online using the neighboring way points or it will follow the given set of POI to detect the obstacles where the path of the vehicle is dictated by the Dubins path planning algorithm. By fixing the detecting sensor range, if any of the UAV is not able to reach any given next way point, or if the obstacle avoidance algorithm is activated so as to prevent the UAV not reaching the given way point then that circle is added into the obstacle region. Once the search is finished, depending upon the search region each of the following requirements are taken by the decision making algorithm.

- Find the area of the obstacle region.
- Find the required number of UAVs to accomplish the mapping task. This can be done based on the area of the obstacle region.
- Generate the way point for each of the UAVs to perform the mapping task.
- Finally, find the shortest way points to each UAVs to reach the starting point of the mapping task from its current location.

As the vehicle moves each of the vehicle is localised with an EKF. At the end of the each cycle (i.e., at the completion of one way point) a local updated map is constructed using the Splinegon technique. In the case of more than one UAV is used in mapping, then an intersection detection algorithm is implemented so as to identify the state of the obstacle and to share the sensor information with the other UAVs. Finally the global updated map is constructed to get the map of the unknown environment. The circles with radius R_c which are packed in using circle packing algorithm is shown in figure 2 (a). Then the search algorithm is carried out by giving a set of point of intrust(POI) way points to each of the UAVs so as to find the obstacle region. The shaded circles where the UAVs could not reach are known as the obstacle region which is shown in figure 2 (b). At the end of the search algorithm, the vehicle are switched from searching mode to the mapping the shape of the complex obstacle. The local update is performed at the end of the each way points. Finally the set of vertices that forms a polygon with line segments in the final update and the global updated map of the given unknown environment is shown in figures 4 (a) and 4 (b).

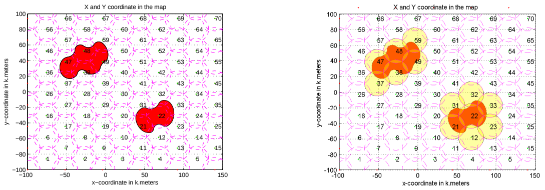


Figure 3: The results of the circle packing algorithm.

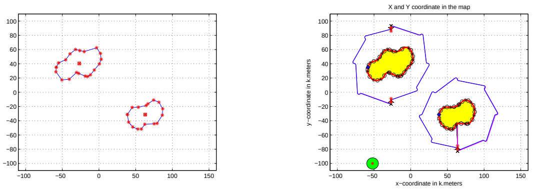


Figure 4: The final updated map using Splinegon.

5 CONCLUSIONS

In this paper the authors have described a novel, computationally attractive, approach in estimating the localisation and mapping for curvilinear objects using multiple UAVs. It enables to map obstacles of curvilinear shape, the data association for the networked sensor platforms and the reactive tasking the UAVs. Future work will extend the Splinegon technique to 3D in the robotic network that will enable the flight paths to have even greater flexibility and will enable the complex 3D shapes to be represented by a small set of parameters.

REFERENCES

Dobkin, D. P. and Souvaine, D. L. (1990). Computational geometry in a curved world. *Algorithmica*, 5(3):421–457.

Dobkin, D. P., Souvaine, D. L., and Wyk, C. J. V. (1988). Decomposition and intersection of simple splinegons. *Algorithmica*, 3:473–485.

Guo, Y. and Qu, Z. (2005). Coverage control for a mobile robot patrolling a dynamic and uncertain environment. *Proceedings of the 5th world Congress on Intelligent Control and Automation*, pages 4899–4903.

Kershner, R. (1939). The number of circles covering a set. *American Journal of Mathematics*, 61(3):665–671.

Kreyszig, E. (1991). *Differential geometry*. Dover Publications, Inc., New York.

Washburn, A. R. (1981). *Search and Detection*. Millitry Applications Section Operations Research Socitey of America.

A COMPARATIVE STUDY BETWEEN CONVENTIONAL AND CONTINUOUS GENETIC ALGORITHMS FOR THE SOLUTION OF CARTESIAN PATH GENERATION PROBLEMS OF ROBOT MANIPULATORS

Za'er Salim Abo-Hammour, Mohammad Suleiman Saraireh

Jordan University, Faculty of Engineering, Mechatronics Department, Amman-Jordan
Mutah University, Faculty of Engineering, Computer Engineering Department, Mutah-Kerak-Jordan
Zaer_hr@yahoo.com, m_srayreh@mutah.edu.jo

Othman M-K. Alsmadi

Jordan University, Faculty of Engineering, Electrical Department, Amman-Jordan
Othman_mk@yahoo.com

Keywords: Robot manipulators, Singularity avoidance, Cartesian path generation, Inverse kinematics problem, Continuous genetic algorithms, Conventional genetic algorithms.

Abstract: In this paper, a comparative study between the continuous and the conventional GAs for the solution of Cartesian path generation problems of robot manipulators is performed. The difference between both algorithms lies in the ways in which initialization phase, the crossover operator, and the mutation operator are applied. Generally, the operators of the Continuous Genetic Algorithms (CGA) are of global nature, i.e., applied at the joint's path level, while those of conventional GA are of local nature, i.e., applied at the path point level. It was concluded from the simulations included that CGAs have several advantages over conventional GAs when applied to the path generation problems; first, the joints' paths obtained using the conventional GA are found to be of highly oscillatory nature resulting in very large net joints displacements consuming more energy and requiring more time. This problem is totally avoided in CGA where the resulting joints' paths are smooth. Second, the CGA has faster convergence speed (number of generations required for convergence) than the conventional GA. Third, the average execution time per generation in the conventional GA is two to three times that in the CGA. This is due to the fact that the conventional GA requires a coding process, which is not the case in the CGA. Fourth, the memory requirements of the conventional GA are higher than those of the CGA because the former uses genotype and phenotype representations while the later utilizes only the phenotype representation.

1 INTRODUCTION

Genetic algorithms, GAs, are broadly applicable, general-purpose, generate-and-test optimization methods based on Darwinian principles of biological evolution, that is, "the survival of the fittest" and the genetic operators. They were developed by Holland (Holland, 1975) to study the adaptive process of natural systems and to develop artificial systems that mimic the adaptive mechanism of natural systems.

Conventional genetic algorithms were used by the robotics community for solving the path generation problems of robot manipulators where the inverse kinematics problem is formulated as an optimization problem and is then solved using GAs based on the use of the forward kinematics model of the manipulator. In this regard, Parker et. *al.*1989 introduced genetic algorithms for solving the inverse kinematics problem of redundant manipulators where GAs were used to move a robot to a target location while minimizing the largest joint displacement from the initial position. After that, Davidor proposed a special GA for path generation problem of redundant manipulators (Davidor, 1991). He considered generating robot path as a typical ordered-dependent process and presented a GA model for this problem. The main characteristics of his algorithm are the use of dynamic individuals structures and a modified crossover operator called analogous crossover. The goal of the proposed GA is to minimize the accumulative deviation between the generated and the desired path.

CGA has been introduced recently as an alternative and efficient technique for the solution of path generation problems of robot manipulators (Abo-Hammour et *al*, 2002). The CGA is that algorithm which depends on the evolution of curves in one-dimensional space. In general, CGAs use smooth operators and avoid sharp jumps in the parameter values. The algorithm was a contribution to the solution of the inverse kinematics problem of manipulators based on the concept of the minimization of the accumulative path deviation. The effect of various CGA operators and genetic-related control parameters, and the effect of various robot-related parameters on the convergence speed of our proposed methodology for Cartesian path generation was explored in (Abo-Hammour, 2005) and (Abo-Hammour, 2002).

CGAs possess several advantages when applied to path generation problems of robot manipulators (Abo-Hammour et *al*, 2002): first, it can be applied to any general serial manipulator with positional degrees of freedom that might not have any derived closed-form solution for its inverse kinematics. Second, to the authors' knowledge, it is the first singularity-free path generation algorithm that can be applied at the path update rate of the manipulator. Third, extremely high accuracy can be achieved along the generated path almost similar to analytical solutions, if available. Fourth, the proposed approach can be adopted to any general serial manipulator including both non-redundant and redundant systems.

In this paper, a detailed comparative study between conventional and CGAs for the solution of path generation problems of robot manipulators in a free-of-obstacles workspace is performed. This study includes the nature of the joints' paths obtained using both algorithms, the effect of the joints' limits on the solutions obtained using conventional genetic algorithm, the influence of the degree of redundancy and the number of knots along the Cartesian path on the convergence speed of both algorithms, and finally a step by step switching from conventional genetic algorithm to CGA. It is to be noted that both algorithms are based on the concept of the minimization of the accumulative path deviation only; no other objective functions are included in this work.

The organization of the remainder of the paper is as follows: the formulation of the path generation problem for solution by genetic algorithms is described in section 2. Section 3 covers both of the CGA and the conventional genetic algorithm in details. The comparative study between the two algorithms is covered in Section 4. Finally, conclusions are given in Section 5.

2 FORMULATION OF THE PATH GENERATION PROBLEM

Let us consider a robot manipulator with M degrees of freedom and N task space coordinates. Assume that a desired Cartesian path, P_{dc}, is given, the problem is to find the set of joint paths, P_θ, such that the accumulative deviation between the generated Cartesian path, P_{gc}, and the desired Cartesian path, P_{dc}, is minimum. In other words, we are interested in the determination of a set of feasible joint angles, which corresponds to a set of desired spatial coordinates of the end-effector in the task space.

It is to be noted that after the sampling process by N_k samples, P_{dc} and P_{gc} are matrices of dimension N by N_k while P_θ is a matrix of M by N_k dimension. After sampling the geometric path, at the path update rate for best accuracy, the generated values of the joint angles using the genetic algorithm, P_θ, are used by the direct (forward) kinematics model of the robot to obtain the generated Cartesian path given by.

$$P_{gc} = F_k(P_\theta) \tag{1}$$

Where F_k represents the forward kinematics model of the manipulator.

The deviation between the desired Cartesian path, P_{dc}, and the generated Cartesian path, P_{gc}, at some general path point, i, is given as.

$$E(i) = \sum_{k=1}^{N} \left| P_{dc}(k,i) - P_{gc}(k,i) \right| \tag{2}$$

The accumulative deviation between the two paths (desired and generated) depends on whether the initial and final joint angles corresponding to the initial and final configurations of the end-effector are given in advance using the inverse kinematics model of the manipulator or through other numerical technique (fixed end points) or the case in which the initial and final joint angles are not given (free end points). For the fixed end points case, the accumulative deviation between the two paths is given by the formula.

$$E = \sum_{i=2}^{N_k-1} \sum_{k=1}^{N} \left| P_{dc}(k,i) - P_{gc}(k,i) \right| = \sum_{i=2}^{N_k-1} E(i) \tag{3}$$

While for the free end points case, the accumulative deviation between the two paths is given by the formula.

$$E = \sum_{i=1}^{N_k} \sum_{k=1}^{N} \left| P_{dc}(k,i) - P_{gc}(k,i) \right| = \sum_{i=1}^{N_k} E(i) \qquad (4)$$

The fitness function, a nonnegative measure of the quality of individuals, is defined as:

$$F = \frac{1}{1 + E} \qquad (5)$$

The optimal solution of the problem is obtained when the deviation function, E, approaches zero and correspondingly the fitness function, F, approaches unity.

3 GENETIC ALGORITHMS

GAs are based on the triangle of genetic reproduction, evaluation and selection (Goldberg, 1989). Genetic reproduction is performed by means of two basic genetic operators: crossover and mutation. Evaluation is performed by means of the fitness function that depends on the specific problem. Selection is the mechanism that selects parent individuals with probability proportional to their relative fitness. The genetic algorithm used in this work consists of the following steps:

1. ***Initialization.*** An initial population comprising of N_p individuals is randomly generated in this phase.
2. ***Evaluation.*** The fitness, a nonnegative measure of quality used as a measure to reflect the degree of goodness of the individual, is calculated for each individual in the population as given in Equation 6.
3. ***Selection.*** In the selection process, individuals are chosen from the current population to enter a mating pool devoted to the creation of new individuals for the next generation such that the chance of a given individual to be selected to mate is proportional to its relative fitness. This means that best individuals receive more copies in subsequent generations so that their desirable traits may be passed onto their offspring. This step ensures that the overall quality of the population increases from one generation to the next.
4. ***Crossover.*** Crossover provides the means by which valuable information is shared among the population. It combines the features of two parent individuals to form two children individuals that may have new patterns compared to those of their parents and plays a central role in GAs.
5. ***Mutation.*** Mutation is often introduced to guard against premature convergence. Generally, over a period of several generations, the gene pool tends to become more and more homogeneous. The purpose of mutation is to introduce occasional perturbations to the parameters to maintain genetic diversity within the population.
6. ***Replacement.*** After generating the offspring's population through the application of the genetic operators to the parents' population, the parents' population is totally replaced by the offspring's population. This is known as non-overlapping, generational, replacement. This completes the "life cycle" of the population.
7. ***Termination.*** The GA is terminated when some convergence criterion is met. Possible convergence criteria are: the fitness of the best individual so far found exceeds a threshold value, the maximum number of generations is reached, or the progress limit, the improvement in the fitness value of the best member of the population over a specified number of generations is less than some predefined threshold, is reached. After terminating the algorithm, the optimal solution of the problem is the best individual so far found. The block diagram of the genetic algorithm is given in Figure1.

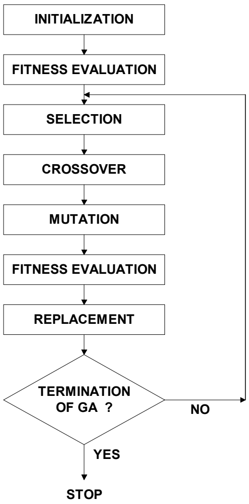


Figure 1: Block Diagram of the Genetic Algorithm.

The conventional genetic algorithm and the CGA used in our work consist of the steps given previously. The evaluation step, selection step, replacement step and the termination step are identical in both algorithms. The differences between both algorithms lie in the initialization phase, the crossover operator, and the mutation operator. These operators have the same goal in both algorithms; the difference lies in the way in which each operator is applied in the corresponding algorithm. These operators are applied at the joint's path level in case of the CGA while they are applied at the path point level in case of conventional genetic algorithm. That is, the operators of the CGA are of global nature while those of conventional genetic algorithm are of local nature. In addition to that, it is to be noted that the conventional genetic algorithm uses the genotype and phenotype data presentations while the CGA uses only the phenotype data presentation. This fact requires a coding process in conventional genetic algorithm, which is not the case in CGA. The CGA is fully described in (Abo-Hammour et *al*, 2002). The reader is kindly asked to read this reference for the complete details about it.

The operators of the conventional genetic algorithm that include the initialization phase, the crossover operator, and the mutation operator are applied at the path point level. In relation to the initialization phase, individuals are generated randomly at the gene level. Conventional crossover involves exchanging genes between each pair of parents selected from mating pool. It is generally applied with relatively high probability of crossover, P_c. Regarding the mutation operator, the bitwise complement mutation is applied in the conventional genetic algorithm at the gene level with some low probability of mutation, P_m. It is realized by performing bit inversion (flipping) on some randomly selected bit positions of children bit strings.

To summarize the evolution process in conventional genetic algorithm, an individual is a candidate solution of the joints' angles; that is, each individual consists of a string of $L=M^* N_k^* N_s$ genes. Initially, N_p individuals are randomly generated representing the initial population. The population undergoes the selection process, which results in a mating pool among which pairs of individuals are crossed with probability P_c. This process results in an offspring's generation where every individual child undergoes mutation with probability P_m. After that, the next generation is produced according to the replacement strategy applied. This process is repeated till the convergence criterion is met where the $M \times N_k$ parameters of the best individual are the required joints' angles.

4 SIMULATION RESULTS

The CGA and the conventional genetic algorithm were used to solve the Cartesian path generation problem of 2R and 3R planar manipulators. The initial settings of the CGA parameters are as follows: the population size is set to 500 individuals. The rank-based selection strategy is used where the rank-based ratio is set to 0.1. The individual crossover probability is kept at 0.9; the joint crossover probability is also set to 0.9. The individual mutation probability and the joint mutation probability are kept at 0.9. Generational replacement scheme is applied where the number of elite parents that are passed to the next generation is one-tenth of the population. The genetic algorithm is stopped when one of the following conditions is met. First, the fitness of the best individual of the population reaches a value of 0.99; that is the accumulative deviation of the end-effector, *E*, of the best individual is less than or equal to 0.01. Second, the maximum deviation at any path point of the best individual is less than or equal to 0.001. Third, a maximum number of 10000 generations is reached. Fourth, the improvement in the fitness value of the best individual in the population over 1000 generations is less than 0.01. It is to be noted that the first two conditions indicate to a successful termination process (optimal solution is found), while the last two conditions point to a partially successful end depending on the fitness of the best individual in the population (near-optimal solution is reached).

The initial settings of the conventional genetic algorithm parameters are similar to those of the CGA except those related to crossover, mutation and coding process which are as following: the crossover probability is kept at 0.7, the mutation probability is kept at 0.01. The uniform crossover method is used as the algorithm's default crossover method. The required accuracy of the phenotype values is set to 0.001 and binary coding scheme is used.

Due to the stochastic nature of GAs, twelve different runs were made for every result obtained in this work using a different random number generator seed; results are the average values whenever possible.

The selected Cartesian path generation problem is of straight line shape as given by:

$$x_{initial} = 0.0, \quad x_{final} = 0.25$$

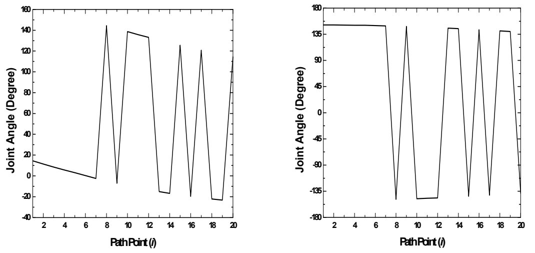
$$P_{dc}(1,i) = X_{dc}(i) = x_{initial} + \frac{x_{final} - x_{initial}}{N_k - 1} * (i-1) \qquad (6)$$

$$P_{dc}(2,i) = Y_{dc}(i) = 0.25, \quad 1 \le i \le N_k$$

Two manipulators are used in this work; 2R planar manipulator and 3R planar redundant manipulator. For the 2R manipulator, the link parameters are $L_1=L_2=L=0.5$ meter. For this case, $N=2$, $M=2$, $\theta_{lower}(h)$ =-180° and $\theta_{upper}(h)$ =180° for $h=1,2$. For the 3R planar redundant manipulator, the link parameters are $L_1=L_2=L_3=0.5$ meter. For this case, $N=2$, $M=3$, $\theta_{lower}(h)$ =-180° and $\theta_{upper}(h)$ =180° for $h=1,2,3$.

The number of path points along the Cartesian path, N_k, is set to 20 points. The initial and final joints' angles corresponding to the initial and final configurations of the end-effector along the Cartesian path are not given (i.e., free end points case).

Initially, the conventional genetic algorithm was used to solve the given path generation problem for both manipulators. For the 2R manipulator, the algorithm reaches a fitness value of 0.99 within 50 generations and the average path point deviation is almost 0.0005 meter. The joints' paths for the first and second joints of the 2R manipulator are shown in Figure 2.

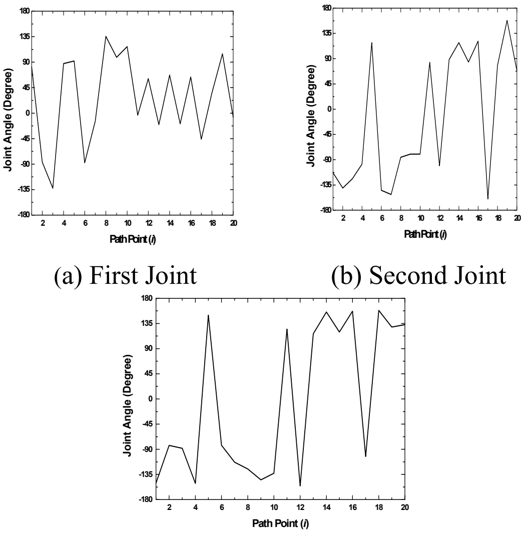


(a) First Joint (b) Second Joint

Figure 2: Joints' Paths of 2R Manipulator Using Conventional Genetic Algorithm for (a) First Joint, and (b) Second Joint.

It is obvious that the resulting solution curves in joint space are highly oscillatory within the given range of the joints' limits. For the given manipulator, there exist two possible solutions for the inverse kinematics problem corresponding to "elbow up" and "elbow down" configurations. It is clear that the resulting solutions for both joints have multiple switching points between these two possible solutions. The switching process from one solution corresponding to one robot configuration to another solution corresponding to other robot configuration results in very large net joints displacements consuming more energy and requiring more time. As a result, while solving such problems, the switching from "elbow up" configuration to the "elbow down" configuration should not be allowed despite the fact that it is still a solution to the problem. Generally, the probability of switching between different solutions increases as the number of feasible solutions of the manipulator increases.

For the 3R planar redundant manipulator, the algorithm reaches a fitness value of 0.99 within 72 generations and the average path point deviation is almost 0.0005 meter. The joints' paths for the first, second and third joints of the 3R manipulator are shown in Figure 3. It is obvious that the resulting joints' paths are highly oscillatory within the range of the joints' limits, which results in large net displacements of the joints.



(a) First Joint (b) Second Joint

(c) Third Joint

Figure 3: Joints' Paths of 3R Manipulator Using Conventional Genetic Algorithm for (a) First Joint, (b) Second Joint, and (c) Third Joint.

The oscillatory behavior of the joints' paths encountered in the conventional genetic algorithm is actually due to the nature of the initialization phase, crossover operator, and mutation operator used in the algorithm. These three operators are applied at the path point level in the conventional genetic algorithm. Conventional initialization phase implies that consecutive path points might have opposite extreme values within the given range of the joint's limits. The problem of oscillatory values among consecutive path points is emphasized when the range of joint's limits is extended as discussed previously. This problem is bypassed in CGA by the use of smooth curves in the initial population that eliminate the possibility of highly oscillating values among the consecutive path points.

Conventional crossover operator results in a jump in the value of the parameter in which the crossover point lies (discontinuity) while keeping

Table 1: Step-by-Step Switching to CGA for the 2R Manipulator.

Initialization Type	Crossover Type	Mutation Type	Avg. Execution Time (Seconds)	Avg. No of Generations	Avg. No. of Switchings
Conventional	Conventional	Conventional	143.99	54	9
Conventional	Conventional	Continuous	160.59	57	7
Conventional	Continuous	Conventional	221.2	78	7
Conventional	Continuous	Continuous	261.44	100	2
Continuous	Conventional	Conventional	117.37	48	1
Continuous	Conventional	Continuous	109.97	49	0
Continuous	Continuous	Conventional	119.09	55	0
Continuous	Continuous	Continuous	99.11	47	0

Table 2: Step-by-Step Switching to CGA for the 3R Manipulator.

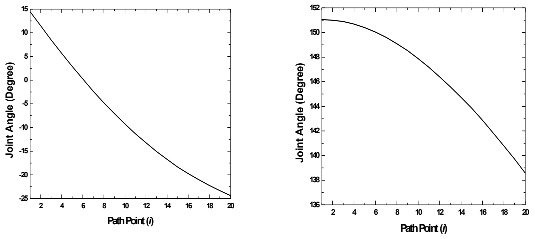
Initialization Type	Crossover Type	Mutation Type	Avg. Execution Time (Seconds)	Avg. No. of Generations	Nature of Joints' Paths
Conventional	Conventional	Conventional	487.72	124	Oscillations With Large Magnitude
Conventional	Conventional	Continuous	390.06	105	Oscillations With Medium Magnitude
Conventional	Continuous	Conventional	295.14	83	Oscillations With Large Magnitude
Conventional	Continuous	Continuous	486.1	144	Oscillations With Medium Magnitude
Continuous	Conventional	Conventional	188.89	53	Oscillations With Small Magnitude
Continuous	Conventional	Continuous	191.2	56	Oscillations With Small Magnitude
Continuous	Continuous	Conventional	181.57	55	Oscillations With Small Magnitude
Continuous	Continuous	Continuous	148.85	49	Smooth Solution Curves

the other parameters the same or exchanged between the two parents. It is clear that each crossing point results in a discontinuity in the joint angles of the obtained children. The worst case obtained regarding the discontinuity of the resulting curves of the children happens in the uniform crossover process. In this scheme, the smoothness of the joint's paths of the parents is completely spoiled since crossover happens at every path point. The solution to the non-smoothness of the resulting joint's paths is through the use of the tangent hyperbolic crossover function used in CGA that results in smooth transition in the joint values of the two parents while generating the two children.

Conventional mutation process changes only the value of the joint angle of the path point in which mutation occurs while keeping other joint angles in the joint's path unchanged. This process results in a jump in the value of the joint angle in which mutation takes place and the overall path will become of oscillatory behavior. The discontinuity in the joint's path depends on the number of mutations that take place in the path and the position of the bit at which mutation takes place; that is, if the mutation bit is leftmost, then the discontinuity will be larger than that of rightmost mutation bit. This problem is solved in CGA by applying the Gaussian mutation function that is of global nature. In our approach, mutation is applied at the joint's path level rather than path point level. As a result, mutation function will start from zero values and increases/decreases slowly till the peak then it will go back to zero values at the other end.

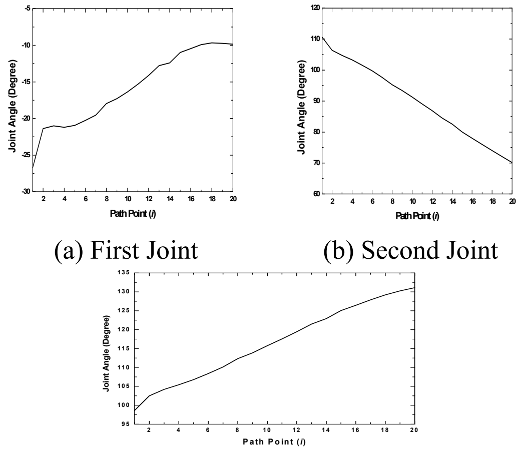
After that, the effect of both versions (conventional and continuous) of the initialization phase, crossover operator and mutation operator on the nature of the joints' paths obtained and the convergence speed of the hybrid algorithm is studied. Table 1 gives the relevant data for the 2R manipulator while Table 2 gives the relevant data for the 3R manipulator. From Table 1, it is clear that the maximum number of switching between the two existing solutions of the inverse kinematics problem for the 2R manipulator happens in case of the conventional genetic algorithm (i.e., conventional types of initialization, crossover and mutation). Furthermore, the initialization phase has the greatest effect on the smoothness/non-smoothness of the solution curves; that is, in case of conventional initialization, the number of switching points is 6 on average while in case of continuous initialization, the number of switching points is 0 on average. It is also clear that as the number of conventional processes decreases, the number of switching points decreases. The minimum execution time and the best convergence speed are achieved using the CGA (i.e., continuous types of initialization, crossover and mutation). Regarding the 3R manipulator, it is clear that the initialization phase has the greatest effect on the smoothness/non-smoothness of the solution curves; that is, in case of conventional initialization, the joints' paths are of oscillatory nature with large or medium magnitude oscillations while in case of continuous initialization, the joints' paths are either smooth or of oscillatory nature with small magnitude oscillations. The minimum execution time and the best convergence speed are achieved using the CGA (i.e., continuous types of initialization, crossover and mutation). For both manipulators, the conventional initialization, continuous crossover and continuous mutation case results in the largest number of generations required for convergence. Regarding the case in which the conventional initialization, continuous crossover and conventional mutation are used which is almost similar to the algorithm proposed by Davidor, it is observed that this hybrid scheme still results in oscillations with large magnitude as shown in Table 2. This is an expected result since the smoothness achieved by the continuous crossover process is disturbed by the conventional mutation process. This goes in agreement with our previous comments about his algorithm that even after the application of the analogous crossover operator, the oscillatory behavior of the joints' paths is not totally avoided due the discontinuities, which might appear in the initialization phase and due to the mutation operator.

The joints' paths for the first and second joints of the 2R manipulator using CGA are shown in Figure 4. It is obvious that the resulting solution curves in joint space are smooth and do not have any switching between the two possible solutions, which results in minimizing the net displacement of the joints. The joints' paths for the first, second and third joints of the 3R manipulator are shown in Figure 5 where similar observations are concluded regarding the smoothness of the solution curves.



(a) First Joint (b) Second Joint

Figure 4: Joints' Paths of 2R Manipulator Using CGA for (a) First Joint, and (b) Second Joint.



(a) First Joint (b) Second Joint

(c) Third Joint

Figure 5: Joints' Paths of 3R Manipulator Using CGA for (a) First Joint, (b) Second Joint, and (c) Third Joint.

5 CONCLUSIONS

In this work, both of the continuous and the conventional genetic algorithms were used for the solution of the Cartesian path generation problems of robot manipulators.

It was noted that the resulting joints' paths using conventional genetic algorithm have multiple switching points among the possible solutions of the non-redundant manipulators while they are of highly oscillatory nature for the redundant manipulators resulting in very large net displacements of the joints for both systems. This oscillatory behavior in conventional genetic algorithm is actually due to the nature of the initialization phase, crossover operator, and mutation operator used in the algorithm. First, the conventional initialization phase results in consecutive path points that might have opposite extreme values within the given range of the joints' limits. Second, the conventional crossover operator results in a jump in the value of the parameter in which the crossover point lies, while keeping the other parameters the same or exchanged between the two parents. Third, the conventional mutation process changes only the value of the joint angle of the path point in which the mutation occurs while keeping other joint angles in the joint's path unchanged. The resulting discontinuity in the joint's path depends on the number of mutations that take place in the path and the position of the bit at which mutation takes place. These three operators are designed in CGA such that they result in smooth joints' paths from one side and maintain an excellent accuracy along the Cartesian path from the other side. Among the three operators, it was noted that the initialization phase has the greatest effect on the smoothness/non-smoothness of the joints' paths. The convergence speed of the CGA in terms of both the number of generations required for convergence and the average execution time is much superior to that of the conventional genetic algorithm.

REFERENCES

Abo-Hammour, Z., Mirza, N., Arif, M., 2002. Cartesian Path Planning of Robot Manipulators Using Continuous Genetic Algorithms. In *Robotics and Autonomous Systems*; 41(4):179-223.

Abo-Hammour, Z., 2005. A Novel Continuous Genetic Algorithms For the Solution of the Cartesian Path Generation Problem of Robot Manipulators. In *Robot Manipulators.*

Abo-Hammour, Z., 2002. Advanced continuous genetic algorithms and their applications in the motion planning of robotic manipulators and the numerical solution of boundary value problems; *Ph.D. thesis*.

Davidor, Y., 1991. Genetic Algorithms and Robotics: A Heuristic Strategy for Optimization, *World Scientific*.

Goldberg, D., 1989. Genetic Algorithms in Search, Optimization, and Machine Learning, Addison - Wesley, New York.

Holland, J., 1975. Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor.

Parker, J., Koogar, A., and Goldberg, D., 1989, Inverse Kinematics of Redundant Robots Using Genetic Algorithms, In *IEEE Conference on Robotics and Automation.*

A MONOCULAR OCCUPANCY GRID FOR LOCAL WMR NAVIGATION

Lluís Pacheco, Xavier Cufí
Technical School, Girona University, Av. Luís Santaló s/n, Girona, Spain
{lluispa, xcuf}@eia.udg.edu

Ningsu Luo, Javier Cobos
Institut de Informàtica i Aplicacions, Girona University, Av. Lluís Santaló s/n, Girona, Spain
ningsu@eia.udg.edu

Keywords: Autonomous mobile robots, Computer vision, Path planning, Obstacle avoidance, Reactive navigation.

Abstract: This work introduces a new methodology to infer environment structure by using monocular techniques. The monocular field of view is constrained to the vicinity of the mobile robot. The cooperative strategy proposed combines DFF and qualitative structure techniques to obtain environment information. The remarkable features of the strategy presented are its simplicity and the low computational cost. In this way, a simplified DFF method, which uses only one frame, has been implemented; hence, scenario information can be achieved when homogeneous radiance background constraint is accomplished. Further structure analysis is developed by computing qualitative structure through time integration series of acquired frames; within a tessellated probabilistic representation consisting in a local occupancy grid framework. Furthermore, the camera pose knowledge is used to correlate the different overlapping image zones. Moreover, time integration of the monocular frames allows a larger environment description suitable for WMR local path planning. Hence, the reported work can be used in obstacle avoidance strategies or reactive behaviours for navigation towards the desired objective.

1 INTRODUCTION

Perception of the environment is based on the sensor system measurements that provide distances and structure knowledge. This essential task could be accomplished by different range systems like ultrasonic sensors, laser rangefinders, or vision based systems. All these sensors have their advantages and disadvantages. However, computer vision based methods, are one of the most attractive. Therefore, they have many interesting advantages that can be summarized as follows: the falling prices of devices and richer information compared with the other traditional ranging devices. In this way the increasing capabilities of the personal computers, offer an interesting range of real time applications. Perception systems based on camera devices have attracted robotic research due these interesting features. Thus, machine vision systems have used some features of eyes, such as stereopsis, optical flow or accommodation, as meaningful clues. SVS (stereo vision systems), OFM (optical flow methods) and DFF (depth from focus) are all methods that permit 3D scene recovery. Studies comparing SVS and DFF are reported in (Schechner and Kiryati, 1998). The results show that while SVS has greater resolution and sensitivity, DFF has greater robustness, requires less computational effort and can deal properly with correspondence and occlusion problems. The need for several images of the same scene, acquired with different optical setups, may be considered as a significant drawback in using DFF methods in major robotic applications. The scientific community has proposed the use of special cameras, such as a multi-focus camera that acquires three images with three different focus positions (Hiura and Matsuyama, 1998). However, other proposals were developed due to a lack of multi-focus commercial cameras. The use of DFF in WMR (wheeled mobile robots) has been reported in (Nourbakhsh, 1997); in which Noubakhsh used three cameras with almost the same scene to achieve robust and efficient obstacle detection.

This work presents a new cooperative monocular strategy; where DFF and QSM (Qualitative

Structure Methods) are combined. Thus, one bit depth can be obtained using the DFF methodology as well as a set of multi-resolution focus thresholds, when homogeneous radiance background constraint is accomplished. However, when homogeneous radiance constraint fails, we propose to use QSM over discrepancy areas in order to infer environment structure by using an occupancy grid framework. Therefore, the main contributions of this research are to propose the occupancy grid as a suitable structure in order to infer qualitative obstacle structure and obtaining larger scenario descriptions. The results depicted are directed towards real applications by using the WMR PRIM, which consists of a differential driven one with a free rotating wheel (Pacheco et al., 2008). The experiments are orientated so as to obtain a local map in the robot's neighborhood that can be used to plan navigation strategies.

This paper is organized as follows. In Section 1, the main ideas and research objectives are presented. Section 2 introduces the DFF methodology as well as the algorithms and results implemented. Section 3 depicts the QSM concept and the related algorithms used. In this way, the local occupancy grid framework is also formulated as a way for time integrating the acquired frames. In Section 4, the experimental preliminary results are drawn by using the mobile platform PRIM. In Section 5 the conclusions and future research are presented.

2 THE CONSTRAINED DFF SYSTEM DESCRIPTION

This section briefly introduces the DFF methodology. The algorithms implemented as well as their results are also depicted by using the available WMR platform. Its significant contribution is the use of a single image to obtain environment information.

The DFF techniques use an image collection of the same scene acquired at different focus positions. Thus, the camera system PSF (point spread function) for unfocused object points produces blurred image points. The PSF frequency domain space transform representations arise in a first order Bessel OTF (optical transfer function), where its main lobe volume can determine the FM (focus measure) expressed as:

$$M_0 = \iint \left| I_i(\omega, \nu) \right| d\omega d\nu \,. \tag{1}$$

where I_i denotes the image considered, ω and ν represent the frequency components. Efficient energy image measures have been proposed as FM (Subbarao et al, 1992). Nayar has proposed a modified Laplacian that improves the results in some textures (Nayar and Nakagawa, 1994). The 3D scene map and passive auto-focus consumer camera systems are interesting applications solved by the DFF. Recovering the 3D information from DFF methods is known as SFF (shape from focus) (Nayar and Nakagawa, 1994).

2.1 The DFF Monocular Algorithms

The algorithms of the machine vision system implemented are based on important assumptions that are generally obtained in normal indoor scenarios, but also in many outdoor scenarios. These constraints are flat and homogenous energy radiance from the floor surface and experimental knowledge of the focus measurement threshold values. Two important aspects, image window size and camera pose, should be considered. The size of windows should be big enough to receive energy information. For example, in the work of Surya, images of 150x150 pixels were used, and the focus measures were computed in 15x15 pixel regions (Surya, 1994). The camera pose will set the scenario perspective and consequently the floor position coordinates that should be used in the WMR navigation strategy. Figure 1 shows the robot and camera configuration considered in this work, whereα, β and φ are angles of the vertical and horizontal field of view and the tilt camera pose respectively. The vertical coordinate of the camera is represented by *H*. The robot coordinates corresponding to each pixel can be computed using trigonometric relationships and the corresponding knowledge of the camera configuration (Horn, 1998). Using trigonometric relationships, the flat floor scene coordinates can be computed as follows:

$$\begin{aligned} X_{i,j} &= \pm \frac{H}{\cos\left(\varphi - \alpha/2 + \Delta\alpha\right)} \tan\left(\Delta\beta\right) \\ Y_j &= H \tan\left(\varphi - \alpha/2 + \Delta\alpha\right) \\ \Delta\beta &= K_i \frac{\beta}{C} \qquad \left(0 \le \mathrm{K_i} \le C/2\right) \\ \Delta\alpha &= K_j \frac{\alpha}{\mathrm{R}} \quad \left(0 \le \mathrm{K_j} \le R\right) \end{aligned} \tag{2}$$

K_i and K_j are parameters used for covering the discrete space of the image pixels. Thus, *R* and *C* represent the image resolution through the total number of rows and columns. It should be noted that for each row position corresponding to scene coordinates Y_j, there are C column coordinates $X_{i,j}$. The above equations provide the available local map coordinates when no obstacle is detected. The algorithms used are explained in the remainder of this subsection. The multigrid representation using low-pass filtering processes can improve the surface radiance homogeneity. Scale space representations can reduce the search space, increasing the computation performance (Gonzalez and Woods, 2002). Therefore, a Gaussian filter is applied to the frames acquired in PAL format, at 768x576 pixels. Three decreasing resolution levels have been used with picture sizes of 384x288, 192x144 and 96x72. The average image brightness is also computed. In order to achieve greater robustness against changes in lightness, brightness normalization is performed (Surya, 1994). The image energy is computed, over 3x3 windows at the top level of the scale-space, using the modified Laplacian method:

$$ML(x,y) = \left|2i(x,y) - i(x-1,y) - i(x+1)\right| + \left|2i(x,y) - i(x,y-1) - i(x,y+1)\right| \tag{3}$$

where $i(x,y)$ represents the corresponding pixel value at spatial coordinates (x,y). The 96x72 scale-space is decreased using a 9x7x2 array, where each cell represents the Laplacian mean value and the corresponding standard deviation mean computed over 10x10 pixel patches. Another interesting statistical parameter that has been used is the standard deviation, which relates to the homogeneity of the floor energy values. The 9x7x2 array is explored, from top to bottom; floor segmentation is carried out, using both energy and standard deviation thresholds.

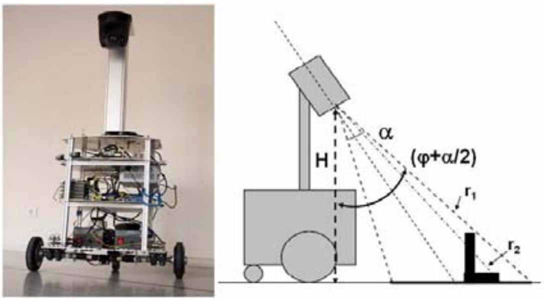


Figure 1: The robot PRIM and the monocular camera configuration. Where α is set to 37º, β (horizontal angle) of 48º, *H* set to 109cm, and a tilt angle of 32º.

2.2 One Bit DFF Experimental Results

The floor threshold has been experimentally computed by averaging several floor images acquired in our lab environment with different kinds of illumination (from 200-2000 lx). Light illumination can change from 2000 lx when light from the sun is clearly present where there is sunlight through the windows, to less than 200 lx in the darker corridor zones. Figure 2 depicts high resolution (130x130 pixel windows) corresponding to different floor images used to compute focus measurement thresholds where the floor texture is clearly visible. It is in the locality of those points where the information about radiance is obtained. Hence, the results obtained with the available experimental set up show the decreasing values when the distance between the camera and the floor is increased. A more complete description of the energy floor radiance measures obtained for each 9x7 visual perception row is shown in (Pacheco et al., 2007); in which the image perspective emerges from a set of multi-resolution thresholds as a function of the camera distances.

Figure 2: Fragments of high resolution floor images (768x576 pixels under different light conditions corresponding to 300, 800, 1400 and 2000 lx, respectively.

Figure 3 shows the modified Laplacian energy and standard deviation values using 9x7 and 96x72 space-resolutions, when typical indoor obstacles are presented. It is shown that 9x7 space resolutions can detect radiance discontinuities but because there was

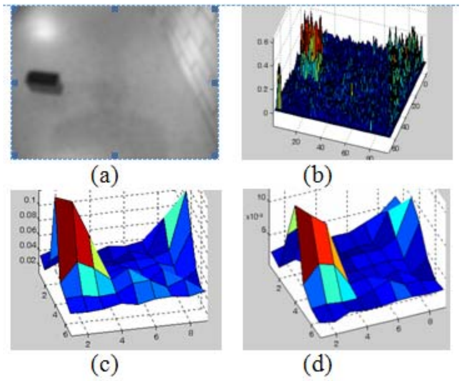


Figure 3: (a) Image with obstacles, 96x72; (b) Modified Laplacian measures; (c) 9x7 Modified Laplacian mean values; (d)) 9x7 standard deviation mean.

a great lack of resolution manifested through soft slopes. Thus, it is necessary to use a fine space-resolution to attain the sharper edges. In this work, 9x7 resolutions are used to detect the local patches where obstacle segmentation is computed within 96x72 space resolution.

Despite the good results achieved, some further improvements should be considered. Hence, when radiance floor discontinuites occur they can be considered obstacles (false positives in some cases). Therefore in next section OFM is introduced, within the occupancy grid framework, to improve the one bit DFF methodology.

3 QUALITATIVE STRUCTURE METODS AND OCCUPACY GRID

The camera pose and local field of view will set the QSM algorithms reported in this section. The different optical flow quantitative approaches are generally based on two classical feasible assumptions, which are BCM (brightness constancy model), and optical flow smoothness. Thus, image motion discontinuities are due to the depth and motion discontinuity boundaries. Hence, there are places where image flow changes are suddenly useful as image segmentation clues, but can cause problems such as optical flow estimation clusters. Therefore, suggestions made to compute the algorithms over small neighborhoods, or region-based matched methods have turn on. Combining local and global optic flow differential methods have been proposed as a way to share benefits from the complementary advantages and short-comings (Bruhn, 2002).

The occupancy field can be depicted by a probability density function that relates sensor measures to the real cell state. The tessellated probabilistic representation has been widely adopted by the scientific community in navigation or mapping issues. Indoor applications research has been mainly concentrated on SLAM (simultaneous localization and mapping) issues (Thrun, 2002). Their use allows sensor fusion or multiple layer representations to segment dynamic objects (Coue, 2006). The perception system used, in this work, consists in monocular and odometer system data. The use of these systems in SLAM is reported in (Cumani et al., 2004).

The main difference of the research depicted in this paper, as compared with Cumani research, is the occupancy grid use that allows integration of multiples frames without constraining their number. Furthermore, it is obtained a local map description suitable for navigation. Thus, the occupancy grid developed research increase the camera narrow field of view, which provides just the vicinity of the robot where floor only is expected to be found. Moreover, the floor model is also proposed as a contribution in order to build the 2D occupancy grid; hence obstacle binary results are time integrated within the local occupancy grid framework by considering such model. The obstacle structure could be inferred by considering optical flow magnification change discrepancies from the floor model.

3.1 The Local QSM Approach

In the present research the camera field of view depicts only the vicinity of the WMR. Perspective projection, as shown in Figure 4, should be assumed.

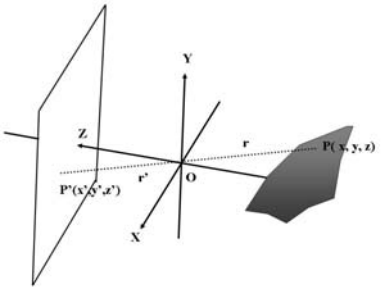


Figure 4: Camera system producing an image that is a perspective projection of the world.

Introducing the coordinate system, where *z* coordinates are aligned to the optical camera, and the xy-plane is parallel to the image plane, the image *P'* corresponding to the point *P* of a scene object is given by the following expressions:

$$\begin{gathered} \frac{x'}{z'} = \frac{x}{z} \quad \frac{y'}{z'} = \frac{y}{z} \qquad -z = r\cos\alpha \\ \Rightarrow r = -\frac{z}{\cos\alpha} = -z\sec\alpha \\ z' = r'\cos\alpha \Rightarrow r' = \frac{z'}{\cos\alpha} = z'\sec\alpha \end{gathered} \tag{4}$$

Where *z'* in the distance between image plane and the camera lenses, and *x'* and *y'* are the image coordinates. The object point coordinates, referred to the optic center *O*, are given by *P= (X, Y, Z)*, being *r* de distance between *P* and *O* and α the angle. The ratio of the distance between two points measured in the image plane and the corresponding points measured in the scene is called magnification *m*.

$$m = \frac{\sqrt{(\Delta' x)^2 + (\Delta' y)^2}}{\sqrt{(\Delta x)^2 + (\Delta y)^2}} = \frac{z'}{z} \qquad (5)$$

For reduced field of views when the optical rays are parallel to the optical axis the magnification *m* is constant for all the image points. However, due to the field of view and camera pose assumed in this research, magnification changes are expected even when just considering a flat floor scenario. Hence, the perspective image formation model arises in magnification changes. Figure 5 shows the magnification changes of the floor model by considering the optical axis ray as the unit of magnification. Therefore, these changes in magnification are made it more complicated to look for image patches with similar motion in order to detect obstacle depth boundaries. However, by using the floor model and the odometer system data, binary floor results can be predicted from frame to frame; then predicted discontinuities arise due to the 3D non floor obstacle shapes that produce unexpected image boundaries.

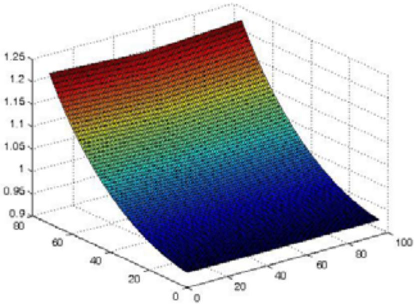


Figure 5: Magnification changes of the floor model by considering the optical axis ray as the unit of magnification**.** The bigger magnification is attained at closer robot positions.

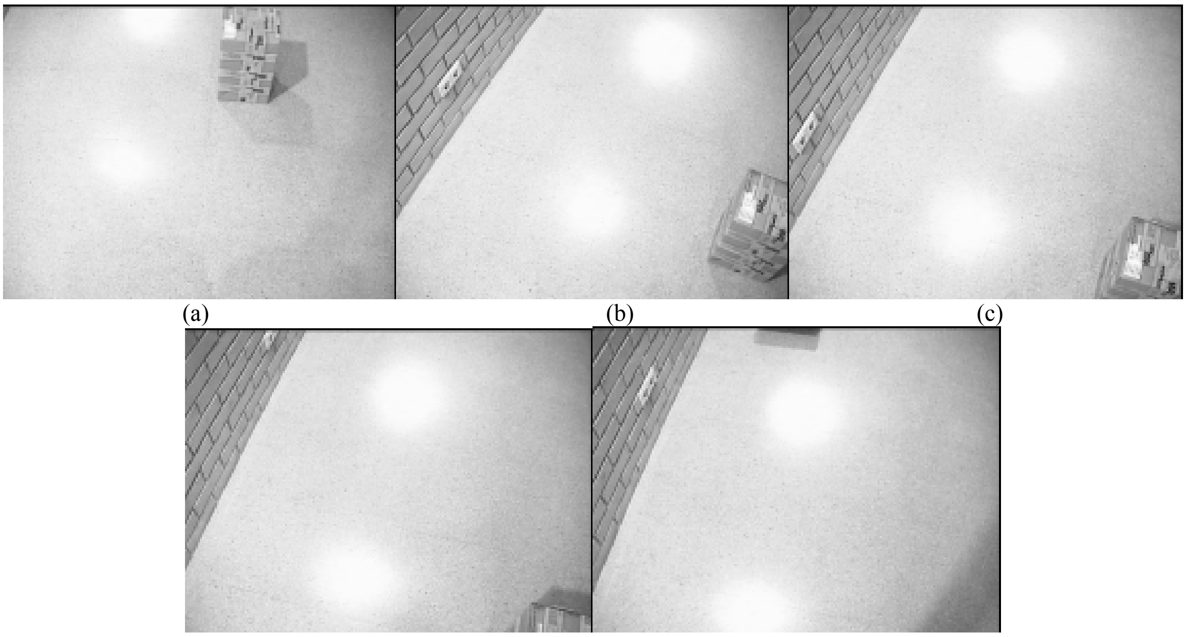
The machine vision system algorithms implemented are based on binary results obtained by the one bit DFF algorithms explained in subsection 2.2. Binary images are obtained in the 96x72 space resolution level, and blob analysis is developed. The blob areas and the extremes of their coordinates are computed and small blobs are removed. Then, the image is analyzed from top to bottom, searching for possible non floor regions. Hence, the QSM can be used to detect the possible obstacles, when important floor energy radiance discrepancies are met. Therefore, using relative robot coordinate increments provided by the odometer system, qualitative structure estimation could be done by comparing predicted positions with the binary results obtained. The time integration of the different frames acquired is introduced in the next section. Thus, the robot coherent interaction with the world can be addressed by using the occupancy grid framework that provides a robust and unified approach to a variety of problems in spatial robot perception and navigation (Elfes, 1989).

3.2 The Local Occupancy Grid Framework

The occupancy grid is considered to be a discrete stochastic process defined over a set of continuous spatial coordinates *(x, y)*. Hence, the space is divided into a finite number of cells representing a 2D position, $1 \le j \le R$ $1 \le i \le C$. The *R* and *C* parameters are the number of rows and columns of the grid respectively. The cell column coordinates are designated by X_i and the rows by Y_j. It is assumed that local occupancy grid data is provided by the on-robot perception system. The occupancy probability is divided into two ranges only: free and occupied. The grid can be updated by using the sensor models and the current information. Hence, given a sensor measurement *m*, the occupancy probability *P(O)* for the different cells, $P(C_{ij})$, can be computed by applying Bayes rule:

$$P(O|C_{ij}) = \frac{P(C_{ij}|O)}{P(C_{ij}|O) + P(C_{ij}|/O)} \qquad (6)$$

Hence, the probability that a cell is occupied $P(O/C_{ij})$ is given by the cell occupancy sensor measurement statistics $P(C_{ij}/O)$ by also considering the probability that the cell will be free $P(C_{ij}//O)$. Thus, free cells have binary results equal to zero; these non-occupied cells belong to coordinates for image pixels within floor radiance thresholds. Other available coordinates are provided through time integration of the acquired frames when radiance energy is bigger than threshold values, by using the floor model, and looking for coincidences with the acquired frames. The unknown probability value is set to 0.5. Therefore, by using the expression (6) with the predicted occupied cells and acquired frames, the grid positions belonging to the floor will provide larger occupancy values. Obstacle positions give intermediate occupancy probabilities due to the discrepancies between the predicted and the acquired image values that arise due to the 3D obstacle shape. Next section depicts some preliminary results experimented with the available WMR platform.



(a) (b) (c)

(d) (e)

Figure 8.a, 8.b, 8.c, 8.d, and 8.e: It is depicted the monocular frames acquired while the WMR is avoiding the first obstacle placed on the direction towards the objective.

4 ON-ROBOT EXPERIENCES

In this section are presented some experimental results using the WMR PRIM. Thus, local navigation with static obstacles is used as a preliminary test of the research introduced in this paper. The navigation and control strategy used, under this reduced field of view, is introduced in other author's work (Pacheco and Luo, 2007). Therefore, the maximum geometric size of the closer obstacle is considered in order to plan safety navigation towards the desired coordinates.

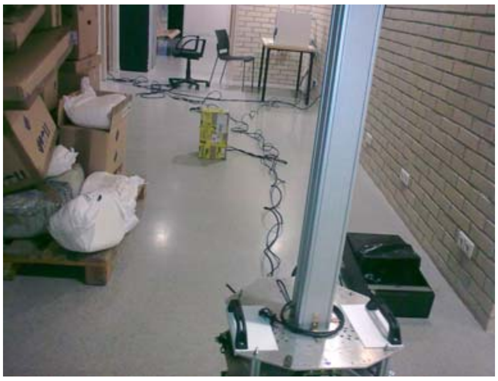


Figure 6: It is presented the real scenario that has been drawn in Figure 7. It is shown the obstacles placed on the floor that the WMR should avoid.

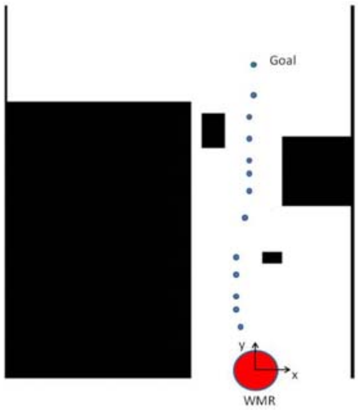
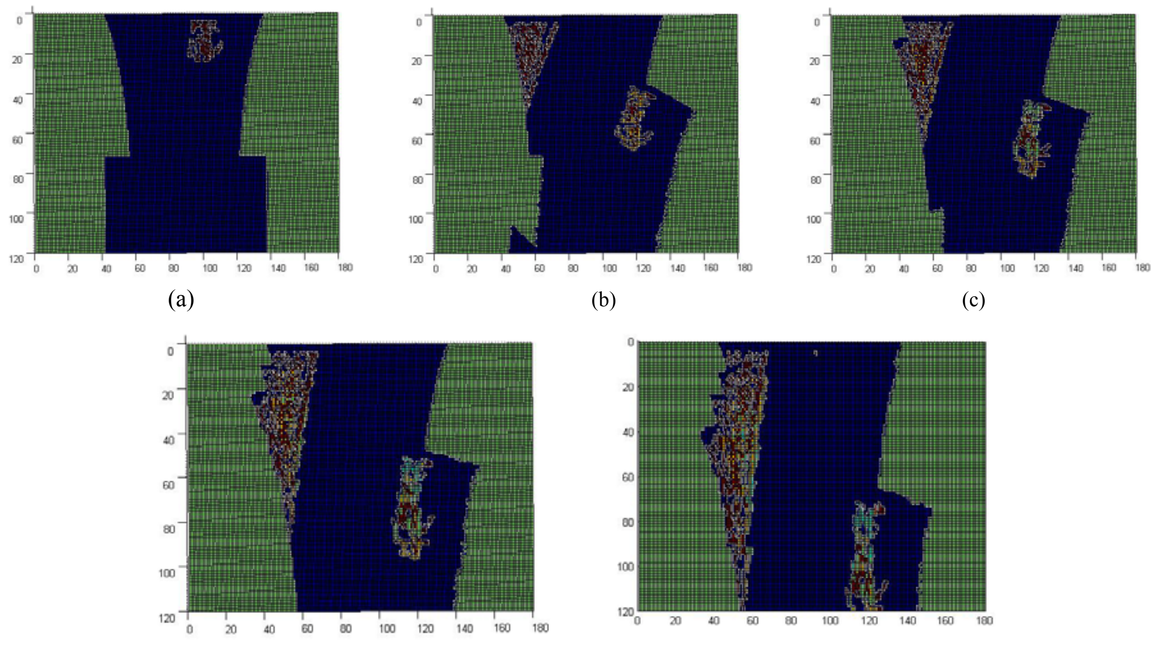


Figure 7: Simplified map scenario where the robot trajectory toward the goal is depicted with blue dots. The obstacles are drawn in black.

Figure 6 shows the scenario where the experiment has been done, and Figure 7 depicts the simplified map scenario with the WMR achieved trajectory.

Thus, Figure 7 shows the lab environment map and the path followed when the WMR starts at the position (0, 0, 90º) towards the desired coordinates (0, 460cm). The scenario contains some static obstacles that the WMR should avoid.

Table 1 depicts the robot coordinates and acquired frames during the WMR navigation.



(a) (b) (c)

(d) (e)

Figure 9: Sequence of occupancy grids obtained by integrating the first 5 acquired frames.

Table 1: Coordinates from where the frames are acquired.

F.1	(0, 0, 90º)	F.7	(1, 293, 76º)
F.2	(-11, 66, 110º)	F.8	(4, 315, 88º)
F.3	(-21, 108, 99º)	F.9	(4, 339, 96º)
F.4	(-25, 141, 97º)	F.10	(6, 375, 89º)
F.5	(-26, 176, 94º)	F.11	(4, 415, 100º)
F. 6	(-8, 248, 83º)	F.12	(12, 456, 74º)

The first 5 frames are acquired during the obstacle avoidance strategy of the first obstacle placed in the middle of the corridor. Figure 8.a, 8.b, 8.c, and 8.d. show these frames.

The first obstacle avoidance strategy consists into turn to the left in order to avoid the collision with the obstacle that appears at the first four frames. It is noted that fifth frame, Figure 8.e, depicts a part of the scenario where doesn't appear the first obstacleThe local occupancy grid built by integrating the first 5 frames is shown in Figure 9.a, 9.b, 9.c, 9.d, and 9.e. It is depicted that in the first frame only the front obstacle is perceived. However, when the other frames are integrated the left wall is integrated. It is observed how the WMR navigation is constrained by the different obstacles obtained on the acquired frames and integrated within the occupancy grid. Therefore the navigation is constrained by both obstacles. Moreover the fifth frame is integrated in Figure 9.e where appear as time integrated the front obstacle and the left wall. Hence, the monocular occupancy grid methodology presented increases the field of view perception, and a better navigation strategy can be planned. The integration of multiple monocular frames also can be used as a framework in order to infer 3D obstacle structure.

5 CONCLUSIONS

The methodology presented in this research has provided a local map suitable for WMR navigation. Therefore a short-term memory has been obtained. Navigation advantages by using short-term memory were reported in previous research (Schäfer et al., 07). However, experimental results conducted to obtain the obstacle structure have some aspects that should focus the future work. The obstacle shape is larger than the real shape due to the magnification changes that arise of perspective. The lack of accuracy increases the path-width, and consequently this can result in larger trajectories or even infeasible path perceptions where available paths are possible. 3D obstacle structure can solve the above problem. But, the results obtained in order to obtain 3D information have some mismatches when overlapping areas between predicted and obtained blobs are analysed. The errors can be produced by the following sources:

- Odometry errors.
- Camera calibration errors.

- Flat floor model differences.

Future work will be addressed to solve the above problems. We believe that the occupancy grid framework can be used to obtain 3D obstacle structure. Therefore, there is not limitation concerning to the number of frames that can be time-integrated. The future goal will consist in to find a set of parameters in order to infer 3D obstacle structure. These set of parameters should be independent of the source of errors pointed in this section. The knowledge of 3D structure can afford several benefits that can be summarised as follows:

- To reduce the trajectories.
- Visual Odometry.
- Landmark detection.

Despite the work that remains undone the methodology presented can be used to direct the future research. Moreover, some good features and results are presented in this work.

ACKNOWLEDGEMENTS

This work has been partially funded by the Commission of Science and Technology of Spain (CICYT) through the coordinated project DPI-2007-66796-C03-02, and by the Government of Catalonia through the Network Xartap and the consolidated research group's grant SGR2005-01008.

REFERENCES

Bruhn A., Weickert J., Schnörr C., 2002. Combining the advantages of local and global optimal flow methods, *In Proc. Pattern Recognition, Lect. Notes in Comp. Science*, Springer-Verlag, 454-462.

Coue, C., Pradalier, C., Laugier, C., Fraichard, T., Bessiere, P., 2006. Bayesian Occupancy Filtering for Multitarget Tracking: An Automotive Application. *The Inter. Journal of Robotics Research*, 25(1) 19-30.

Cumani A., Denasi S., Guiducci A., Quaglia G., 2004. Integrating Monocular Vision and Odometry for SLAM. *WSEAS Trans. on Computers*, 3(3) 625-630.

Elfes, A., 1989. Using occupancy grids for mobile robot perception and navigation. *IEEE Computer,* 22(6) 46-57 .

Gonzalez, R. C., Woods, R. E., 2002. Digital Image Processing, *Prentice Hall Int. Ed*., Second Edition.

Hiura, S., Matsuyama, T., 1998. Depth Measurement by the Multi-Focus Camera, *Proc. IEEE CVPR*, 953-959.

Horn, B. K. P., 1998. Robot Vision, McGraw-Hill Book Company, MIT Press Edition, 12th printing.

Nayar S.K., Nakagawa, Y., 1994. Shape from Focus, *IEEE Trans. PAMI*, 16(8), 824-831.

Nourbakhsh, I., Andre, D., Tomasi, C., Genesereth, M.R., 1997. Mobile Robot Obstacle Avoidance via Depth from Focus, *Robotics and Autom. Systems*, Vol. 22, 151-158.

Pacheco, L., Luo, N., 2007. Trajectory Planning with Control Horizon Based on Narrow Local Occupancy Grid Perception. *Lect. Notes in Control and Inform. Sciences 360*, Springer-Verlag, pp. 99-106.

Pacheco, L., Cufí, X., Cobos, J., 2007. Constrained Monocular Obstacle Perception with Just One Frame, *Lect. Notes in Comp. Science,* Springer-Verlag, Pattern Recog. and Image Analysis, Vol. 1, 611-619.

Pacheco, L., Luo, N., Ferrer, I., Cufí,, X., 2008. Control Education within a Multidisciplinary Summer Course on Applied Mobile Robotics, *Proc. 17th IFAC World Congress,* pp. 11660-11665.

Schäfer H., Proetzsch M., Berns K., 2007. Obstacle Detection in Mobile Outdoor Robots, *Proc. Inter. Conf. on Informatics in Control, Autom. and Robotics*, pp. 141-148.

Schechner, Y., Kiryati, N., 1998. Depth from Defocus vs. Stereo: How Different Really Are They?, *Proc. IEEE CVPR,* Vol. 2, 256-261.

Subbarao, M., Choi, T., Nikzad, A., 1992. Focusing Techniques, *Tech. Report 92.09.04*, Stony Brook, NewYork.

Surya, G., 1994. Three Dimensional Scene Recovery from Image Defocus. *PHD thesis*, Stony Brook, New York.

Thrun S., 2002. Robotic mapping: a survey. Exploring Artificial Intelligence in the New Millennium*, Morgan Kaufmann*, San Mateo.

SEGMENTS OF COLOR LINES
A Comparison through a Tracking Procedure

Michèle Gouiffès, Samia Bouchafa
Institut d'Electronique Fondamentale, UMR 8622, Université Paris Sud 11, France
michele.gouiffes@u-psud.fr, samia.bouchafa@u-psud.fr

Bertrand Zavidovique
Institut d'Electronique Fondamentale, UMR 8622, Université Paris Sud 11, France
bertrand.zavidovique@u-psud.fr

Keywords: Computer vision, Color image processing, Level lines, Color lines, Segments features, Tracking, matching.

Abstract: This paper addresses the problem of visual target tracking by use of robust primitives. More precisely, we evaluate the use of color segments features in a matching procedure and compare the dichromatic color lines (Gouiffès and Zavidovique, 2008) with the existing ones, defined in the HSV color space. The motion parameters of the target to track are computed through a voting strategy, where each pair of color segments votes first for one new location, then for two scale changes. Their vote is weighted according to the pairing relevancy and to their location in the bounding box of the tracked object. The comparison is made in terms of robustness to color illumination changes and in terms of quality (robustness of the location of the target during the time). Experiments are carried out on pedestrian and car image sequences. Finally, the dichromatic lines provide a better robustness to appearance changes with fewer primitives. It finally results in a better quality of the tracking.

1 INTRODUCTION

Since the last decades, computer vision and image processing assume a particular importance in robotics. For instance, in the emerging field of intelligent vehicle, the car manufacturers compete to propose assistance multisensor systems based on lasers or vision, in order to ensure a better road safety. In addition to being less and less expensive, vision sensors offer several advantages, the primary of which is to provide a large amount of information on wide regions: depth or motion for example.

Motion or stereovision analysis requires a robust matching of several primitives between two images. In that context, extracting robust features remains a key problem.

Indeed, non-stationary visual appearance usually jeopardizes the matching. Partial occlusions, clutter of the background or a complicated relative motion of the object with respect to the camera (in a moving vehicle for example) are among classical difficulties. Partial occlusions can be dealt with by matching a large amount of sparse features extracted from objects, such as points for example (Baker, 2004). Indeed, it is implausible that the whole features be occluded simultaneously.

Global features based on color invariants (Gevers and Smeulders, 1999), or local features like corners, points, segments, level lines (Caselles et al., 1999) can answer to the problem of photometric changes. Level lines are indeed an interesting alternative to edge-based techniques, since they are closed and less sensitive to external parameters. They provide a compact geometrical representation of images and they are, to some extent, robust to contrast changes. For instance, junctions and segments of level lines have been used successfully in matching processes in the context of stereovision for obstacle detection (Suvonvorn et al., 2007)(Bouchafa and Zavidovique, 2006).

Of course, the choice of the matching strategy has to be led by the nature of the features. That explains partly the large amount of tracking methods, among which correlative and differential methods (Hager and Belhumeur, 1998)(Jurie and Dhome, 2002), kernel-based techniques (Comaniciu and Meer, 2002) and active contours (Paragios and Deriche, 2005) for instance.

This paper compares the robustness of our color segments based on the dichromatic model (Gouiffès and Zavidovique, 2008) with the luminance and HSV color lines defined by (Caselles et al., 2002) and (Coll and Froment, 2000), through an appropriate matching procedure. This method is designed to robustly track rigid and non rigid objects in images sequences. The strategy chosen is based on a weighted voting process in the space of the motion parameters.

The remainder of the paper is structured as follows. Section 2 describes the extraction of the color segments. Then, the matching procedure is explained in Section 3. To finish, the results of section 4 show the efficiency of the proposed color features for matching.

2 SEGMENTS OF COLOR LINES

The concept of level lines is recalled in section 2.1. Then, section 2.2 focuses on the extraction of the segments. Their characterization is finally described in section 2.3.

2.1 Color Lines

Let $I(\mathbf{p})$ be the image intensity at pixel $\mathbf{p}(x,y)$ of coordinates (x,y). It can be decomposed into upper $\mathcal{N}^u$ or lower $\mathcal{N}^l$ level sets:

$$\mathcal{N}^u(\mathcal{E}) = \{\mathbf{p}, I(\mathbf{p}) \geq \mathcal{E}\}, \mathcal{N}^l(\mathcal{E}) = \{\mathbf{p}, I(\mathbf{p}) \leq \mathcal{E}\} \quad (1)$$

where $\mathcal{E}$ denotes the considered level. The topographic map results from the computation of the level sets for each $\mathcal{E}$ in the gray level range. The level lines, noted $\mathcal{L}$, are defined as the edges of $\mathcal{N}$ and form a set of Jordan curves. This concept has been expanded to color in (Coll and Froment, 2000) and (Caselles et al., 1999). The authors use the HSV color space, the components of which are less correlated than RGB's. Also, this representation is claimed to be in adequacy with perception rules of the human visual system. However, they favor the intensity for the definition of the topographic map. Unfortunately, since the hue is ill-defined with unsaturated colors, this kind of a representation may output irrelevant level sets, due to the noise produced by the color conversion at a low saturation.

More recently, the dichromatic lines have been introduced in (Gouiffès and Zavidovique, 2008). They are based on the Shafer model which states that the colors of most Lambertian objects are distributed along several straight lines in the RGB space, joining the origin $(0,0,0)$ to the diffuse color components $\boldsymbol{c}_b(\mathbf{p})$. Therefore, while gray level sets are extracted along the luminance axis of the RGB space, these color sets are designed along each body (or diffuse) reflection vector $\boldsymbol{c}_b$. On each of those vectors, a color can be located by its distance ρ to the origin (the black color), and each vector is located by its zenithal and azimuthal angles (θ, ϕ), in a spherical frame noted TPR in this paper.

These lines provide a good trade-off between compactness and robustness to color illuminant changes. The present evaluation compares the segments extracted in RGB, HSV and TPR through the actual and generic application of tracking.

2.2 Extraction of Color Segments

The segment extraction here is an extension to color of the recursive procedure described in (Bouchafa and Zavidovique, 2006). It exploits the inclusion property of the level sets to extract the segments of level lines. The procedure tracks lines until they split. Along the search, straight subparts, *i.e.* segments, are isolated. The procedure starts at each point $\mathbf{p}$ and first determines which color channel is the most appropriate to track the line. In this paper, the component k of lowest contrast is chosen. Indeed, when a color line exists on this channel, it is likely to exist in both other components, and consequently to lay on a real physical contour of the object. This strategy aims at reducing the extracted noise and the number of segments to match. Once the channel is chosen in $\mathbf{p}$, we determine iteratively which one among $\mathbf{p}$'s 8-connected neighbors is its successor. Each successor becomes the current pixel and the procedure repeats until stopping criteria get true. $\mathbf{q}$ is the successor of $\mathbf{p}$ when the following conditions are respected:

1. At least, one line $\mathcal{L}$ passes between $\mathbf{q}$ and $\mathbf{p}$: $|I(\mathbf{p}) - I(\mathbf{q})| \leq \lambda$.
2. The tracked $\mathcal{L}$ of the chosen path belongs to the same groups of level lines being tracked from the beginning.
3. The interior (vs. exterior) of the corresponding $\mathcal{N}$ is kept on the same side.
4. The tracked level lines remain straight.

For further readings, one can refer to (Bouchafa and Zavidovique, 2006). At that stage, a set of segments $\mathcal{S} = \{s_i\}$ has been extracted from the image.

2.3 Characterization of Color Segments

Fig.1 illustrates the characterization of the segments. A segment s_i is characterized geometrically and colorimetrically: the coordinates of its central point $\mathbf{p}_i = (x_i, y_i)$, its length l_i, its angle α_i, its color. We note ${\mu_i}^L(k)$ and ${\mu_i}^R(k)$, for $k = 1..3$, the mean color on channel k, respectively on the left (L) and on the right hand (R) of the segment s_i.

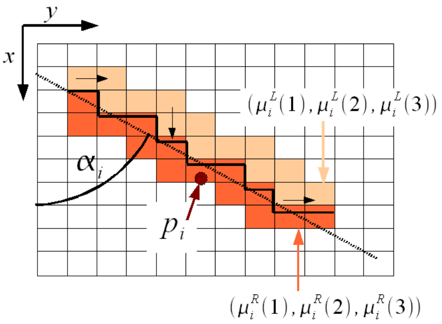

y
x
$(\mu_i^L(1), \mu_i^L(2), \mu_i^L(3))$
α_i
p_i
$(\mu_i^R(1), \mu_i^R(2), \mu_i^R(3))$

Figure 1: Characterization of the color segment.

The following section describes the matching of these color features, based on the definition of a similarity between color segments.

3 MATCHING AND TRACKING

Be I^t and I^{t-1} two subsequent frames at current and previous times t and $t-1$. Object O at time t, denoted as O^t, is described spatially by its bounding box $\mathcal{BB}^t$ of height H^t and width W^t and its centroid $\mathbf{P}^t$, as shown on Fig.2. It can reasonably be selected through a fast motion analysis scheme (Lacassagne et al., 2008) for example.

Knowing the previous object O^{t-1} in I^{t-1}, the tracking consists in computing its new position in I^t by matching the segments exhibited according to section 2.

As in most non-rigid trackers (Comaniciu and Meer, 2002), the object motion is assumed a composition of a translation and two scale changes A_x and A_y along x and y respectively. Since matching is performed between two subsequent frames and supposing a small relative motion object/camera, we further assume a low warping of the object. Therefore, we consider that the new object is located in a search area $\mathcal{V}(O^{t-1})$ which is $\mathcal{BB}^{t-1}$ enlarged by a factor x2. We also consider that the scale changes range in $[1-A, 1+A]$, where A is the maximum possible percentage of scale change.

To secure unambiguous tracking, one needs to consider a large enough number of pairs together. In Fig.2, the object is represented by a set of segments, which are plotted in black. A set of segments $\mathcal{S}^{t-1} = \{s_i\}$ is extracted in O^{t-1} and a set $\mathcal{S}^t = \{s_j\}$ is extracted in $\mathcal{V}(O^{t-1})$. In a first stage, each feature s_i is entitled to match with each feature s_j located in $\mathcal{V}(s_i)$ in I^t. The similarity function explained below evaluates how well features match.

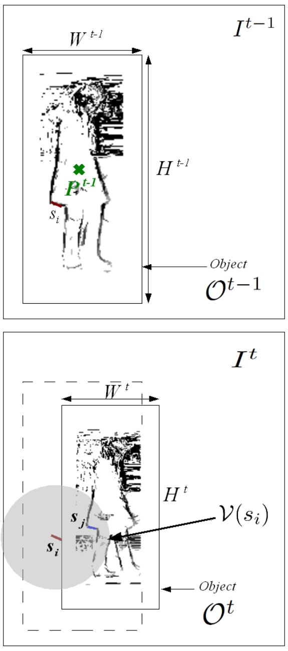


Figure 2: Illustration of the tracking procedure.

3.1 Similarity Function

For all $s_i \in I^{t-1}$ and all $s_j \in \mathcal{V}(s_i) \subset I^t$ (see Fig.2), we define a similarity function based on a color distance $\mathcal{C}_\mu(i,j)$ and the angle difference $\mathcal{C}_\alpha(i,j) \in [0,1]$:

$$\mathcal{C}_\mu(i,j) = \mathcal{C}_0 \sum_{k=1}^{3} |\mu_i^L(k) - \mu_j^L(k)| + |\mu_i^R(k) - \mu_j^R(k)| \quad (2)$$

$$\mathcal{C}_\alpha(i,j) = (|\alpha_i - \alpha_j|_{\text{modulo}\pi})/\pi \quad (3)$$

$\mathcal{C}_0$ is a normalization value which depends on the dynamics of the image, typically $\mathcal{C}_0 = 2^N/6$ for an image coded on N bits. We deduce the following similarity function ($\in [0,1]$):

$$\mathcal{C}(i,j) = 1 - a_\mu \mathcal{C}_\mu(i,j) - a_\alpha \mathcal{C}_\alpha(i,j) \text{ with } a_\mu + a_\alpha = 1 \quad (4)$$

a_μ and a_α balance the similarity criteria. The higher $\mathcal{C}(i,j)$, the more similar s_i and s_j. In order to reduce the number of potential matches, two additional criteria have to be met beforehand:

- s_i and s_j have comparable sizes so they respect the criterion D_l: $D_l = \{1 \text{ when } 1-A \leq l_j/l_i \leq 1+A, \text{ else } 0\}$
- s_i and s_j have comparable directions so they respect the criterion D_α: $D_\alpha = \{1 \text{ when } |\alpha_i - \alpha_j| mod_\pi < T_\alpha, \text{ else } 0\}$, where T_α is a threshold, high enough not to be critical.

3.2 Computation of the New Object Location and Scale

The estimation of both centroid and scales relies on a voting process. Each potential pair of features (s_i, s_j), with $s_j \in \mathcal{V}(s_i)$ votes first to one candidate centroid $\mathbf{P}_j$, each vote being weighted considering the *relevancy* of the pairing features. The notion of relevancy translates in terms of the similarity defined in (2) and in terms of the location of the feature within $\mathcal{BB}^{t-1}$. Indeed, similarly to mean-shift methods (Comaniciu and Meer, 2002), a Gaussian weighting function $K(\mathbf{p}_i)$ is considered for each primitive. In order to cope with partial occlusions and cluttered background, a higher confidence is granted to locations $\mathbf{p}_i$ close to the centroid $\mathbf{P}^t$ compared to peripheral ones.

3.2.1 Estimation of the New Location $\mathbf{P}^t$

Each feature s_i previously extracted on O^{t-1} is assigned a vector $\mathbf{v}_i$ which goes from $\mathbf{p}_i$ to the previous centroid $\mathbf{P}^{t-1}$ such that $\mathbf{v}_i = \mathbf{P}^{t-1} - \mathbf{p}_i$. Since small object motions are conjectured, the scale is assumed to be constant in a first approximation. Therefore, if s_i is correctly matched with s_j of centroid $\mathbf{p}_j$, the candidate centroid $\mathbf{P}_j$ is likely to be located around $\mathbf{p}_j - \mathbf{v}_i$. The uncertainty is lifted only in the rare cases where the object is planar, its motion is strictly fronto-parallel and its scale does not change. In order to model this uncertainty, a 2D Gaussian function $\varepsilon(\mathbf{p}, \sigma_A)$ assigns weights at once to $\mathbf{P}_j$ and to few of its neighbor points. Its standard deviation σ_A expresses the tolerated uncertainty on $\mathbf{P}^t$ due to a scale change $A : \sigma_A = max(AW^t, AH^t)$. Finally, the centroid $\mathbf{P}^t$ is the point $\mathbf{P}_j$ collecting the maximum votes:

$$\mathbf{P}^t = \arg \max_{\mathbf{P}_j \in \mathcal{V}(O^{t-1})} \left(\sum_{s_i} \left(\sum_{s_j \in \mathcal{V}(s_j)} C(ij) K(\mathbf{p}_i) \right) \varepsilon(\mathbf{P}_j, \sigma_A) \right) \quad (5)$$

3.2.2 Estimation of the Scale Changes

At that stage, each pair (s_i, s_j) voted for a centroid candidate $\mathbf{P}_j$. Then, a centroid $\mathbf{P}^t$ was finally estimated as in (5). From there on, we only consider pairs which had voted for a centroid value close enough to the final centroid -*i.e* they respect the scale restriction A on the object size. The scale change values $A_x(i,j)$ and $A_y(i,j)$ are computed for each pair (s_i, s_j) of color features.

$$A_x = \frac{x_i - x^{t-1}}{x_i - x^t} \; A_y = \frac{y_i - y^{t-1}}{y_i - y^t} \quad (6)$$

Similar to the centroid estimation, a weight is assigned to each A_x or A_y value depending on the location in the object and the similarity function. A_x^t is again the scale which collects the maximum votes:

$$A_x^t = \arg \max_{A_x \in [1-A, 1+A]} \left(\sum_{s_i} \left(\sum_{s_j \in \mathcal{V}(s_i)} C(i,j) g(\mathbf{p}_i) \right) \right) \quad (7)$$

Likewise, A_y^t is computed. Once the centroid and the scales have been found, the boundaries of the new current object are well defined and some new color segments are extracted in the subsequent image. The object is lost when the maximum vote is too low.

4 RESULTS

Let us first compare the robustness of the procedures against lighting changes, then on two road sequences.

4.1 Robustness to Lighting Changes

In these first experiments, we use 10 objects of the ALOI image data base[1] viewed under 8 lighting directions and then considering 12 illuminant colors. Fig.3 shows an example of direction variation and Fig.4 illustrates the color changes. The maximum scale change has been fixed to $A = 0.1$ and the color level is $\lambda = 5$. $a_\alpha = a_\mu = 0.5$ in the similarity function (4) and $T_\alpha = \pi/4$.

In the first image, we select manually a window of interest to be tracked and evaluate the matching stationarity during the lighting changes, for the three color representations: RGB, HSV (Coll and Froment, 2000)(Caselles et al., 1999) and TPR(Gouiffès and Zavidovique, 2008). Fig.7 compares the mean variations of the centroids along with lighting changes. Obviously, our color segments provide a better robustness against light variations, since the centroid motion is the smallest for most illumination changes.

In addition, tables 1 and 2 collect the evaluation parameters, namely the number of segments which have been paired, and the *quality Q* of the motion estimation, which is computed as the percentage of pairs which have voted for the estimated motion. Note that the number of segments extracted with the approach TPR is the lowest. That reinforces the conclusions emanated from (Gouiffès and Zavidovique, 2008), *i.e* the compactness of this topographic map. Moreover, TPR provides a better quality of matching (higher values of $Q(\mathbf{P}^t)$, $Q(A_x)$ and $Q(A_y)$) with a lower number of segments, whatever the lighting variations. The good quality of the motion estimation finally explains the good stability of the centroid demonstrated in tables 1 and 2.

[1] more details are available on http://staff.science.uva.nl/ aloi/

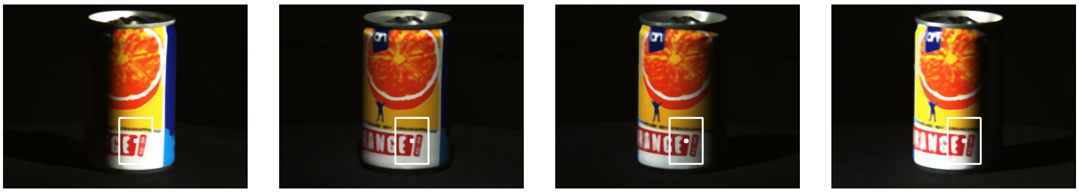


Figure 3: Example of tracking result on the ALOI image data base (object 616) for a change of lighting direction.

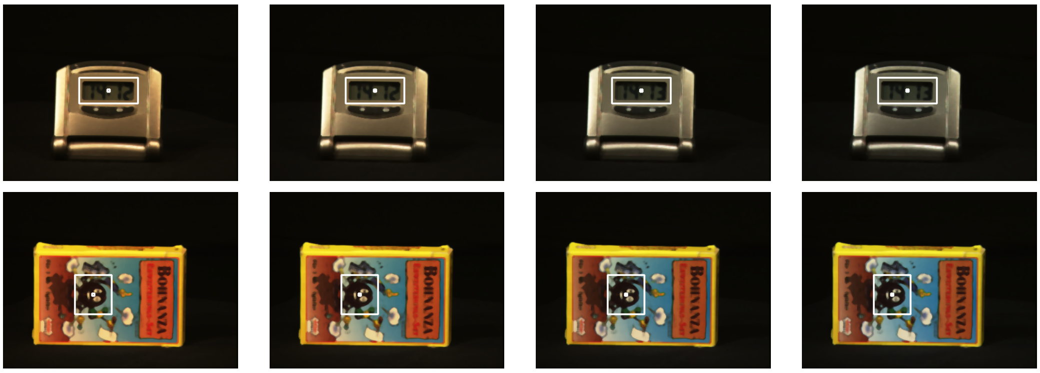


Figure 4: Example of tracking result on the ALOI image data base (objects 104 and 101) for a change of illuminant color.

Table 1: Qualitative results when the lighting direction varies.

Color	Nb	$Q(\mathbf{P}^t)$	$Q(A_x)$	$Q(A_y)$
RGB	670	1,6	60,7	56,1
IST	1054	1,3	56,2	46,2
TPR	518	2,4	65,7	61,6

4.2 Object Tracking

Our tracking procedure is tested here on two different road sequences, the first frames of which are shown on Fig.5 (a) and Fig.6 (b). Only the HSV and TPR segments are compared, since RGB segments did not proved to be efficient in previous experiments. The first image sequence (Fig.5 (a)) `dtneu_nebel`[2] shows an evolving scene acquired under the fog. The blue car is selected manually in the 10$^t h$ frame and has to be tracked until it goes out of the field of view. Note that the appearance of the car changes during the sequence.

The second image sequence (Fig.6 (a))[3] shows a walking pedestrian who turns back and moves away from the camera. The results obtained with HSV segments are shown on images 5(b) and 6(b). The car is lost 10 iterations after its detection, and the tracking of the pedestrian is not accurate. The results of the TPR approach are displayed Fig.5 (c) and Fig.6 (c). Obviously, these latter features provide a far better matching accuracy, since the car and the pedestrian are correctly tracked despite changes in appearance.

Table 2: Qualitative results when the color of illuminant is changed.

Color	Nb.	$Q(\mathbf{P}^t)$	$Q(A_x)$	$Q(A_y)$
RGB	1067	3,8	62,6	63,6
IST	1159	4,0	66,0	61,0
TPR	795	6,3	75,3	68,0

5 CONCLUSIONS

This article introduces some features - segments - bound to dichromatic lines. Their stability for further use was here tested in a tracking procedure, under appearance changes and illuminant color variations. Motion parameters are computed through a common weighted voting process. The dichromatic segments provide the highest tracking quality compared to other segments defined in HSV or RGB spaces. In addition, a lower number of segments is extracted in TPR. Indeed, such "TPR" lines fit the object physical boundaries and are less noise-sensitive, while being robust to lighting changes.

[2] This sequence has been acquired by the KOGS/IAKS Universität Karlsruhe. It is available on http://i21www.ira.uka.de/image_sequences/

[3] LOVe Project: http://love.univ-bpclermont.fr/

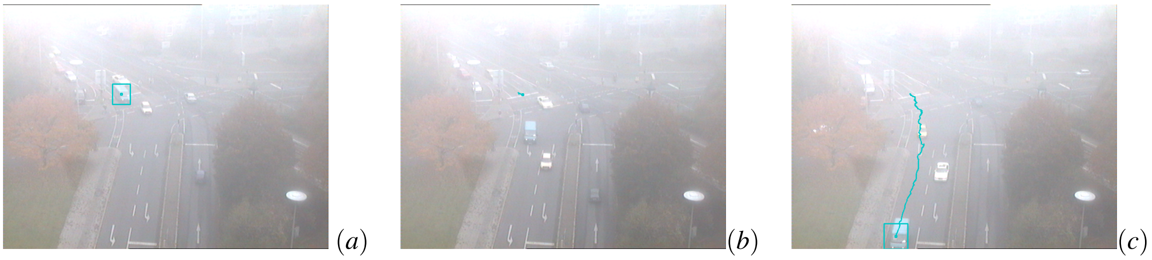


Figure 5: (a): Initial images with their selected object. (b): Results with HSV segments. (c): Results produced with our segments.

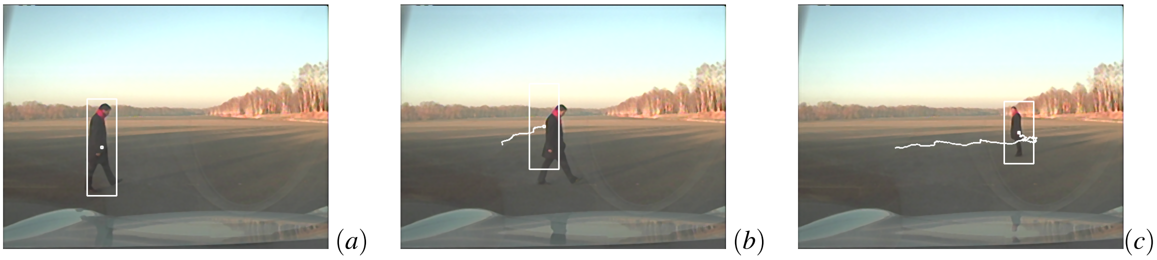


Figure 6: (a): Initial images with their selected object. (b): Results with HSV segments. (c): Results produced with our segments.

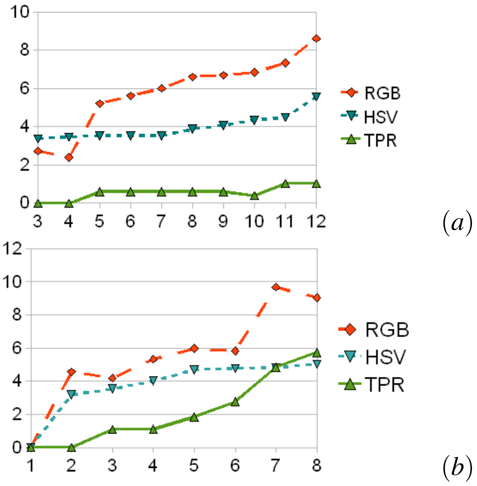


Figure 7: Evolution of the centroid of the object: (a) for different colors of illuminant, (b) for different directions of lighting.

REFERENCES

Baker, S. (2004). Lucas-kanade 20 years on : a unifying framework. *International Journal of Computer Vision*, 56(3):221–255.

Bouchafa, S. and Zavidovique, B. (2006). Efficient cumulative matching for image registration. *IVC*, 24:70–79.

Caselles, V., Coll, B., and Morel, J.-M. (1999). Topographic maps and local contrast change in natural images. *IJCV*, 33(1):5–27.

Caselles, V., Coll, B., and Morel, J.-M. (2002). Geometry and color in natural images. *Journ. of Math. Imag. and Vis.*, 16(2):89–105.

Coll, B. and Froment, J. (2000). Topographic maps of color images. In *15th ICPR*, volume 3, pages 613–616.

Comaniciu, D. and Meer, P. (2002). Mean-shift: a robust approach toward feature space analysis. *III Trans. on PAMI*, 24:603–619.

Gevers, T. and Smeulders, A. W. M. (1999). Colour based object recognition. *Pattern Recognition*, 32(3):453–464.

Gouiffès, M. and Zavidovique, B. (2008). A color topographic map based on the dichromatic model. *Eurasip Journal On IVP*.

Hager, G. D. and Belhumeur, P. N. (1998). Efficient region tracking with parametric models of geometry and illumination. *IEEE Trans. on PAMI*, 20(10):1025–1039.

Jurie, F. and Dhome, M. (2002). Hyperplane approximation for template matching. *IEEE Trans. on PAMI*, 24(7):996–1000.

Lacassagne, L., Manzanera, A., Denoulet, J., and Mérigot, A. (2008). High performance motion detection: Some trends toward new embedded architectures for vision systems. *Journal of Real Time Image Processing, DOI10.1007/s11554-008-0096-7*.

Paragios, N. and Deriche, R. (2005). Geodesic active regions and level set methods for motion estimation and tracking. *CVIU*, 97(3):259–282.

Suvonvorn, N., Coat, F. L., and Zavidovique, B. (2007). Marrying level-line junctions for obstacle detection. In *IEEE ICIP*, pages 305–308.

OBSTACLE DETECTION USING STRUCTURED BACKGROUND

Ghaida Al Zeer, Adnan Abou Nabout and Bernd Tibken
Chair of Automatic Control, Faculty of Electrical, Information and Media Engineering
University of Wuppertal, Rainer-Gruenter-Str. 21, 42119 Wuppertal, Germany
{alzeer, nabout, tibken}@uni-wuppertal.de

Keywords: Image Processing, Mobile Robotics, Path Planning, Obstacle Detection.

Abstract: The paper presents an obstacle detection method for mobile robots using a structured background. This method is based on differences of appearance between obstacles and the background in the camera images. The basic idea is to cover the ground of the workspace with a known structure. If part of this structure is obscured, this can be detected and indicates the existence of an obstacle. The method uses a reference symbol, chosen to exhibit certain features, to construct a reference image of the structured background. To detect possible obstacles we calculate the Fourier descriptors (FD) of all contours included in a given image and compare them with those of the stored reference symbol. This enables us to recognize all reference symbols which are not obscured by obstacles. We then determine the positions and dimensions of all existing obstacles by calculating the occupied symbol areas. The method is implemented as part of a robot-vision system for fully automated stockkeeping. In this paper the results are shown using a MATLAB implementation.

1 INTRODUCTION

Obstacle detection is an important and essential task in a system with mobile robots (Al Zeer, Nabout and Tibken, 2006) and (Al Zeer, Nabout and Tibken, 2007). The method of detection depends on the available information about the obstacles and their environment.

There are various methods of obstacle detection, based on different approaches (Simmons, 1996) and (Sabe, Fukuch, Gutmann, Ohashi, Kawamoto and Yoshigahara, 2004). Two different types of strategies are distinguished. The first strategy is called range-based obstacle detection. This method is based on the measurement of the distance between the obstacle and an implemented sensor, such as an IR-Distance Sensor. The second method is called appearance-based obstacle detection, in which the obstacles, e.g. in a color picture, are separated from the background and classified using their appearance properties (Ulrich and Nourbakhsh, 2000).

Within our work an extended appearance-based method for obstacle detection has been developed, which does not use the appearance of an obstacle, but the appearance of the background. This method, obstacle detection using structure background, was developed for our robot-vision system (Al Zeer, Nabout and Tibken, 2007) and allows a reliable detection and localization of obstacles. Grayscale images are available as part of a robot-vision system. These grayscale images are acquired by a camera system mounted on the ceiling of the workspace area. Therefore the calculation of the obstacle positions will be carried out by image evaluation. Objects in the workspace at the time of image acquisition are considered as static obstacles and they are taken into account in the calculation of collision-free routes. Objects that inter the workspace after the image acquisition are considered as dynamic obstacles (Borenstein and Koren 1990). These obstacles can be detected by the distance sensor which is mounted on the vehicle. Encountering a dynamic obstacle causes the vehicle to stop immediately and to provoke a re-imaging to calculate a new route. Several implementations of the new method will be described in detail and then their realization will be further illustrated.

The paper consists of four sections. In section 2 the principle of the method “obstacle detection using structured background” is explained. Section 3 is concerned with the selection of a suitable reference symbol. In section 4 some results related to the MATLAB implementation are shown and discussed.

2 OBSTACLE DETECTION

The basic idea of this method is to cover of the ground in the workspace with a known structure. If part of this structure is obscured by an obstacle, this can be detected. The structure can be generated by distributing reference symbols in a regular pattern in the workspace of the mobile robot. The distribution of symbols results in a regular and known pattern of contours in every picture of the workspace. Thus, the distribution of the center positions for the reference symbols comprises a matrix structure, similar to the pixel matrix of a bitmapped image.

The use of a symbol pattern as reference feature of the ground leads to an extension of the conceptual specifications of the developed robot-vision system. The implementation of a new strategy of obstacle detection in the robot-vision system was necessary, because it was observed that a guaranteed identification of obstacles through direct detection of objects is not possible because of lighting errors. The condition that the ground has to be covered with a symbol pattern is not an especially unrealistic burden, e.g. in the case of automated stockkeeping. To cover the floor of a warehouse with custom-patterned tiles, painted- or pasted- on symbols is a relatively simple requirement that does not involve a great effort or a high cost. Moreover, in the case of a robot-vision system, this requirement does not violate the principles or change the basic idea of the system.

To verify the results, using the robot-vision system, the extended method was implemented in Matlab and was tested under real conditions. Only common Matlab functions were used in the implementation. The obstacle detection using structure background is based on the following assumptions:

- First, an appropriate symbol pattern has to be applied to the workspace floor. The symbol used to generate the pattern must be suitable for accurate detection. The shapes of the symbol must differ greatly from the shapes of expected objects and also from possible shapes caused by noise and other errors.
- All symbols must be identical, i.e. they all have to be the same shape and size, and they must be arranged in a regular pattern with pre-defined distances to each other.
- The symbols must be large enough for the camera to render their shape and contours with sufficient accuracy. This depends on the resolution of the camera used and the distance between camera and ground.
- The symbols should not have reflective surfaces, so they will not cause reflection noise under bright lighting.

In order for the symbols to comprise a regular pattern on the ground, they have to be applied so that their centers form a regular grid, i.e. the distance between the centers of the symbols must be equal. In the following the individual steps of the newly developed method will be described in detail.

2.1 Creating a Reference Symbol

The reference symbol is used as a basis of comparison for the detection system, so its Fourier descriptors (FDs) are detected and saved (Nabout, 1993).

Figure 1 shows an example for such a symbol. Here "h" is the height, "b" the width, and "c" the center (center of area) of the symbol. The symbol orientation is chosen in suitable way to cover the maximum area of the workspace.

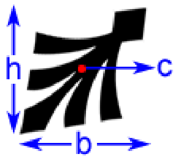


Figure 1: Reference symbol.

The center of gravity of the symbol area can be calculated according to the following general Eq. (1) (Nabout, 1993)

$$X_c = \frac{1}{F}\sum_{i=1}^{n}\int_{x_i}^{x_{i+1}} x\,y\,dx \qquad Y_c = \frac{1}{2F}\sum_{i=1}^{n}\int_{x_i}^{x_{i+1}} y^2\,dx \tag{1}$$

Here, (X_c, Y_c) are the coordinates of the center of the symbol, whose contours were approximated by a polygon with n vertices.

The contour features of the symbol are computed using the contour extraction methods described in (OKE) (Nabout, 1993). Consequently, the original grayscale image is first converted to a binary image. Then the contours of the reference symbol are extracted using the OKE method. Freeman-code is used to describe the contours (cf. Figure 2 and Figure 3).

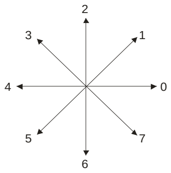


Figure 2: Freeman-Code.

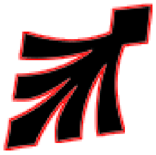


Figure 3: Contours of the reference symbol.

The OKE method consists of the following steps:

1. Segmentation of the image into overlapping 2*2 or 4*4 windows.
2. Evaluation of these windows according to the extraction rules in time with the image sample rate (extraction of basic contour elements).
3. Sorting of the contour elements to generate closed contours.

For the reference symbol used here (Fig. 3), the following Freeman chain was obtained using the MATLAB implementation.

```
000000666566666666566666665444444544444556656656666656666666
66666666666666676666766676667645444444445444222222222222222
222212222212222122212235565556556565665656656665666566666
656666664445444445432222221222122212121212121123555555555556
556555656556565656665444444454444454222212222122222101010100
101101101101111111111113455455454545454454444445443422122 2
221220010000000001000100010010010101010110110113454454445
44445444444454444444444444443222122222222007007000070000
70000000000000000000010000100010101222222122222200001000000
001
```

The starting point of the Freeman-chain is in this example (86, 10). The extracted contour is shown in figure 3.

To approximate the contour of the reference symbol the KKA method (cf. Nabout) is used. The KKA method is carried out as follows:

1. Segmentation of the chain of contour elements into partial sequences that satisfy certain conditions.
2. Detection of noise chains and merging them with neighboring chains.
3. Sorting of partial chains, approximation of curved contour parts.

After these steps, the FDs of the reference symbol are calculated by a Fourier-analysis and are saved in a data structure. Fig. 4 shows the symbol after contour approximation.

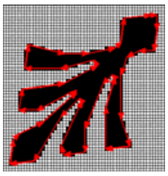


Figure 4: The approximated contour of the symbol of Fig.1.

2.2 Creation of a Reference Image

To create a reference image, the workspace floor is covered with symbols forming a grid structure as described above. Then an image of the empty workspace without obstacles is acquired (Figure 5).

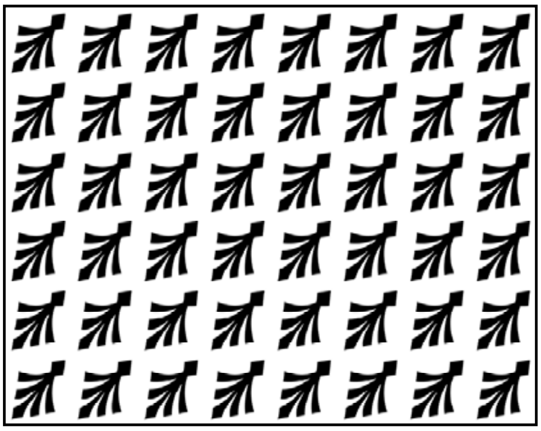


Figure 5: Workspace floor with reference symbols.

The reference image should be created under real conditions. The acquisition parameters are set so that all the symbols in the picture are completely visible. Since a camera-acquired image always contains other contours in addition to the symbols (noise, errors, etc.), as a first step *all* the contours in the picture are extracted and approximated using the OKE and KKA-methods and saved in the form of polygonal approximations. To remove the contours created by noise we adopt a length filter using an upper and lower threshold for the contour length. Subsequently, the FDs and centers of area gravity for the symbol polygons are computed. In our example, the reference image contains 48 symbols. By comparing the FDs of the extracted polygon with the FDs of the reference symbol, all existing symbols in the reference image can be detected. Their centers then determine the grid structure (Figure 6).

[47, 45	47, 138	:
146, 45	:	744, 322
247, 45	:	47, 413
346, 45	744, 138	:
445, 45	47, 229	:
544, 45	:	744, 413
644, 45	:	47, 506
744, 45	744, 229	:
:	47, 322	:
:	:	744, 506]

Figure 6: Coordinates of center of area gravity in reference image with 8*6 symbols.

In our case, a grid with 8*6 symbols is detected. Since the identification of the symbols by comparison the shape of the extracted contours with the shape of the reference symbol is the key to a reliable detection of all symbols, it is necessary to use a reference symbol whose shape (represented through its FDs) differs greatly from those of other occurring objects or possible noise. To choose the reference symbol as a circle, for example, would result in confusion with all circular contours that arise due to lighting errors. Such a reference symbol is not suited for the present application.

For this reason, the following strategy to select a suitable reference symbol was used in this work.

3 SELECTION OF A SUITABLE REFERENCE SYMBOL

Within this work, a total of 30 symbols were investigated in order to select the most suitable symbol for generating the grid structure. Figure 7 shows 10 of those symbols.

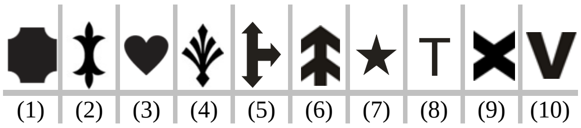


Figure 7: Some representatives of the symbols considered for the generation of a symbol pattern.

The goal here is to find a symbol which gives the smallest confusion risk with other contours that may occur. For this reason, the FDs of all considered symbols were calculated and compared using the minimum distance method (Nabout, 1993). Figure 8 shows for example the first 20 FDs of the above given symbols in a graphical comparison.

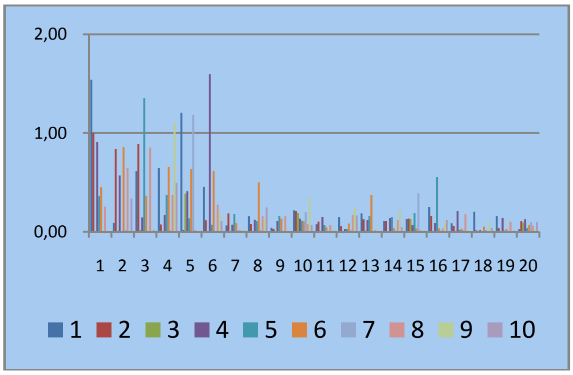


Figure 8: The first 20 FDs for the symbols of Fig.7.

The Euclidean distances d_i between every two symbols were calculated, according to Eq. (2).

$$d_{ij} = \sqrt{\sum_{k=1}^{n} \left(\mathrm{A_{ik}} - \mathrm{A_{jk}}\right)^2} \; ; \qquad i,j = 1,\ldots,30 \qquad (2)$$

d_{ij}: Euclidean distance between object i and j
n: Number of Fourier descriptors
A_{ik}, A_{jk}: Amplitude spectrum of object contour i, j

The most suitable symbol, with the smallest confusion risk, is that one which fulfills the following three conditions:

1. The chosen symbol must have big distances to all other symbols. This is fulfilled for that symbol i with the maximum mean value $\bar{d}_i$ of the Euclidean distances $\mathrm{d_{ij}}, j\colon 1,\ldots,30$.
2. The symbol must have approximately equal distances to all other symbols. This is fulfilled for that symbol i with the minimum variance $\mathrm{v_i}$ of the Euclidean distances.
3. The chosen symbol must have partially regular contour rather than irregular to contrast to noise contours.

The means $\bar{d}_i$ and the variance $\mathrm{v_i}$ are determined for every symbol i according to Eq. (3).

$$\bar{d}_i = \frac{1}{\mathrm{M}} \sum_{\mathrm{j=1}}^{\mathrm{M}} \mathrm{d_{ij}}$$
$$\mathrm{v_i} = \frac{1}{\mathrm{M}-1} \sum_{\mathrm{j=1}}^{\mathrm{M}} (\mathrm{d_{ij}} - \bar{\mathrm{d}}_{\mathrm{j}})^2 \qquad (3)$$

Here, M is the number of considered symbols.

Finally, the symbol i with the maximum mean value and the minimum variance is selected according to Eq. (4).

$$\mathrm{i_{opt}} = \operatorname{argmin}\left(\frac{\mathrm{v_i}}{\bar{\mathrm{d}}_{\mathrm{i}}}\right) \qquad (4)$$

To fulfill the third condition we calculated all values $\frac{\mathrm{v_i}}{\bar{\mathrm{d}}_{\mathrm{i}}}$ and sorted them according to their values from small to large in a list (L). Then we chose the first symbol in the list (symbol 2) which shows partially regular contour shape. The following two Figures show the mean value and the variance for the objects of Fig.7.

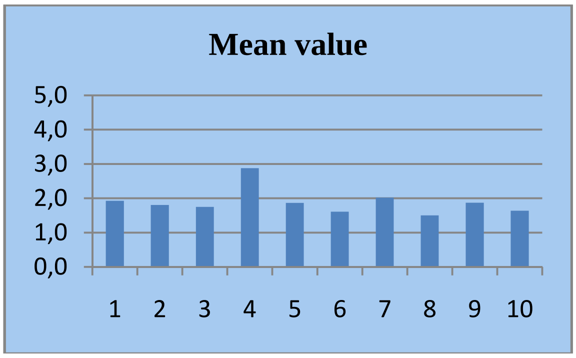


Figure 9: Mean for each symbol (10 symbols).

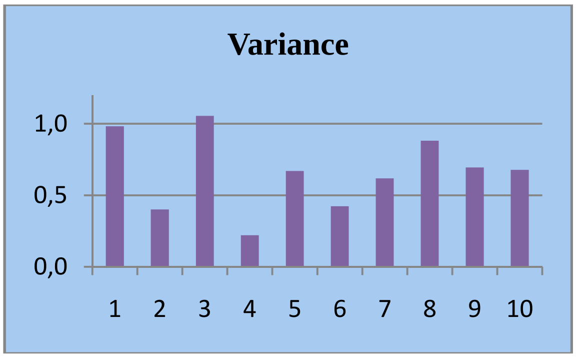


Figure 10: Standard variance for each symbol (10 symbols).

4 DETECTION OF OBSTACLES

To detect the obstacles, first a picture of the workspace is taken. This picture contains the reference symbols on the structured floor, as well as a certain number of obstacles. Figure 11 shows an image with simulated obstacles.

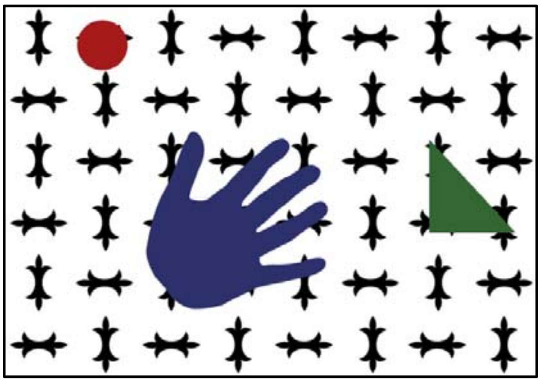


Figure 11: Reference image with obstacles.

To evaluate this image, the following steps were executed:

1. Contour extraction using OKE method
2. Contour approximation using KKA method
3. Computation of FDs
4. Determination of the centers of all detected reference symbols in the image.

After performing these steps and comparing the FDs to those of the reference symbols by the minimum distance method we received for the given example 35 detected reference symbols. The detection occurs, if the computed minimum distance value between a considered object and the reference symbol is smaller than a specified threshold. In our case, the threshold was determined *a priori* and set to the value 0.5.

If the minimum distance is greater than this threshold, the considered object is designated as an obstacle. To locate the positions and sizes of existing obstacles the centers of found symbols in the current image are compared with the centers of the symbols in the reference image. The centers of the detected symbols and the centers of the symbols in the reference image must be identical, within a certain tolerance value. For our example, we obtain the following results (Fig. 12).

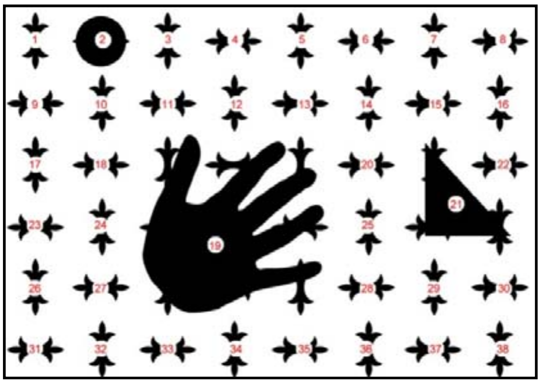


Figure 12: Binary image with numbered objects.

Here, there are 13 symbols obscured by obstacles, whose centers are also known.

The calculation of the positions and sizes of the obstacles to determine the region occupancy is illustrated in the following section.

4.1 Mapping the Obstacles into the Workspace

In an X-Y coordinate system, whose origin is situated in the upper left corner of the workspace (cf. Fig.11), each symbol S_i is represented by the position of its center (x_i, y_i). For two adjacent symbols on a horizontal line, the distance between the centers of these two symbols is *dx*. Similarly, for two adjacent symbols, which are arranged vertically, it is *dy*.

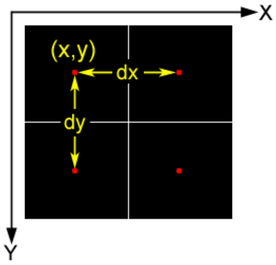


Figure 13: Center of a macro pixel with horizontal and vertical distance to its neighbours.

In order to detect the positions and size of the existing obstacles in the workspace, a binary image with the same resolution as the original grayscale image is generated. The image is then divided into regular $M \times N$ blocks, where M is the number of columns and N is the number of rows in the symbol grid. All pixels within one block are assigned as "0" (black), if the symbol represented by this block is obscured by an obstacle, otherwise, the pixels are assigned the value "1" (white). Figure 14 shows the binary image generated for the example of Figure11.

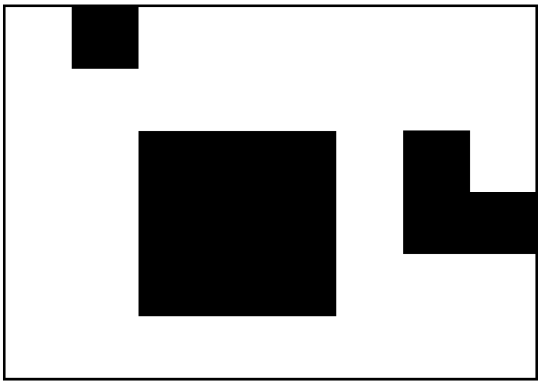


Figure 14: Binary image showing occupancy of obstacles.

As the picture shows, there are three black areas, which represent the positions of the obstacles. In order to determine the size of the obstacles, the contours of the areas are described as polygons. For these polygons the Minimum Area Rectangles of the obstacle regions (MAR) are then calculated (Al Zeer, Nabout and Tibken, 2008). These regions represent the workspace area occupied by the present obstacles.

5 CONCLUSIONS

With the obstacle detection method presented here, it is possible to detect existing obstacles in the workspace of mobile robots. This method uses grayscale images and a specially chosen reference symbol. The results show that symbols with irregular contour shape are not qualified to be used as reference symbol, since the recognition process leads to confusion with possible lighting errors and other noise. The paper shows how to choose a good reference symbol using a mathematical formula.

The method was developed for the detection of obstacles in a robot vision system which is part of a fully automated stockkeeping application. In this context the results of the obstacle detection described in this paper were also used for path planning using auxiliary corners.

REFERENCES

Al Zeer, G., Nabout, A., Tibken, B., Path Planning for Mobile Robots by Means of Approximate Routes, 2007 IEEE International Conference on Control and Automation, Guangzhou, CHINA, May 30 to June 1, 2007, pp. 2468-2473.

Al Zeer G., Nabout, A., Tibken, B., Hindernisvermeidung für Mobile Roboter mittels Ausweichecken, 52nd Internationales Wissenschaftliches Kolloquium, Technische Universität Ilmenau, 10.–13. Sep. 2007, pp. 437-442.

Al Zeer, G., Nabout, A., Tibken, B., Extended Method for Path Planning Using Auxiliary Corners, the 3rd IEEE International Conference on Information & Communication Technologies: from Theory to Application, ICTTA'08, 7-11 April 2008, Damascus – Syria.

Borenstein, J., Koren, Y., Real time obstacle avoidance for fast mobile robots in cluttered environments. IEEE Conf. Robotics and Automation, pp. 572-577, 1990.

Nabout A., Modulares Konzept und Methodik zur wissensbasierten Erkennung komplexer Objekte in CAQ-Anwendungen, 1993.

Simmons. R., The Curvature-Velocity Method for Local Obstacle Avoidance. In Proc. IEEE Conf. Robotics and Automation, 1996.

Sabe, K., Fukuch, M., Gutmann, J., Ohashi, T., Kawamoto K., Yoshigahara, T., Obstacle Avoidance and Path Planning for Humanoid Robots using Stereo Vision.Proceedings of the International Conf. on Robotics and Automation, ICRA'04, New Orleans, April 2004.

Ulrich, I., Nourbakhsh, I., Appearance-Based Obstacle Detection with Monocular Color Vision, Proceedings of the AAAI National Conference on Artificial Intelligence, Austin, TX, July/August 2000.

A MHT-BASED ALGORITHM FOR PERFORMANCE ESTIMATION IN DT-MRI BAYESIAN TRACKING METHODS

L. M. San José Revuelta
ETSI Telecommunication, University of Valladolid, Campus Miguel Delibes, Valladolid, Spain
lsanjose@tel.uva.es

Keywords: DT-MRI. Fuzzy system, Medical image processing, Reliability estimation, Bayesian modelling.

Abstract: This paper deals with the development of a recursive fuzzy inference system that can be applied to estimate the error probability of several tracking algorithms used in medical image processing systems. Specifically, we are interested in the fiber bundles estimation process (*fiber tracking*) in diffusion tensor (DT) fields acquired via magnetic resonance imaging (MRI). As tracking algorithm we have considered a variation of the Bayesian tracking scheme proposed by Friman and Westin. This paper studies the analogies between this tracking approach and a typical Multiple Hypotheses Tracing (MHT) system, for which fuzzy systems are closely related. This comparison leads to the development of a SAM (Standard Additive Model) fuzzy system that on-line estimates the certainty of the estimated fiber tracts. Its low computational load as well as its efficiency in very isotropic volumes are its main advantages.

1 INTRODUCTION

The technique of Diffusion Tensor Magnetic Resonance Imaging (DT-MRI) measures the diffusion of hydrogen atoms within water molecules in 3D space. Since in cerebral white matter most random motion of water molecules are restricted by axonal membranes and myelin sheets, diffusion anisotropy allows depiction of directional anisotropy within neural fiber structures (Ehricke, 2006).

There exist many important applications for white matter tractography: brain surgery, white matter visualization using fiber traces and inference of connectivity between different parts of the brain, to name a few.

The great majority of DTI visualization techniques focuses on the integration of sample points along fiber trajectories and their three-dimensional representation (Mori, 2002). These streamline-based approaches are called *fiber tracking* and they usually make use only of the principal eigenvector of the diffusion ellipsoid as an estimate of the predominant direction of water diffusion in a voxel (Ehricke, 2006). Nevertheless, and due to some deficiencies in these tracking algorithms and several shortcomings inherent in datasets (noise, partial voluming), they may depict fiber tracts which do not exist in reality or miss to visualize important branching structures. In order to avoid misinterpretations, the viewer of the visualizations must be provided with some information on the uncertainty of a depicted fiber and of its presence in a certain location. This task can be efficiently tackled if a Bayesian approach is used.

In this paper, we have considered a Neural network-based simplified implementation of a well-known Bayesian tracking algorithm (Friman, 2005). Specifically, this algorithm has been implemented with a simplification method based on those used in (San José, 2005) in the context of a Bayesian detector for digital multiuser communications.

Our goal is to establish a parallelism between a standard Bayesian tracking scheme and another procedure, the Multiple Hypotheses Tracking (MHT) strategy (Alberola, 1999; Reid, 1979), which is directly related to fuzzy logic and, to our knowledge, has not been directly applied to fiber estimation. The thus developed fuzzy system will calculate more reliable estimates of the depicted tracts certainty.

2 BAYESIAN TRACKING ALGORITHM

Bayesian modelling has already been applied to fiber tracking. However, its main drawback is the large computational load involved. In this paper we propose to use the Bayesian algorithm of Friman and

Westin (Friman, 2005) with the *Stochastic Drawing Sampling Selection* (SDSS) scheme developed in (San José, 2005) for complexity limitation. This Bayesian algorithm is next described.

The goal of the Bayesian modelling is to find a pdf of the local fiber orientation[1] $p(\hat{\mathbf{v}}_k|\hat{\mathbf{v}}_{k-1}, D)$, where vectors $\hat{\mathbf{v}}_k$ and $\hat{\mathbf{v}}_{k-1}$ contain the path samples up to time k or $k-1$, respectively, and D denotes the measured diffusion data. If a model that relates the diffusion measurements D with the underlying tissue properties and architecture is assumed, then it must contain at least one fiber direction $\hat{\mathbf{v}}_k$ and a set of nuisance parameters denoted by θ. Thus, applying the Bayes theorem,

$$p(\hat{\mathbf{v}}_k, \theta|\hat{\mathbf{v}}_{k-1}, D) = \frac{p(D|\hat{\mathbf{v}}_k, \theta)p(\hat{\mathbf{v}}_k|\hat{\mathbf{v}}_{k-1})p(\theta)}{p(D)} \qquad (1)$$

where we have assumed that the prior distribution can be factorized $p(\hat{\mathbf{v}}_k, \theta|\hat{\mathbf{v}}_{k-1}) = p(\hat{\mathbf{v}}_k|\hat{\mathbf{v}}_{k-1})p(\theta)$. The main problems found are (Friman, 2005): (i) the calculation of $p(\hat{\mathbf{v}}_k|\hat{\mathbf{v}}_{k-1}, D)$ needs to marginalize Eq. (1) over θ, and (ii) the normalizing factor

$$p(D) = \int_{\hat{\mathbf{v}}_k, \theta} p(D|\hat{\mathbf{v}}_k, \theta)p(\hat{\mathbf{v}}_k|\hat{\mathbf{v}}_{k-1})p(\theta) \qquad (2)$$

is difficult to evaluate due to the high-dimensional integral and the intractable integrand. Eq. (1) has to be calculated in every step in the sequential sampling of the fiber paths and, unless an approximation for the integral in Eq. (2) is found, the cost is prohibitive.

Some attempts have been made to approach this problem. In (Friman, 2005), a solution based on drawing samples from a pdf defined on the unit sphere is proposed. This is accomplished by evaluating the pdf at a sufficiently large number of points evenly spaced over the unit sphere, effectively approximating the continuous pdf with a discrete pdf, from which it is straightforward to draw the random samples. However, the continuous pdf must be densely enough sampled, specifically, Friman proposes to use 2,562 predefined points thus involving an important computational burden. At this point, we propose to use a sampling strategy where those points (*hypotheses*, in the Bayesian terminology of (San José, 2005)) with the largest probabilities have more chances to be selected. However, notice that some randomness is introduced in the selection procedure. This way, those directions with the highest probability to prolong the current fiber path will *probably* be selected. Specifically, we have implemented the Stochastic Drawing Sampling Selection (SDSS) algorithm in order to reduce the number of sampled points in the above-mentioned unit sphere.

3 COMPARISON BETWEEN BAYESIAN AND MHT

A fuzzy version of Reid's classical Multiple Hypotheses Tracking (MHT) algorithm (Reid, 1979) was proposed in (Alberola, 1999). This system is based on the likelihood discrimination and it was applied to the tracking of natural language text-based messages. It shows the possibility of handling information about any time-varying phenomenon, as long as the phenomenon can be described by means of a few keywords, and the phenomenon itself is statistically causal in the sense that the distribution of future states is statistically dependent on the past observed states.

It is not difficult to see the following parallelism that leads to the possibility of a tract probability estimation based on text-messages (fuzzy-messages): (i) the natural-language messages in (Alberola, 1999) and the noisy DT-MR image constitute, in both cases, the source of *noisy* or *ambiguous* information, (ii) the *tracks* used in the MHT algorithm, which are defined as *sequences of associated symbols*, can be clearly associated to the possible sequences of points in the 3D space, in the tracking context, (iii) the MHT system associates multiple messages generated along time by using a specific stochastic model for the applications' dynamics. In our case, this model can be the information provided by the measured anisotropy, (iv) the term *target* denotes some condition that generates observable phenomena. In our context, these targets are the sequences of points that define a tract.

As a consequence, the MHT system can be viewed as a Bayesian approach for multiple targets tracking. Theoretically, this algorithm conserves *all* the hypotheses that explain the observation until certain time, together with an estimation of the probability of each hypothesis. At the end, the hypothesis with the highest likelihood is taken as the solution. On the other hand, the Bayesian tracking algorithm maintains a finite set of hypotheses (section 2) with their associated probabilities, and a tract is coloured and visualized based on these data.

4 PROPOSED FUZZY SYSTEM

In this section we propose a recursive SAM (Standard Additive Model) fuzzy subsystem that allows to monitor the performance of a DT-MRI tracking system. The SAM model allows to work with linguistic descriptions and ambiguities. This kind of description allows to fuzzy-quantify the errors in the tractography problem. On the other hand, the uncertainty in the prediction of the future positions found in the MHT of

[1] Using the notation found in (Friman, 2005).

(Alberola, 1999), resembles the creation of new fiber tracts based on the previous ones.

The system here proposed consists in three connected fuzzy inference engines (FIEs) –see Fig. 1. It is necessary to develop an algorithm where the inputs to the MHT system have some correlation in time.

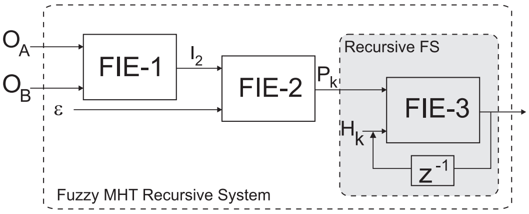


Figure 1: Recursive SAM fuzzy system for estimation of the error probability of the estimated error tracts.

The inputs O_A and O_B to the FIE-1 are two different tracts (hypotheses) estimated by the algorithm sharing in common the first and the last points (in practice, both tracts must start and finish in near voxels). These tracts are prolonged on one side with a new sample every time a new point is considered (at every iteration of the tracking algorithm), while the last point of the tracts is lost. This way, compared tracts have always the same length.

In order to evaluate the similarity between two tract hypotheses O_A and O_B, it is necessary to quantify their proximity using a 3D distance. As a consequence, a *similarity coefficient* that depends on the distance between these two considered tracts can be assigned.

In order to implement a fuzzy system, we must establish a relation between this *crisp* value (defined in $[0-K]$) and the fuzzy sets where a linguistic variable is defined, i.e.,

$$\begin{bmatrix} K \\ K-1 \\ \vdots \\ 1 \\ 0 \end{bmatrix} \longrightarrow \begin{bmatrix} \textit{Very Unlikely} \\ \textit{Unlikely} \\ \vdots \\ \textit{Likely} \\ \textit{Very Likely} \end{bmatrix} \tag{3}$$

This allows to obtain the possible fuzzy values of I_2 (output of the first FIE and input to the second).

Next, we relate the *prediction error* ε used as input in the FIE-2 with the anisotropy observed in the last (currently processed) point of the tract. This way, if a large anisotropy is obtained, the tract would be rather smooth in the proximity of the current voxel and ε will take a small value for those hypotheses (future points to expand the current tract) that involve a small change in the fiber direction. On the other hand, when the anisotropy is small (isotropic area), parameter ε would be the same for every direction (hypotheses). The value of ε must, also, be fuzzified.

This way, FIE-1 estimates the likelihood of two close tracts while FIE-2 weights this estimate with respect to the prediction error (that is inversely proportional to the anisotropy) and obtains a second likelihood. This value is used to update the *global likelihood* (or *global reliability*), which is a measure of the tracking estimation error probability. This third process is performed by FIE-3. Thus, this third block updates, with a feedback system, the previous system knowledge every time a new point is processed.

5 NUMERICAL RESULTS

5.1 Synthetic Images

First, four different synthetic DT-MRI data in a 50 × 50 × 50 grid have been generated (three of them –*cross*, *earth* and *log*– can be seen in Fig. 3 of (San-José, 2006) while the fourth one, named *star*, –the most complex one– is new. To make the simulated field more realistic, Rician noise was added in the diffusion weighted images which were calculated from the Stejskal-Tanner equation using the gradient sequence in (Westin, 2002) and a *b*-value of 1000.

The desired noisy synthetic diffusion tensor data was obtained using an analytic solution to the Stejskal-Tanner equation. The eigenvectors in the isotropic areas were $\lambda_1 = \lambda_2 = \lambda_3$, while in the remaining voxels of the image $\lambda_1 = 7$, $\lambda_2 = 2$, $\lambda_3 = 1$. In our study, the SNR varies from 13 to 29 dB.

The "star" image consists of six orthogonal sine half-waves, each of them with arbitrary radius. Notice that this scenario constitutes the most complicated situation since the diffusion field experiments variations with the three coordinate axes and there exists a crossing region.

The reliability of four approaches for estimating the tracts certainty is first studied. These methods are: (i) the tracking algorithm described in (San-José, 2006), ("ALG"), (ii) the Bayesian algorithm described in section 2 ("BAY"), (iii) ALG with the fuzzy engine for probability of error estimation ("ALG+Fuzzy"), and (iv) BAY with the fuzzy procedure ("BAY+Fuzzy"). Figure 2 shows the mean probability of wrong estimation (average value in 25 executions) and Table 1 presents the mean variance of these estimators, for five different signal qualities ranging from 13 to 29 dB.

Analyzing the results it can be seen that: (i) the probability of error increases as the SNR of the original image improves; more complex images have

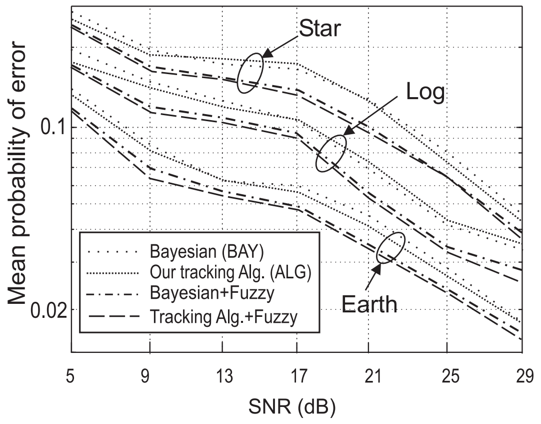


Figure 2: Mean probability of error for the tracking methods studied, with and without the fuzzy procedure for estimation of the probability of error. Three synthetic images were used: earth, log and star.

Table 1: Variance of the tracking error estimation method. Each cell represents the values for: BAY/**ALG**/Fuzzy-estimation.

	SNR (dB)		
	13	21	29
Earth	4.2/**3.7**/1.6	2.3/**1.9**/0.7	1.4/**0.9**/0.2
Log	4.5/**4.1**/2.0	2.8/**2.3**/0.9	1.5/**0.9**/0.3
Star	6.4/**5.1**/2.8	3.2/**2.7**/1.5	2.2/**1.7**/0.7

larger tracking error estimates, (ii) in general, the accuracy of the ALG method is slightly better than the BAY approach, and (iii) the tracking error of both methods (BAY and ALG) improves notably when the fuzzy engine is used for estimation. These figures are closer to the real probability of error when a human expert manually evaluates the tracts obtained.

Table 1 shows how the fuzzy procedure greatly decreases the variance of the estimator, leading to more robust and accurate estimations, specially for low quality images. The values shown in each cell represent the variance of the different estimation approaches: BAY, ALG and the fuzzy-based estimation using the strategy proposed in section 4. The fuzzy method obtained very similar results when combined to both BAY and ALG. Thus, only one value is included in each cell.

It can be observed that the fuzzy approach gets estimates with much smaller variances. This estimation procedure is scarcely influenced by both the SNR of the image and image complexity (in terms of anisotropy). This implies that the estimation convergence will not depend on the presence of branching or crossing areas of the MR figure –as it would be the case in real DT-MR images.

5.2 Real Images

Finally, we have applied the proposed tracking algorithm to a real DT-MR image. Specifically, we have selected the *corpus callosum* of the brain.

The variance of the same four estimation methods has been evaluated. Results are shown in Table 3. Once again the improvement on the estimates reliability can be observed for both BAY and ALG.

Table 2: Variance of different probability of error estimation methods.

BAY:	8.4	BAY+fuzzy:	3.8
ALG:	7.4	ALG+fuzzy:	3.2

If noisy voxels are present along the paths of interest it is worth noting that the MHT-based fuzzy method is less sensitive to these variations. The reason is that the MHT performs a kind of *smoothing* or data *filtering*, which decreases the disturbing effects of the occasionally high noisy samples (this is addressed using the FIE-3 in Fig. 1).

ACKNOWLEDGEMENTS

The authors acknowledge the Spanish CICYT for research grant TEC2007-67073/TCM.

REFERENCES

Alberola, C., C. G. V. (1999). Tracking with text-based messages. In *IEEE Intell. Systems (14)*.

Ehricke, H. H., K. U. G. W. (2006). Visualizing mr-dt fields by dynamic fiber tracking and uncertainty mapping. In *Computers & Graphics (30)*.

Friman, O., W. C.-F. (2005). Uncertainty in white matter fiber tractography. In *Proc. MICCAI 2005, LNCS 3749*.

Mori, S., v. Z.-P. C. M. (2002). Fiber tracking: principles and strategies – a technical review. In *Nuclear Magnetic Resonance in Biomedicine (15)*.

Reid, D. B. (1979). An algorithm for tracking nmultiple targets. In *IEEE Tr. Automat. Contr., vol. AC-24*.

San José, L. M. (2005). Hypotheses control-based strategies for the simplification of bayesian multiuser detectors. In *Proc. IEEE XIV Workshop on Machine Learning for Signal Processing*.

San-José, L. M., M. M. A. C. (2006). A new method for fiber tractography in diffusion tensor magnetic resonance images. In *Proc. ICSIP 2006, vol. I*.

Westin, C.-F. e. a. (2002). Processing and visualization for dt-mri. In *Med. Image Analysis (6)*.

MULTI SCALE MOVING CONTROL METHOD
FOR AUTONOMOUS OMNI-DIRECTIONAL MOBILE ROBOT

Masaki Takahashi and Takafumi Suzuki
Department of System Design Engineering, Keio University, 3-14-1, Hiyoshi, Kohoku-ku, Yokohama 223-8522, Japan

Keywords: Collision avoidance, Service robots, Hierarchical control, Omni-directional mobile robot.

Abstract: This paper proposes a hierarchical moving control method for autonomous omni-directional mobile robot to achieve both safe and effective movement in a dynamic environment with moving objects such as humans. In the method, the movement of the robot can be realized based on prediction of the movement of obstacles by taking account of time scale differences. In this paper, the design method of the proposed method based on the virtual potential approach is proposed. In the method, modules that generate the potential field are structured hierarchically based on the prediction time to each problem. To verify the effectiveness of the proposed method, the numerical simulations and the experiments using a real robot are carried out. From the results, it is confirmed that the robot with the proposed method can realize safe and efficient movement in dynamic environment.

1 INTRODUCTION

Recently, various essential technologies of an autonomous mobile robot such as a self localization scheme, an environmental map formation and path planning, learning algorithm and communication are developed in the area of robot. In addition, a variety of service robots which offers service with the actual environment with other moving objects, including people are proposed and developed(B. Graf, 2004)-(R. Bischoff). A variety of tasks are required for such a service robot, but here we will focus on problems related to moving, which is the most fundamental and important of tasks. In the environment include humans, safe and efficient movement should be required. As for the movement of the autonomous mobile robot, the problem which has the various time scales, such as arrival to destination, the collision avoidance for the obstacle and the emergency collision avoidance for the sudden obstacle, occurs simultaneously. Therefore, the robot should keep coping with the problem according to circumstance.

This paper proposes a hierarchical moving control method for autonomous omni-directional mobile robot to achieve safe and effective movement in a dynamic environment with moving objects such as humans. The hierarchical control method considers a variety of prediction time to each action, such as destination path planning, obstacle avoidance within the recognizable range, and emergency avoidance to avoid spontaneous events. In the method, several modules for each action are composed in parallel. The vertical axis is prediction time scale in the control system. In the lowest module, the robot can move to goal safely and efficiently by planning from the environment information which is obtained in advance. On the other hand, in the higher module, the robot moves more safely by using the estimated information of obstacles based on shorter prediction time to avoid them. By integrating the output of each module, it is possible to realize the safe and efficient movement according to the situation.

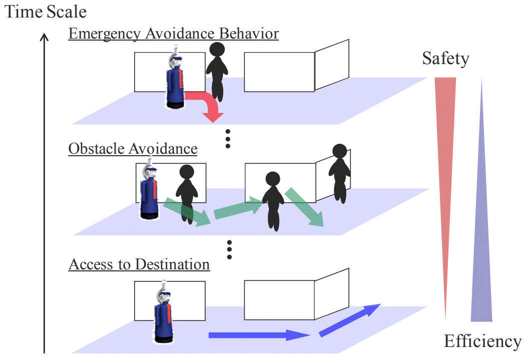


Figure 1: Problem Establishment for Action of Service Robot.

In this paper, as one example of design method of the proposed control method, the design method which is based on the virtual potential method is presented (Khatib, 1986) (Y. Koren, 1991). Firstly, the module which generates the potential field based on each prediction time is formed hierarchically. Secondly, the virtual force which is derived from the respective potential fields is synthesized. Thirdly, the velocity command is decided on the basis of the resultant force. To verify the effectiveness of the proposed method, the numerical simulations which suppose the environment where the obstacle exists were carried out. Moreover, the experiments using the real apparatus of the autonomous omni-directional robot were carried out.

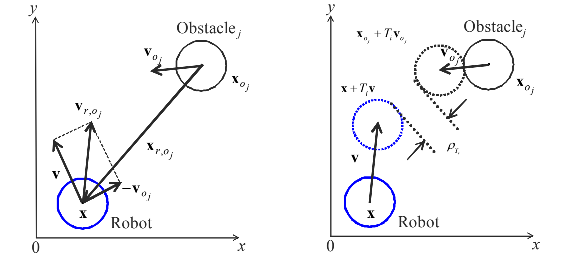


Figure 2: World coordinate system and predicted shortest distance.

2 HIERARCHICAL ACTION CONTROL METHOD

2.1 Nomenclature

Symbol	Quantity
T_i	prediction time
ρ_d	distance between the robot and the destination
ρ_{T_i}	predicted shortest distance between the robot and the obstacle
ρ_0	minimum of repulsive potential
$\mathbf{x}$	position vector of robot
$\mathbf{x}_d$	position vector of the destination
$\mathbf{x}_{o_i}$	position vector of object $_j$
$\mathbf{x}_{r,o_i}$	position vector of the obstacle O_j relative to the robot
$\mathbf{v}$	velocity vector of robot
$\mathbf{v}_{o_i}$	velocity vector of object $_j$
$\mathbf{v}_{r,o_i}$	velocity vector of the obstacle O_j relative to the robot
$U_j^{T_i}$	virtual potential about object $_j$ on each Time scale
$\mathbf{F}_j^{T_i}$	virtual force vector from $U_j^{T_i}$
i	index of each Time scale
j	index of object
x	x-axis
y	y-axis

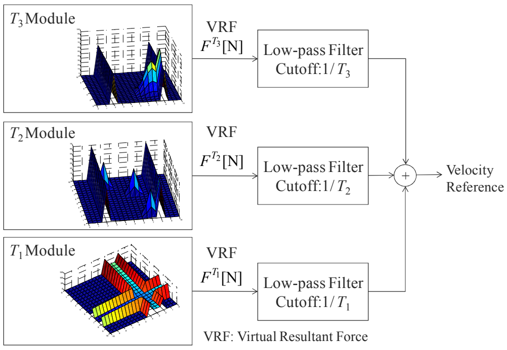


Figure 3: Output of each module and integration.

2.2 Design Approach

The module is a potential function with the prediction time as a parameter, and generates a potential field for each problem and virtual force on the robot is calculated. In this study, the proposed potential function was designed based on the repulsive potential reported by Khatib (Khatib, 1986).

$$U = U_{\mathbf{x}_d} + U_o \tag{1}$$

$$U_{\mathbf{x}_d} = k_a \rho_d \tag{2}$$

$$U_o = \sum U_j^{T_i} \tag{3}$$

$$U_j^{T_i} = \begin{cases} \eta T_i \left(\dfrac{1}{\rho_{T_i}} - \dfrac{1}{\left\| T_i \mathbf{v}_{r,o_j} \right\| + \rho_0} \right)^{\frac{1}{T_i}} & , \rho_{T_i} \le \left\| T_i \mathbf{v}_{r,o_j} \right\| + \rho_0 \\ 0 & , \rho_{T_i} > \left\| T_i \mathbf{v}_{r,o_j} \right\| + \rho_0 \end{cases} \tag{4}$$

where $\mathbf{x}_d$ is the destination position and $U_{\mathbf{x}_d}$ is an attractive potential field. In the proposed method, a repulsive potential function in consideration with prediction time T_i is used.

A force for the position $\mathbf{x}$ of the robot is derived from the following equation.

$$\mathbf{F}(\mathbf{x}) = -\frac{\partial U}{\partial \mathbf{x}} \tag{5}$$

where $\dfrac{\partial U}{\partial \mathbf{x}}$ denotes the partial derivation vector of the total virtual potential U. From Eqs. (2) and (5), the attractive force allowing the position $\mathbf{x}$ of the robot to reach the goal position $\mathbf{x}_d$ is as follows:

$$\mathbf{F}_{\mathbf{x}_d} = -k_a \frac{\partial \rho_d}{\partial \mathbf{x}} \tag{6}$$

From Eqs. (4) and (5), the repulsive force to the obstacle O_j are as follows:

$$\mathbf{F}_j^{T_i} = \begin{cases} \eta \left(\dfrac{1}{\rho_{T_i}} - \dfrac{1}{\left\| T_i \mathbf{v}_{r,o_j} \right\| + \rho_0} \right)^{\frac{1}{T_i}-1} \dfrac{1}{\rho_{T_i}^2} \dfrac{\partial \rho_{T_i}}{\partial \mathbf{x}} & ,\rho_{T_i} \le \left\| T_i \mathbf{v}_{r,o_j} \right\| + \rho_0 \\ 0 & ,\rho_{T_i} > \left\| T_i \mathbf{v}_{r,o_j} \right\| + \rho_0 \end{cases} \tag{7}$$

The command vector **F** of the robot is derived from the following equation.

$$\mathbf{F} = \mathbf{F}_{\mathbf{x}_d} + \mathbf{F}_o \tag{8}$$

When combining the virtual force derived from a potential field which is generated at each module, we consider the robot as a point mass. The velocity command with the same magnitude and direction is determined by combining the forces to the robot. In addition, the potential approach has a vibration problem caused by the magnitude of velocity and roughness of the control period. Thus, in the method, a low pass filter on each element of the virtual force output in each module is used to suppress such vibration as shown in Fig.3. It was confirmed that safe and effective motion is possible even in a situation where movement to the destination, avoiding moving obstacles, and emergency avoidance all coexist. In the simulations, each low pass filter uses the reciprocal of each prediction time as a cut-off frequency.

3 EXPERIMENTAL RESULTS

3.1 Experimental Environment

To verify the effectiveness of the proposed method in the actual situation, the experiments using the real robot were carried out. The robot size is L 0.55 × W 0.75 × H 1.25 m and the weight of the robot is about 60 kg. In order to recognize environment, the stereo camera and the stemma camera, the laser range finder and the ultrasonic sensor are loaded, but, in this research the robot recognizes environment making use of only the laser range finder. The velocity limit of the robot is 0.5m/s and the acceleration limit is 1.0m/s^2.

Figure 4 shows the experimental environment to verify the effectiveness of the proposed method to a static single obstacle. The initial position of the robot is (0 m, 0 m). The obstacle size is L 0.20 W 0.33 H 0.50 and its initial position is (-0.5 m, 3.0 m). Figure 7 shows the experimental environment. In this case, the moving obstacle bursts through the blind corner at the speed of 0.5 m/s when the robot comes close to the corner.

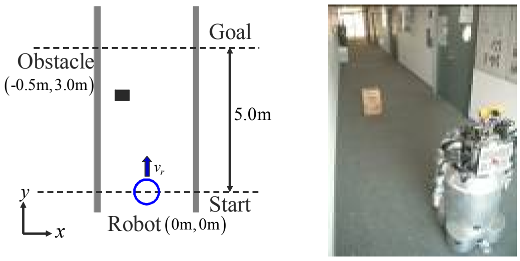
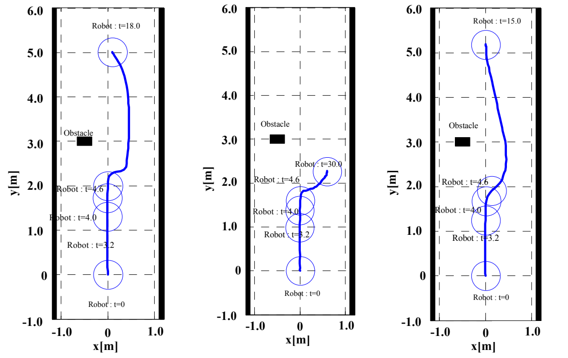


Figure 4: Experimental Environment.

3.2 Experimental Results

Figure 5 (a), (b) and (c) show the trajectory of the robot by using the Khatib ($\rho_0 = 0.8$, $\eta = 0.064$), the Khatib ($\rho_0 = 1.5$, $\eta = 0.064$) and the proposed method respectively.

From the result in Fig.5(a), it was confirmed that the robot comes close to the obstacle because the repulsive potential fields for the obstacle is small. Fig.5 (b) shows that the robot does not approach to the obstacle because the influence of the obstacle is large. In addition it receives the influence of repulsive force from the wall and thereby this can lead to the stable positioning of the robot before reaching its goal. On the other hand, it was confirmed in Fig.5(c) that the robot with the proposed method can reach its goal earlier than other methods without colliding with the obstacle.



(a) $\rho_0 = 0.8, \eta = 0.064$ (b) $\rho_0 = 1.5, \eta = 0.064$ (c) The proposed method.

Figure 5: Experimental Result.

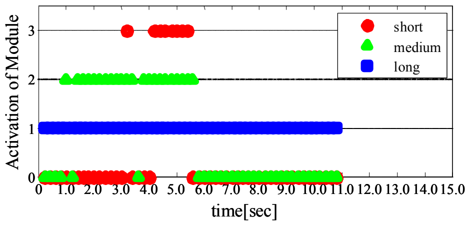


Figure 6: Time History of the Activation of Module.

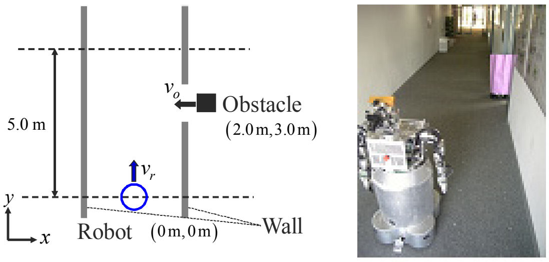
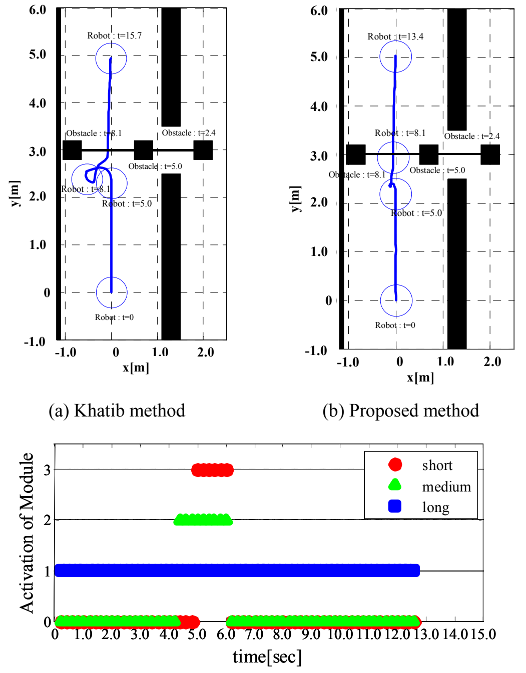


Figure 7: Experimental Environment.



(a) Khatib method

(b) Proposed method

(c) Time History of the Activation of Module in the proposed method

Figure 8: Experimental Result.

Figure 8(a) and (b) show the trajectories of the robot and the moving obstacle by using the Khatib and the proposed method respectively. Figure 8(c) shows the time history of the activation of module in the proposed method. The robot can reach the goal without colliding with the obstacle. However, the robot moves in the direction of movement of the obstacle because the predicted information of the obstacle is not used. Thereby, the arrival time to the goal is longer than our method.

From the results in Fig.8 (b), it was confirmed that the robot recognizes the moving obstacle and then stops on the moment and starts the movement to the goal after the obstacle passes over. As shown in Fig.8(c), the robot can move without colliding against the moving obstacle by acting on the emergency avoidance module simultaneously with the collision avoidance module around 5.0sec which it approaches to the robot.

4 CONCLUSIONS

This study proposed the hierarchical action control method for an autonomous omni-directional mobile robot to realize the safe and effective movement. In the method, the module with different prediction time processes in parallel, and the command velocity to the robot is decided by integrating them. As for each module, the selection condition is different according to relative position and velocity about the robot and the obstacle.

From the results of the numerical simulations and the experiments, it was confirmed that the robot can reach the goal efficiently without colliding with both the static and the moving obstacles by using the estimated information of them.

REFERENCES

Graf, B., Hans, M. and Schraft, R. D.: “Mobile Robot Assis-tant,” *IEEE Robotics and Automation*, vol.11, no.2, pp.67-77, 2004.

DeSouza, G. N. and Kak, A. C.: “Vision for Mobile Robot Navigation: A Survey,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.24, no.2, pp.237-267, 2002.

Thrun, S. et al: “MINERVA: A Second-Generation Mu-seum Tour-Guide Robot,” *Proc. IEEE Int. Conf. on Robotics and Automation*, pp.1999-2005, 1999.

Bischoff, R. and Graefe, V.: “HERMES - A Versatile Personal Robotic Assistant,” *Proc. IEEE* vol.92, no.11, pp.1759-1779, 2004.

Khatib, O.: “Real-time Obstacle Avoidance for Manipulators and Mobile Robots,” *Int. J. of Robotics Research*, vol.5, no.1, pp.90-98, 1986.

Koren, Y. and Borenstein, J.: “Potential Field Methods and Their Inherent Limitations for Mobile Robot Navigation,” *Proc. IEEE Int. Conf. on Robotics and Automation*, pp.1398-1404, 1991.

LARGE-SCALE DEXTEROUS HAPTIC INTERACTION WITH VIRTUAL MOCK-UPS
Methodology and Human Performance

Damien Chamaret, Paul Richard
LISA Laboratory, University of Angers, 62 avenue Notre Dame Du Lac, Angers, France
{damien.chamaret, paul.richard}@univ-angers.fr

Sehat Ullah
IBISC Laboratory, University of Evry, 40 rue du Pelvoux, Evry, France
sehat.ullah@ibisc.univ-evry.fr

Keywords: Virtual Environment, Virtual Mock-up, Large-scale, Haptic Interaction, Human Performance.

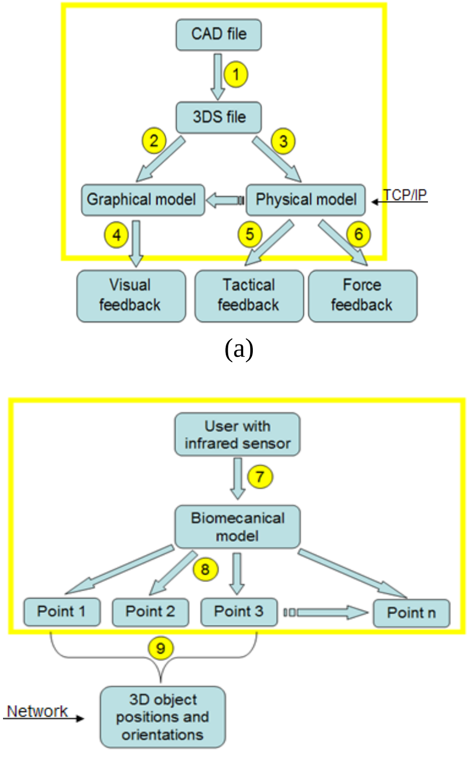
Abstract: We present a methodology for both the efficient integration and dexterous manipulation of CAD models in a physical-based virtual reality simulation. The user interacts with a virtual car mock-up using a string-based haptic interface that provides force sensation in a large workspace. A prop is used to provide grasp feedback. A mocap system is used to track user's hand and head movements. In addition a 5DT data-glove is used to measure finger flexion. Twelve volunteer participants were instructed to remove a lamp of the virtual mock-up under different conditions. Results revealed that haptic feedback was better than additional visual feedback in terms of task completion time and collision frequency.

1 INTRODUCTION

Nowadays, Car manufacturers use Computer Aided Design (CAD) to reduce costs, time-to-market and to increase the overall quality of products. In this context, physical mock-ups are replaced by virtual mock-ups for accessibility testing, assembly simulations, operation training and so on. In such simulations, sensory feedback must be provided in an intuitive and comprehensible way. Therefore, it is of great importance to investigate the factors related to information presentation modalities that affect human performance. This paper presents a methodology for both the efficient integration and dexterous manipulation of CAD models in a physical-based virtual reality simulation. The user interacts with a virtual car mock-up by using a string-based haptic interface that provides force sensation in a large workspace. An experimental study was carried out to validate the proposed methodology and evaluate the effect of sensory feedback on operator's performance. Twelve participants were instructed to remove a car's lamp from a virtual mock-up. Three experimental conditions were tested concerning sensory feedback associated with collisions with the virtual mock-up: (1) no-feedback (only graphics), (2) additional visual feedback (colour) and (3) haptic feedback. Section 2 describes the CAD-to-VR methodology. Section 3 presents the virtual environment (VE) that allows large-scale haptic interaction with the virtual car mock-up. In section 4, the experimental study and the results are presented. The paper ends by a conclusion and gives some tracks for future work.

2 CAD-TO-VR METHODOLOGY

The proposed CAD-to-VR methodology involves different steps (illustrated in Figure 1a), such as model simplification (1), model integration (2-3). The graphical model is used for visual display of the virtual mock-up (4), while the physical one is used for both tactile and kinaesthetic feedback (5-6). Our methodology for model simplification allows to decrease the number of polygons of the CAD models while keeping the same level of visual quality. Model integration allows to obtain both graphical and physical models of CAD data. Physical models are built using PhysX engine (www.nvidia.com).



(a)

(b)

Figure 1: Schematic of the CAD-to-VR methodology (a) and human interaction using the mocap system (b).

3 VIRTUAL ENVIRONMENT

Our methodology also allows the integration of both the graphical and physical models of users (Figure 1b). A biomechanical model is used for the animation of operator's hand and arm (7). In order to get accurate position and orientation tracking of the user, an infrared camera-based motion capture system is used. Six reflected markers are placed on the operator's body (8): three markers on the data-glove to assess hand position and orientation (9), one marker on a cap worn by the operator for head tracking, and two markers on the operator's arm.

The large-scale VE provides force feedback using the SPIDAR system (Space Interface Device for Artificial Reality) (Ishii and Sato, 1994). Stereoscopic images are displayed on a rear-projected large screen (2m x 2.5m) and are viewed using polarized glasses. The SPIDAR system uses a SH4 controller from the Cyverse (Japan). In order to provide force feedback to both hands, a total of 8 motors are placed on the corners of a cubic frame surrounding the user. In order to provide haptic grasping feedback to the operator, a prop (see Figure 2) was used (Chamaret et al., 2008).

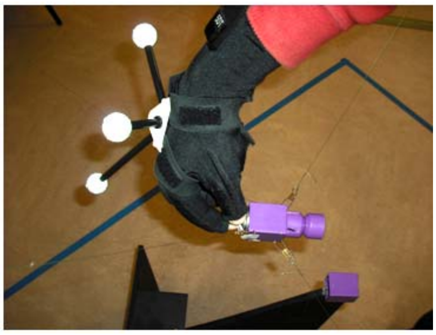


Figure 2: The prop (real car lamp inside a plastic cap) used for grasping feedback.

Poor grasp of the prop or a bad calibration due to unexpected movements may cause problems of feedback coherency between grasping (prop) and simulated forces (SPIDAR). To avoid these problems, three zones were defined: (a) a free zone where the user can freely moves his/her hand (hand position/orientation and fingers flexion) using a 5DT data glove, (b) an assistance zone (d_1 = 10 cm from the virtual lamp) where the user is no more able to change fingers flexion, and (c) a grasping zone (d_2 = 5 mm from the virtual lamp) where the grasping gesture is realized (Figure 3):

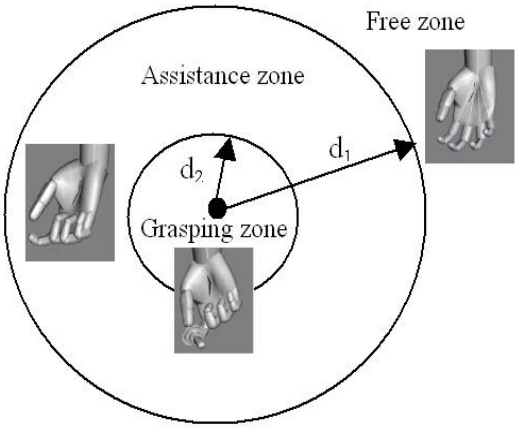


Figure 3: Illustration of the three zones used for the grasping simulation and assistance.

4 EXPERIMENTAL STUDY

The aim of this experiment is twofold: (1) validate the proposed CAD-to-VR methodology including operator's biomechanical model integration, and (2) investigate the effect of haptic and visual feedback on operator performance in a task involving extraction and replacement of a car's lamp in a virtual car mock-up.

4.1 Experimental Set-up

The experimental set-up is illustrated in Figure 4. The user interaction with the virtual mock-up using the camera-based mocap system. Global force feedback is provided using the SPIDAR system. Local (grasp) feedback is achieved using the prop.

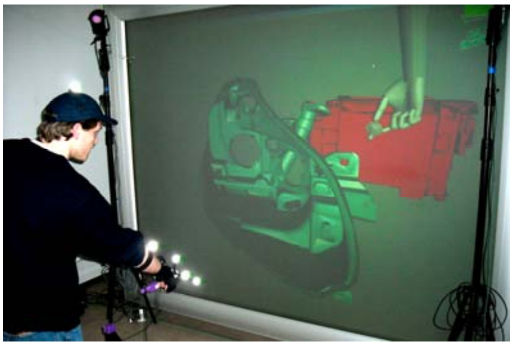


Figure 4: Illustration of a user performing the task.

4.2 Procedure

Twelve volunteer students participated in the experiment. They were naives in the use of virtual reality technique. Each participant had to perform the maintenance task in the following conditions:

- C1: no additional feedback (only graphics);
- C2: additional visual feedback (colour);
- C3: haptic feedback (from SPIDAR).

The task has to be repeated three times for each condition. Conditions were presented in different order to avoid any training transfer. Participants were in front of a large rear-projected screen at a distance of approximately 1.5 meter. They worn a 5DT data glove equipped with three reflective balls (Figure 2). In order to get acquainted with the system each participant performed a pre-trial of the task in C1 condition.

4.3 Data Collection

The following data were collected during the experiment for each single trial:

- task completion time
- number of collisions

4.4 Results

Results were analysed through ANOVA. We examine the effect sensory feedback on (a) task completion time and (b) collision time. Then, we look into the learning process associated with the different sensory feedback.

4.4.1 Task Completion Time

Results, illustrated in Figure 5, revealed that sensory feedback has a significant effect on task completion time: (F(2,11)=14.08; p<0.005). A statistical difference between conditions C1, C2 and C3 was observed. In C1 condition the average completion time was 30.34 sec (STD = 3.1). Average completion time was 26.45 sec (STD = 1.8) for C2 (additional visual feedback) and 22.24 sec (STD = 3.4) for C3 (haptic feedback). Thus visual and haptic feedbacks allow increasing performance, as compared with the open-loop case (no additional feedback), by 12.8 % and 16 % respectively. Haptic feedback increase performance by 15.6 % as compared to additional visual feedback. However, participants' performance was more disparate.

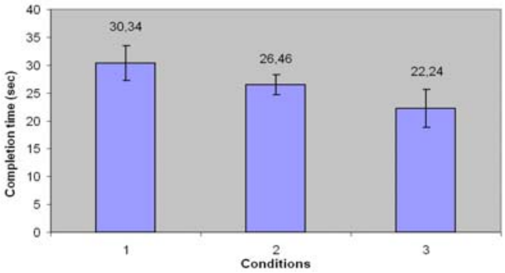


Figure 5: Completion time versus conditions.

4.4.2 Number of Collisions

Results, illustrated in Figure 6, revealed that sensory feedback has a significant effect on the number of collisions: (F(2,11)=63.70; p < 0.005). As previously, a statistical difference between C1, C2 and C3 conditions was observed. In C1 the average number of collisions was 6.64 (STD = 0.58). The average number of collisions was 4.83 sec (STD = 0.15) for C2 and 4.05 (STD = 0.8) for C3. Thus visual and haptic feedbacks led to a significant reduction of the number of collisions as compared to the open-loop case, by 27.3 % and 39.0 % respectively. Haptic feedback increase performance by 16.2 % as compared with additional visual feedback. As for task completion time, participants' performance was more disparate in condition C3.

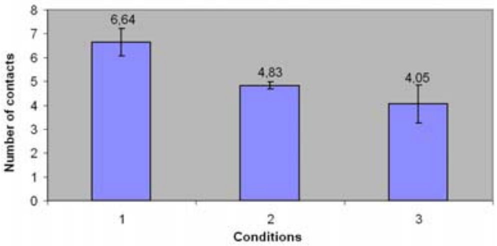


Figure 6: Number of collisions versus conditions.

4.4.3 Learning Process

The learning process is defined here by the improvement of participant performance associated with task repetitions. We analysed the learning process associated with both task completion time and number of collisions. Although each participant repeated the task three times only, a learning process was observed for all conditions (Figure 7, 8, and 9).

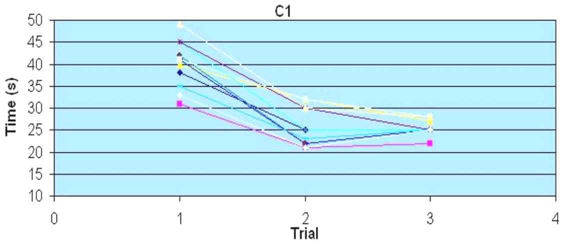


Figure 7: Learning process associated with condition 1.

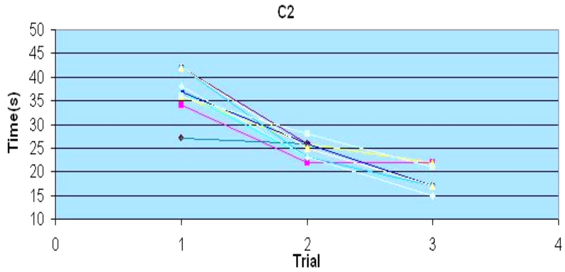


Figure 8: Learning process associated with condition 2.

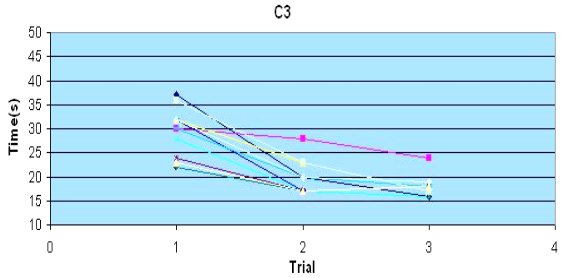


Figure 9: Learning process associated with condition 3.

Average task completion time was 40.2 sec at the first trial and 25.4 sec at the last trial for condition C1, 36.7 sec at the first trial and 18.1 sec at the last trial for condition C2, and 29.2 sec at the first trial and 17.9 sec at the last trial for condition C3. This results in a performance improvement of about 37%, 50%, and 48% for conditions C1, C2 and C3 respectively.

Concerning the number of collisions, we observed a poor learning process for each condition. This result is not very surprising for C1 condition since no feedback was displayed for collisions. In the C3 condition, participants were good at the first trial. This shows that the haptic interface is user-friendly and efficient. The poor learning process associated with C2 condition may be explained by the lack of spatial information as is it the case with force feedback (sensation of force direction during collision).

5 CONCLUSIONS

This paper presented a methodology for both the integration and dexterous manipulation of CAD models with biomechanical model in a physical-based virtual reality simulation. The user interacts with a virtual car mock-up using a string-based haptic interface that provides force sensation in a large workspace. Twelve participants were instructed to remove a lamp of the virtual mock-up under different conditions. Results revealed that haptic feedback was better than additional visual feedback to reduce both task completion time and collision frequency. In the near future we plan to integrate haptic guides in order to assist the users to reach and grasp the cars lamps in a more efficient way.

REFERENCES

Ishii M., Sato M., 1994. A 3d Spacial interface Device using Tensed Strings. *Presence, 3(1).*

Chamaret D. Richard P. Ferrier J.L., 2008. From CAD Model to Human-Scale Multimodal Interation with Virtual Mockup: An Automotive Application. *5th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2008) Madeira, Portugal.*

INTEGRATED PATH PLANNING AND TRACKING FOR AUTONOMOUS CAR-LIKE VEHICLES MANEUVERING

Fernando Gómez-Bravo, Diego A. López
Departamento de Ingeniería Electrónica, S. I. y Automática, University of Huelva, 21819 La Rábida, Huelva, Spain
fernando.gomez@diesia.uhu.es, diego.lopez@diesia.uhu.es

Francisco Real, Luis Merino, José M. Matamoros
Robotics, Vision and Control Group,University of Seville, Camino de los Descubrimientos, 41092 Sevilla, Spain
freal, merino, jmatamoros@cartuja.us.es

Keywords: Autonomous Car-like vehicle, Maneuvering, Motion Planning, Path-tracking.

Abstract: This paper proposes a new method for control car-like vehicles maneuvering. For this purpose, traditional planning and tracking algorithms has been modified in order to follow complex maneuvers in a frame of a distributed control architecture. Thus, planning and tracking algorithm run in different platforms exchanging data in order to control the maneuvers performance.

1 INTRODUCTION

Autonomous navigation of car-like vehicles is a well known topic that has attracted the attention of many researchers (Paromtchik et al., 1998), (Ollero et al., 1999), (Gomez-Bravo et al., 2001), (Wada et al., 2003), (Cuesta et al., 2004), (Daily and Bevly, 2004). Navigation in cluttered scenarios usually involves maneuvering that require stopping the vehicle and changing the sign of the vehicle velocity to avoid collisions. The non-holonomic constraints of the car-like vehicles play an important role. Path tracking (Ollero et al., 1994), (Ollero et al., 1996) and path planning techniques (Latombe, 1991), (Laumont et al., 1994), (LaValle, 2006) have been largely approached in mobile robot literature. However, generating and tracking car-like maneuvers are not frequently reported (Cheng et al., 2001), (Wada et al., 2003), (Cuesta et al., 2004). This paper presents a new integrated architecture particularly designed for planning and tracking maneuvers. Furthermore, this strategy can be also applied when complex maneuvers are not required.

Real-time path planning usually requires significant computational resources that could overload the limited on-board processing capability. Then, a suitable alternative is to implement the planner on external dedicated servers that could provide this capability to one or several networked autonomous vehicles. These servers could be also networked with sensors in the environment providing information for navigation (Gomez-Bravo et al., 2007).

In this paper a distributed implementation integrating both maneuvers planning and tracking techniques for autonomous car-like maneuvering is presented. The novelty of this method is the adaptation of the planning and tracking method so that both can work in different computer, establishing a communication process between the planner and the tracker system in order to control effectively the maneuver performance.

On the one hand, the planning method applied in this paper is based on the Rapidly Exploring Random Trees algorithm (RRT) (LaValle, 1998), (Bruce and Veloso, 2002) (LaValle, 2006). This technique can be easily extended to non-holonomic vehicles providing simple solutions for car-like maneuvers generation (LaValle and Kuffner, 1999), (Gomez-Bravo et al., 2008). In the present approach, besides obtaining path for maneuvering in complex scenarios, the planner is capable of dividing the originally computed maneuver into different sections in order to facilitate the tracking task and allowing to modify easily the original path if changes in the initial map are detected. On the other hand, the path tracking technique is a modified version of the well known pure pursuit geometric approach (Ollero, 2001). The original tracking strategy has been modified so that this method can be applied for maneuvers tracking. Moreover, and different from previous approaches, in the architecture presented in this paper the planner determines the maneuver performance by establishing some of the tracker parameters.

The paper is organized as follows: in the next section the basis of the car-like vehicles maneuvering is introduced and the application of the RRT to maneuvers generation is also presented. Section 3 is devoted to describe the adaptation of the path tracking algorithm for maneuvering. In section 4 the proposed implementation is presented. Finally, in section 5, experimental results, obtained with a real autonomous car-like vehicle when maneuvering in an outdoor scenario, are presented. The paper ends with the conclusions and the references.

2 CAR-LIKE VEHICLES MANEUVERING

2.1 Car-Like Vehicles

Car-Like vehicles are non-holonomic systems characterized by kinematics constraints resulting in nonintegrable differential equations that should be taken into account for motion planning and obstacle avoidance.

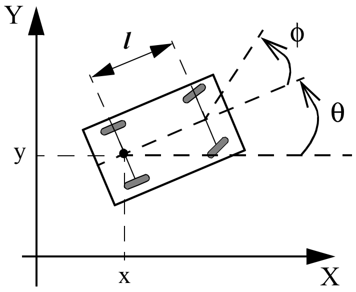


Figure 1: Car-like vehicle.

Usually, the kinematics model of car-like vehicles is expressed as:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v(t) \\ v(t)\frac{\tan\phi(t)}{l} \end{bmatrix} \quad (1)$$

where (x,y) is the position of the rear reference point, and θ is the vehicle's heading, both in a global reference frame, $v(t)$ is the linear velocity, $\phi(t)$ is the steering angle, that defines the curvature of the path, and l is the distance between the rear and front wheels (see Fig. 1).

Typically, the values of $\phi(t)$ are constrained by the value ϕ_{max}, accomplishing the relation

$$R_{min} = \frac{l}{\tan\phi_{max}} \quad (2)$$

where R_{min} is the minimum curvature radius.

Each wheel of the vehicle presents a kinematics constraint which prevents the robot from moving to the orthogonal direction of the wheels longitudinal axe. Then, the kinematics of these vehicles is usually characterized by the following differential nonholonomic equation

$$\dot{x} \cdot \sin\theta - \dot{y} \cdot \cos\theta = 0. \quad (3)$$

In the next section, the basis of the method for coping with this constraint, when planning trajectories, are presented.

2.2 Continuous Navigation vs. Maneuvering

There are different approaches to car-like vehicles motion planning (Laumont et al., 1994), (Cheng et al., 2001), (Gomez-Bravo et al., 2008). Path planning techniques provides trajectories based on different searching techniques (Latombe, 1991), (LaValle, 2006). Particularly, the randomized methods have attracted considerable attention (Barraquand and Latombe, 1991), (Amato and Wu, 1996), (Cheng et al., 2001), (Bruce and Veloso, 2002). Due to the non-holonomic nature of these vehicles, many of these methods require further processing in order to obtain a path accomplishing the kinematics constraints. Thus, derivable and continuous curvature trajectories are frequently provided by these techniques. However, when cluttered scenarios are involved, complex maneuvers may be needed, and path generation should also include *inversor configurations*, (Latombe, 1991), i.e configurations where the sign of the vehicle velocity changes. In these cases, discontinuous curvature trajectories can also be considered as admissible paths, although the kinematics constraints still have to be accomplished.

2.3 Planning Maneuvers

General strategies for maneuvers generation have not been frequently addressed, (Latombe, 1991), (Laumont et al., 1994), (LaValle, 2006). Recent approaches have focused attention in car-like parking maneuvers (Paromtchik et al., 1998), (Gomez-Bravo et al., 2001), (Cuesta et al., 2004).

The method presented in this paper is based on a planner previously presented in (Gomez-Bravo et al., 2007) and (Gomez-Bravo et al., 2008). This method takes the advantage of a randomized generation process, the original Bidirectional RRT algorithm (LaValle, 1998), (Bruce and Veloso, 2002) adapted to non-holonomic vehicles. Firstly, RRT is used without considering any kinematics constraints providing a data structure (two trees with free collision configurations) connecting the initial with the goal configuration (see Fig. 2).

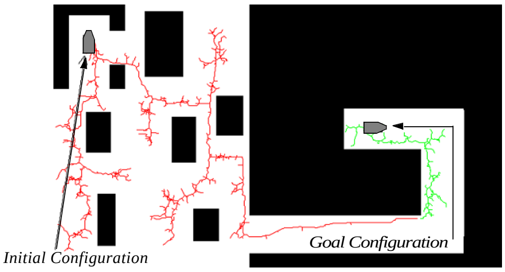


Figure 2: RRT connecting trees.

In the next step, the tree provided by the algorithm is turned into a sequence of feasible admissible paths by means of a set of *connecting path* previously designed (see Fig. 3). Each of these connecting paths are built from a set of basic canonical maneuvers. Examples of these canonical maneuvers for car-like vehicles have been previously reported in (Gomez-Bravo et al., 2001), (Cuesta et al., 2004) and (Gomez-Bravo et al., 2008).

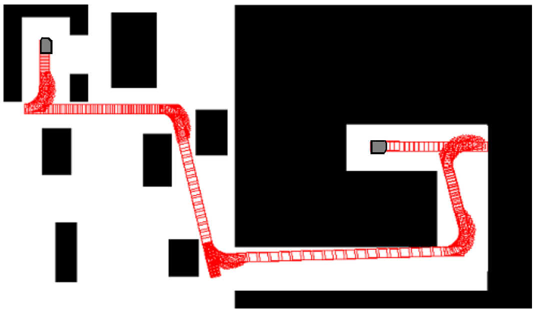


Figure 3: Final path.

As it is shown in (Gomez-Bravo et al., 2008), by means of this procedure, any two configurations can be connected by an admissible path. It is remarkable that, with this methods, the planner generates trajectories for both strategies: traditional continuous navigation, without stopping, or maneuvering navigation when necessary. That is, the planner includes inversors on the trajectory when a cluttered environment requires the vehicle to stop several times.

3 MANEUVERING NAVIGATION CONTROL

In this section the basis of the tracking method are illustrated. The particular problems of performing maneuvers are also addressed. Finally, the convenient modifications of the path following method for executing complex maneuvers are presented.

3.1 Continuous Path-tracking

Different approaches have been proposed for path following, where the vehicle position is estimated by an Extended Kalman Filter (EKF) (Grewal and Andrews, 1993) using Global Positioning System (GPS) (Daily and Bevly, 2004) and odometry.

In the approach presented in this paper, an accurate path-tracking strategy is accomplished by applying a modified version of the "Pure-pursuit" algorithm (Ollero, 2001). In this algorithm a point of the reference path is selected at each time instant so that the steering angle is obtained according to the expression:

$$\phi = \arctan(l \cdot \frac{2\triangle}{L_H^2}) \tag{4}$$

where l is defined in Fig. 1 and L_H is a parameter called the look-ahead, which represents the distance between the target point and the vehicle's current position, and $\triangle$ is the lateral distance between the robot and the target point (see Fig. 4).

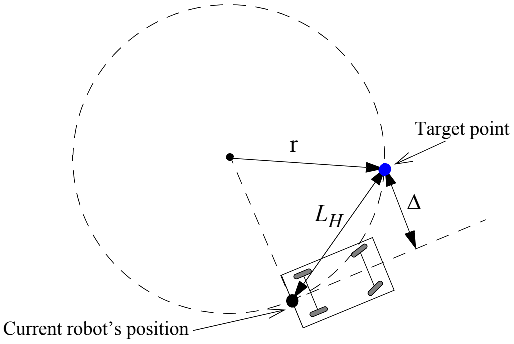


Figure 4: Circular trajectory.

With this driving strategy the vehicle will follow a circular arc from its current position to the target point. Usually the point of the path is selected by the following procedure: a) firstly, the nearest point, (x_n, y_n), to the vehicle is obtained; b) secondly, from (x_n, y_n) the target point (x_t, y_t) is found as the one which is at the distance L_H, from the current vehicle's position in the direction of the desired motion, (see Fig. 5). In this way at each time instant, from an estimated vehicle configuration, a new point of the path is selected and the vehicle navigates following the desired direction.

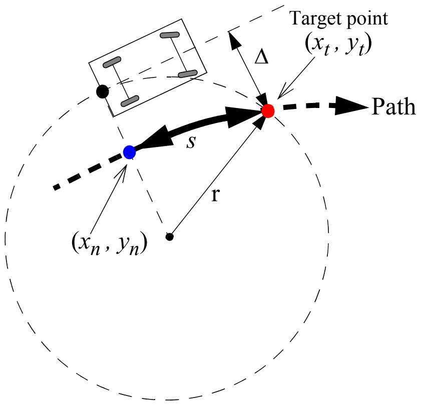


Figure 5: Pure-pursuit.

There could be different criteria for selecting the value of L_H according to the velocity and the shape of the path. In this approach the selection of the velocity and the look-ahead value is determined by the planner taking into account the characteristics of the planned trajectory.

3.2 Maneuvers Path-tracking

Traditionally, continuous navigation is based on a sequential procedure: first the planner provides a path connecting the starting point with the goal configuration and finally the tracking algorithm is applied so that the vehicle follows the trajectory until the vehicle arrives to the end of the path. In this way, path-following algorithms are usually designed so that the tracking error decrease as the vehicle follows the reference path. However, errors use to increase when the vehicle arrive to the goal configuration. Clearly, stopping the vehicle involves slipping phenomenon, particularly when the vehicle navigates outdoor on irregular terrains. Obviously, this is a drawback, specially when the vehicle is performing maneuvers as the vehicle has to stop several times. Thus, if a trajectory to be followed presents inversors, the traditional procedure need to be improved.

The present approach is based on developing two independent modules, the planner and the tracker, that take the responsibility of planning and tracking the maneuver respectively. These two modules will interact so that they manage the maneuver performance. The planner, once a initial trajectory has been generated, will divide it in *path sections*, separated by inversors. Each path section will have associated a kinematics profile and a look-ahead determined by the planner. The tracker will receive sequentially each of the path sections and will apply the tracking algorithm until the inversor configuration is reached. When the vehicle is stopped over a inversor the tracker will ask the planner for the next section of the path. This procedure will be repeated until the whole maneuver is performed.

Clearly, stopping the vehicle over the inversor is a very difficult task. A simple approach for managing the stopping process consist on decreasing, the vehicle velocity as it is closed to the inversor. The tracker will stop the vehicle when the position error is lower than a threshold. However a problem could appear when this strategy is applied. If the vehicle configuration and the inversor fall in a circular arc which radius is shorter than minimum curvature radius, R_{min}, the vehicle will be trapped in a circular trajectory in which the position error never decrease (see Fig. 6).

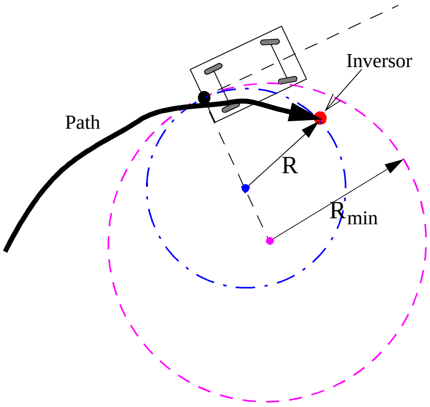


Figure 6: Pure pursuit problem.

Thus, the tracking algorithm will stop following that section of the path if this situation is detected, and asks the planner for the next section to follow.

4 DISTRIBUTED IMPLEMENTATION

Motion control of autonomous vehicles requires real time attention to different task. For instance, getting the GPS lecture, reading the proximity sensors, running the path tracking algorithm etc, are procedures that need a tailored use of the time into the control loop. Nevertheless, path planning usually presents a high computational cost. Then, implementing these algorithms and the vehicle control program into the same computer machine could negatively affect the distribution of the time control loop. In order to avoid this problem, in this approach, the planning algorithm is executed on a computer different to the one containing the autonomous vehicle control program. Thus a distributed strategy has been implemented by using an auxiliary server. This computer has been configured so that planning task are executed. Moreover other external interactions are also performed (collecting information from external sensors or attending human interface for instance). The novelty of this approach is based on the relation between the vehicle control program and the services running into the auxiliary server. As was mentioned in the previous section, there are parameters of the tracking algorithm (velocity and look-ahead) that are computed in the auxiliary server from the path provided by the planning algorithm.

The proposed implementation is shown in Fig. 7. The auxiliary server communicates with the autonomous vehicle by means of WIFI devices, using the TCP protocol. In this way the YARP protocols and services have been used. Considering the OSI Reference Model the YARP protocols could be placed on the Session Layer. This procedure allow transmitting object data defined in the C/C++ format. These protocols provide several net ports independent from the operative system. Thus, it represents a flexible solution for implementing the dialog between the navigation control algorithm and the process running at the auxiliary server, being independent from the computer platform.

The global planner program has two process running in parallel. The first one is the planner itself, i.e, the responsible of providing admissible paths. The second one, establishes and maintains the communication between the planner and the tracker through the YARP port. By means of this process the planner receives continuously the vehicle position and the goal point over the current path section. If the vehicle navigates too far from reference path or the scenario changes, the planner decides whether the current path section is computed again or not. As was described before, the path is divided into sections separated by the inversor configurations, and the vehicle follows sequentially the path sections. Thus, in case of any possible eventuality (new obstacles on the map, large position error or the vehicle being trapped by a circular trajectory), the planner modifies the affected path section. Only if these changes involve the current path section, the vehicle has to stop. Otherwise, the modifications are transmitted to the autonomous vehicle in a sequential procedure.

Additionally, the planner determines the value of the look-ahead and velocity associated to each section of the path. It establishes these values according to their characteristic, taking into account the curvature and the length. An heuristic procedure for setting the vehicle velocity has been implemented. Thus a long and straight path section can be followed with a high linear velocity whereas a short and curved section requires a low velocity value. Likewise, different criteria for selecting the Look-ahead according to the velocity and the shape of the path can be found in (Ollero et al., 1994) and (Ollero et al., 1996).

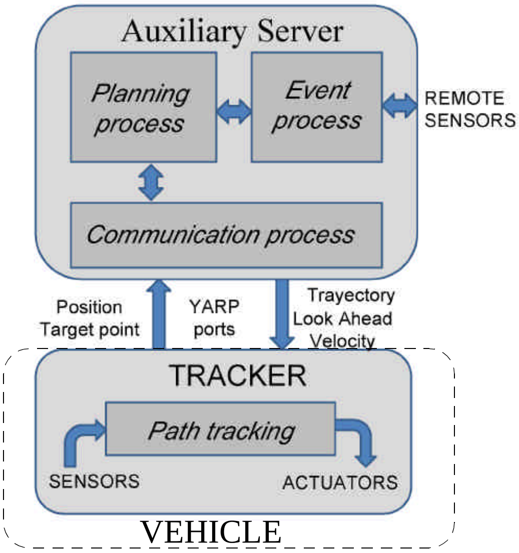
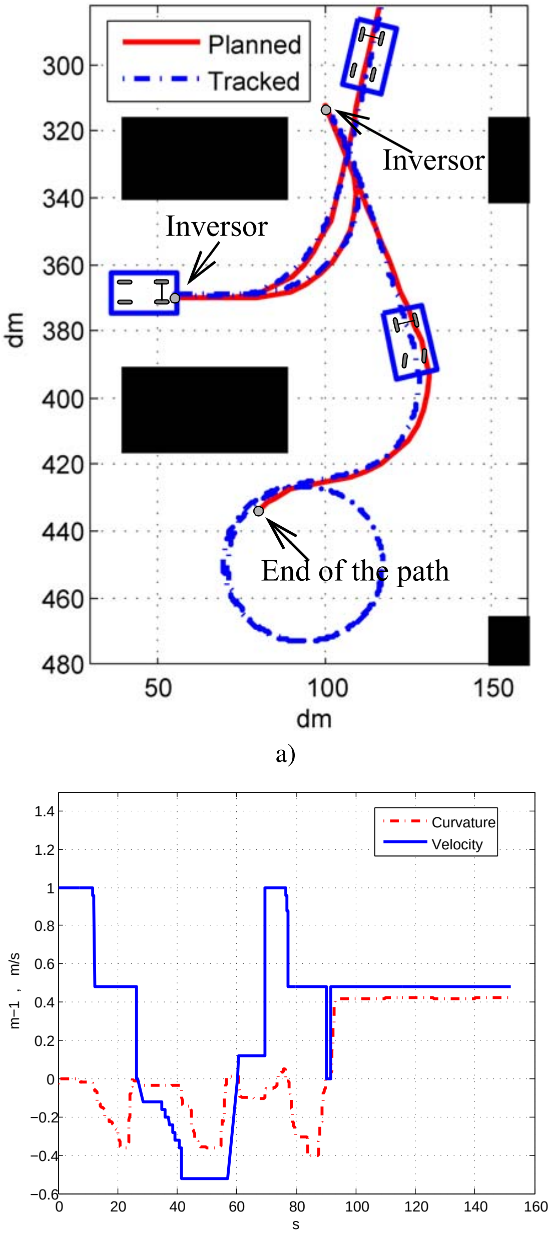


Figure 7: Distributed implementation.

5 EXPERIMENTAL RESULTS

The experiments presented in this section have been performed with ROMEO-4R, an autonomous car-like vehicle build at the Sevilla University (Ollero et al., 1999). The navigation control program runs into an industrial PC-Pentium III under Linux (Debian 2.6). The control program has been implemented in C++ and attends different concurrent process. Thus an EKF has been implemented so that GPS and odometry data are combined to obtain a robust pose estimation. At the same time, the path tracking algorithm is executed, applying the techniques described above, being continuously connected whit the planner program at the auxiliary server. Likewise, the autonomous vehicle also attends different type of proximity sensor (ultrasonics, LIDAR etc) that can be used for the improvement of the local obstacle avoidance. This technique is not addressed in the paper. The auxiliary server is implemented in a laptop computer, with a 1.8 Ghz AMD processor under Windows XP.

The planner and all the services running in the auxiliary server have been implemented in Java. Additionally, a connection with an external wireless sensor network has been also developed (Gomez-Bravo et al., 2007).

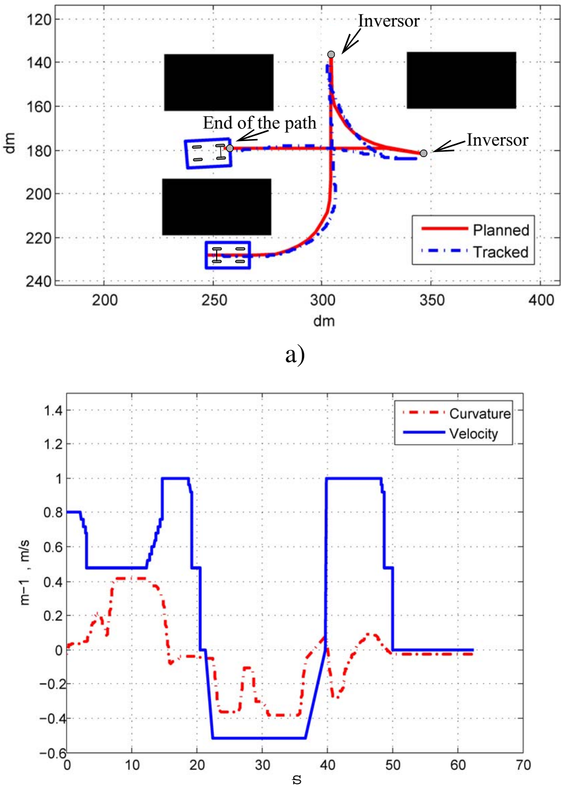


a)

b)

Figure 8: Experiment 1: a) Vehicle trajectory and b) Vehicle velocity and curvature.

Fig. 8 a) shows an experiment in which the path presents three sections and two inversors. In this experiment the vehicle fell trapped at the end of the path, it was specially configured in other to illustrate this type of situation. Fig. 8 b) presents the evolution of the velocity and curvature. Note that the velocity kept constant value as the vehicle navigated around the inversor and the curvature got the maximum value that represents the minimum curvature radius.



a)

b)

Figure 9: Experiment 2: a) Vehicle trajectory and b) Vehicle velocity and curvature.

In Fig. 9 a) a new experiment is shown. Due to the irregularities of the terrain, the vehicle motion is affected by perturbations that make the vehicle trajectory being different to the reference path. However, the vehicle finally achieved the final desired configuration. Fig. 9 b) presents the evolution of the velocity and the curvature. Observe how, velocity was decreased along the curved section and was increased when a straight section was followed.

Finally in Fig. 10 some pictures of ROMEO 4R performing a maneuver are shown. These pictures were recorded during the experiment presented in Fig. 9

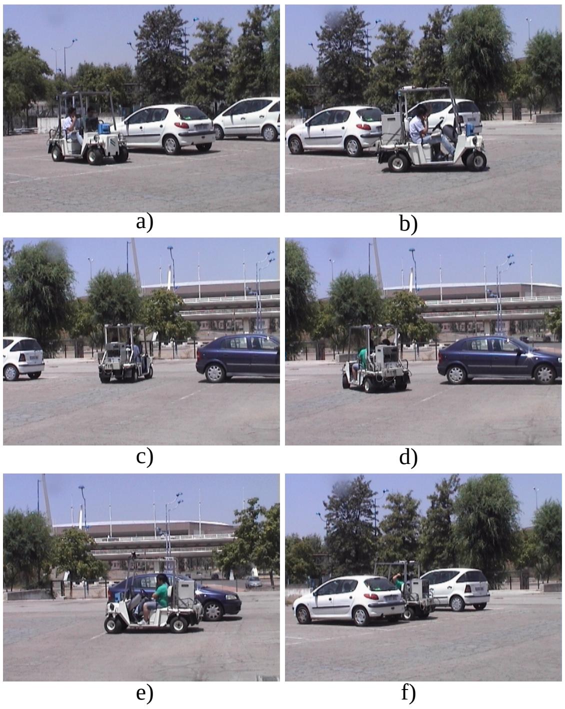


Figure 10: ROMEO 4R performing the second maneuver.

6 CONCLUSIONS

This paper presents a new approach for autonomous car-like vehicles maneuvering. The method allows navigation of robots with non-holonomic constraints in cluttered scenarios, providing, when necessary, continuous paths or complex maneuvers, where the vehicle has to change the sign of the velocity. This approach allows to distribute the computational task for computing the path and tracking the maneuver. Thus, a new implementation has been proposed so that the planning and the tracking algorithm exchange continuously data in order to enhance the maneuver performance. The method has been validated in real experiment with the autonomous car-like vehicle ROMEO-4R built at the Sevilla University.

ACKNOWLEDGEMENTS

This work has been supported by the National Spanish Research Program, project DPI2008-03847, and the Project URUS funded by the European Commission under grant IST-045062.

REFERENCES

Amato, N. M. and Wu, Y. (1996). A randomized roadmap method for path and manipulation planning. *IEEE Int. Conf. Robot. and Autom*, pages 113–120.

Barraquand, J. and Latombe, J. C. (1991). Robot motion planning: A distributed representation approach. *Int. J. Robot*, pages 167–194.

Bruce, J. and Veloso, M. (2002). Real-time randomized path planning for robot navigation. *International Conference on Intelligent Robots and Systems*, pages 2383–2388.

Cheng, P., Shen, Z., and LaValle, S. M. (2001). Rrt-based trajectory design for autonomous automobiles and spacecraft. *Archives of Control Sciences*, pages 167–194.

Cuesta, F., Gomez-Bravo, F., and Ollero, A. (2004). Parking manoeuvres of industrial-like electrical vehicles with and without trailer. *IEEE Trans. on Industrial Electronics*, pages 257–269.

Daily, R. and Bevly, D. (2004). The use of gps for vehicle stability control. *IEEE Trans. on Ind. Electron*, pages 270–277.

Gomez-Bravo, F., Cuesta, F., and Ollero, A. (2001). Parallel and diagonal parking in nonholonomic autonomous vehicles. *Engineering Aplication of Artificial Inteligence*, pages 419–434.

Gomez-Bravo, F., Ollero, A., Cuesta, F., and Lopez, D. (2007). Rrt-d: A motion planning approach for autonomous vehicles based on wireless sensor network information. *6th IFAC Symposium on Intelligent AutonomousVehicles*, page C.D.

Gomez-Bravo, F., Ollero, A., Cuesta, F., and Lopez, D. (2008). A new approach for car-like robots manoeuvering based on rrt. *Robotica*, pages 10–14.

Grewal, M. S. and Andrews, A. P. (1993). *Kalman Filtering Theory and Practice*. Prentice-Hall.

Latombe, J. C. (1991). *Robot Motion Planning*. Kluwer Academic Pulisher.

Laumont, J., Jacobs, P., Taix, M., and Murray, M. (1994). A motion planner for nonholonomic mobile robots. *IEEE Trans. on Robotics and Autom*, pages 577–593.

LaValle, S. M. (1998). Rapidly-exploring random trees: A new tool for path planning. In *TR 98-11*.

LaValle, S. M. (2006). *Planning algorithms*. Cambridge University Press.

LaValle, S. M. and Kuffner, J. J. (1999). Randomized kinodynamic planning. *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 473–479.

Ollero, A. (2001). *Manipuladores y Robots Moviles*. Marcombo Boixareu.

Ollero, A., Arrue, B. C., Ferruz, J., Heredia, G., Cuesta, F., Lopez-Pichaco, F., and Nogales, C. (1999). Control and perception componenets for autonomous vehicle guidance. application to the romeo vehicles. *Control Eng. Practice*, pages 1291–1299.

Ollero, A., Garcia-Cerezo, A., and Martinez, J. (1994). Fuzzy supervisory path tacking of autonomous vehicles. *Control Engineering Practice*, pages 313–319.

Ollero, A., Garcia-Cerezo, A., and Martinez, J. (1996). Design of a robust high performance fuzzy path tracker for autonomous vehicles. *Journal of Systems Science*, pages 799–806.

Paromtchik, I. E., Damm, M., and Matioukhina, L. I. (1998). Autonomous maneuvers of a nonholonomic vehicle. *Intelligent Autonomous Systems. International Scientific Issue*, pages 38–45.

Wada, M., Yoon, K. S., and Hashimoto, H. (2003). Development of advanced parking assistance system. *IEEE Trans. on Ind. Electron*, pages 4–17.

EXPRESSION OF EMOTIONS THROUGH BODY MOTION
A Novel Interface For Human-Robot Interaction

Nelson Gonçalves and João Sequeira
Institute of Systems and Robotics, Insituto Superior Técnico, Lisbon, Portugal
ngoncalves@isr.ist.utl.pt, jseq@isr.ist.utl.pt

Keywords: Human-robot interaction, Emotions.

Abstract: An approach is presented for the expression of basic emotions through only the agent body pose and velocity. The approach is applied in human-robot interaction scenarios, where both humans and robots communicate only through their relative position and velocities. As a result, an interface for human-robot interaction is obtained, which does not require the use of haptic devices or explicit communication with humans, verbal for instance. The small set of emotions that can be conveyed enable humans and robots to anticipate the intentions of the opponent and adapt their behavior accordingly. The aproach is implemented using a webcam, simple vision processing algorithms and Hidden Markov models. The results of preliminary experiments are presented.

1 INTRODUCTION

The problem considered in this paper is the recognition of emotions in human-robot interaction (HRI) scenarios without explicit communication or the usage of haptic devices.

Such HRI problems can occur in many common applications. An example is that of a mobile robot advertising and selling products in a supermarket. Although it can move directly towards approach potential clients, this behavior may be considered too intrusive and unpleasant. Therefore, the robot must first estimate the interest of the clients without using haptic or voice interfaces. If a reasonable interest is perceived, the robot should then approach the clients.

Another application of interest is active surveillance, where mobile robots must intercept and identify intruders. These are not expected to cooperate and can even sabotage the robots. The intentions of the intruders must then be estimated at a safe distance and without explicit communication. In this application, the mobile robots can move aggressively, directly towards the intruders at high velocity. The purpose is to intimidate them and also to block potential exit pathways. In these applications, the use of traditional interface devices, such as voice or touch, is not efficient. The main reason is that in these applications, humans and robots keep some distance between them during most of the time. Another reason is that explicit communication, verbally for instance, may not be possible due to ambient background noise.

The proposed approach is to express and perceive emotions through the body pose and velocity. A friendly emotion can be expressed through a smooth path, executed at a low velocity. The antagonistic emotion of anger, may be expressed through sharp, discontinuous paths performed at a high velocity. An advantage of the proposed approach is an increase of the available bandwidth for human-robot communication, since the body motion is another possible communication channel. Another advantage is that agents can perceive the intentions of opponents at some distance and adapt their behaviors accordingly. This is relevant to robots in adversarial environments.

The remainder of this paper is as follows. A review of the literature is presented in Section 2. In Section 3 the nature of emotions and their forms of expression are discussed. A classifier for the recognition of emotions is presented in Section 4, which is evaluated in a set of preliminary experiments described in Section 5. Finally, in Section 6 the approach is discussed and future work is presented.

2 RELATED WORK

In the HRI problem considered, humans are not expected to explicitly communicate with robots or to use haptic interfaces. This is an uncommon scenario in HRI applications, (Fong et al., 2003; Goodrich and Schultz, 2007), where typical interfaces make use of voice, touch and human facial expressions. Nevertheless, the information conveyed through the body pose and velocity was considered in (Breazeal, 2003) and applied in practice in (Finke et al., 2005).

An early study on the expression of emotions in both humans and animals was conducted by Darwin in (Darwin, 1872). It is reported that the human body motion and stances, when expressing an emotion, are similar to when acting in accordance. For example, the body stances when expressing anger are almost identical to those when preparing for an actual attack. Also, it is well known that the state of mind has a strong influence on the motion of a person, (Nakamura et al., 2007). This is often exploited in computer animation to increase the realism of human characters, (Becheiraz and Thalmann, 1996; Neff and Fiume, 2006).

In neuro-psychological studies of human emotion, facial expressions typically receive much more attention than other forms of expression, (de Gelder et al., 2004). But in (Atkinson et al., 2004), it was found that emotions could be recognized from static and dynamic body stances. This was case also when human motion was represented using only a cloud of points. Finally, in (den Stock et al., 2008) the body motion was also found to bias the recognition of bimodal emotions from sound and vision cues.

3 EXPRESSION OF EMOTIONS

The nature of emotions is, to the best of the authors knowledge, an unsolved problem. Therefore, in this section an attempt is made to understand the nature emotions and how they can be perceived and expressed.

In the pioneer work by Darwin, (Darwin, 1872), and William James, (James, 1884), is argued that at least some emotions are a form of instinctive reaction to stimuli received from the environment. The reason is the similarity in some expressions among humans from very different cultures. Also, it is not plausible that a conscientious process it at the origin of emotions in animals. Nevertheless, since these initial contributions many other definitions of emotion have been unsuccessfully proposed, (Scherer, 2005).

Although the question of what is an emotion is yet unanswered, it is more relevant for the HRI problem to answer questions related to the causes of emotions. The answer to these questions is stated in terms of the causation categories by Aristotle, (Russell, 2004). If these are known, then suitable models can be build and used for perceiving and expressing emotions. The first question to be posed is: "why do humans express emotions ?". A possible answer is given in terms of the final causation category:

Assumption 1 (Manifestation of Emotions). *The final cause of an emotion is the change in the agent state, perceptible to external observers.*

The final causation category is identified with the concepts of purpose and ultimate goals. Thus, in this paper it is assumed that the purpose of an emotion is to be announced to others, through a change in the agent state. It is clear that other answers are possible if other causes are identified. The answer could be given in terms of specific hormones or physiological mechanisms, such as in (Scherer, 2005). These answers belong, respectively, to the material and efficient causation categories.

The final causation category is used because an important design guideline is obtained. That the emotions an agent can express do not form part of the state. In order to understand this argument, consider the case where emotions form part of the agent state. Then the sequence of emotions being expressed is uniquely determined by the state past history and dynamics. As a result, the agent state changes could be known in advance and there would be no need to express them. Therefore, in this paper emotions are considered not part of the agent state, but instead part of the agent actions. The difference between emotions and other actions is that the former cannot be applied to the environment. As the result of this discussion, a definition for emotions is obtained.

Definition 1 (Agent Emotion). *An emotion is an action executed by the agent on his state, producing an externally perceptible state change.*

This definition is useful for the design of HRI interfaces. For an application example, consider the case where facial expressions are used to express emotions. Let the state of the agent, human or robot, be the configuration of the mouth and the eyebrows. An emotion is then the act of displaying a particular configuration of the mouth and eyebrows. Similarly, emotions can be perceived by identifying the respective state configurations.

The previous definition does not provide clues on how the state is altered through an emotion. Thus, the next question is: "why is an emotion expressed through some state changes and not others ?". The answer is given using the efficient causation category, which is related to the concepts of method or function. A possible answer is then that the forms of emotion expression in humans are function of evolutionary pressures. An immediate consequence is that un-

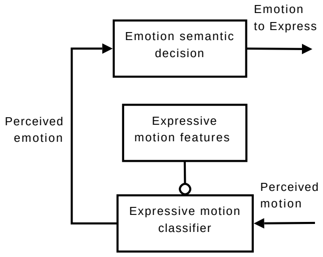

Emotion semantic decision
Emotion to Express
Perceived emotion
Expressive motion features
Perceived motion
Expressive motion classifier

Figure 1: Architecture for emotion perception and decision.

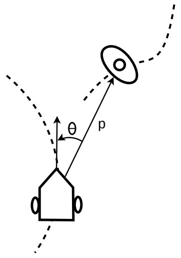


Figure 2: Typical HRI situation.

der different environments, different forms of expression would emerge. Another consequence is that a learning algorithm could be employed to determine the human forms of expression. But in general, it would require that humans and robots interact during an excessively long period of time. Therefore, it is more practical to mimic, where possible, the forms of expression in humans and domestic animals.

4 PERCEPTION OF EMOTIONS

In the remainder of the paper and when clear from context, humans and robots are both referred to as agents. It is assumed that agents move in a 2D plane. Furthermore, robots are assumed not to possess any anthropomorphic features.

The proposed architecture for the perception of emotions is presented in Figure 1. The motion of humans is perceived and classified using features of interest defined *a priori* by the system designer. Thus the recognition problem can be formulated without knowledge on the semantics of emotions, since from Definition 1, only state changes must be perceived. After the type of emotion is perceived, the robot must decide which emotion to display. In this step is required knowledge of the context and also the meaning of emotions. The solution is presented in Sub-Section 4.2, where the notion of empathy is used.

4.1 Expressive Motion Classifier

The design of the classifier of emotions from the human body motion is formulated as time series classification problem. The human body is approximated by the geometric center and the features of interest are the human pose and velocity relative to the robot. This choice is based on the expression of emotions by human actors described in (Atkinson et al., 2004) and the social distances presented in (Becheiraz and Thalmann, 1996; Pacchierotti et al., 2006). An accurate estimation of the features values is not required. The reason is that the basic emotions, such as fear and disgust, are fundamental to survival and are expressed with clear state changes.

A typical situation for HRI through motion is depicted in Figure 2. The mobile robot is represented by the polygonal shape and both agents are moving at different linear velocities. The robot is able to measure the relative pose of the human, at a constant rate Δ^{-1}. Since the robot is also moving, it will perceive an apparent motion of the human. This effect must be corrected to prevent erroneous classifications.

Consider a static frame $\{w_k\}$, which is coincident with the robot body frame at time t_k. Let $p(t_k)$ be the position of the human measured by the robot and assume that the human is static. Then, for a small enough interval Δ, at t_{k+1} the measured value should be

$$\hat{p}(t_{k+1}) = R(\omega(t_k)\Delta)(p(t_k) + v_r(t_k)\Delta) \qquad (1)$$

where $v_r(.)$ is the robot linear velocity, $\omega(.)$ the angular velocity about the robot frame origin and $R(.)$ is the rotation matrix from frame $\{w_k\}$ to the robot frame at time t_{k+1}. Let $p(t_{k+1})$ be the actual measured value by the robot at time t_{k+1}. If the human is not stationary, then the predicted and measured values are not equal and their difference is due to the human velocity

$$v_h(t_k) = (\hat{p}(t_{k+1}) - p(t_{k+1}))\Delta^{-1} \qquad (2)$$

The vector of observed motion features is then

$$f_k = (\|p(t_k)\|, \theta(t_k), \|v_h(t_{k-1})\|) \qquad (3)$$

where $\theta(t_k) = atan(p(t_k))$. The first two features model static properties of the expression of emotions, which are linked to focus of the agent on the observer. The relative velocity of the human is related to the intensity of the emotion.

The block diagram of the emotion classifier is presented in Figure 3. The input is an array of feature vectors, $m_{[k,k+n]} = (f_k, f_{k+1}, \dots, f_{k+n})$. Through vector quantization, each feature vector f_k is replaced by a symbol s_k. Then the probability of each Hidden Markov Model e_i generating the sequence of symbols, $s_{[k,k+n]}$, is computed with the forward algorithm. The

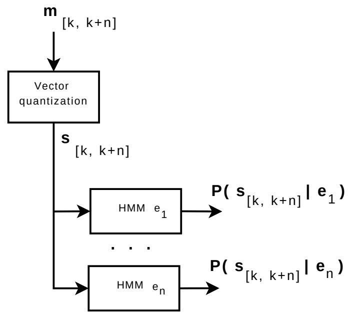


Figure 3: Emotion classifier from motion.

output of the classifier is a vector with the normalized value of these probabilities, $P(s_{[k,k+n]}|e_i)$.

A similar approach was used in (Takeda et al., 2007) with good results in estimating the next dance step for a robotic dance partner.

4.2 Emotion Semantic Decision

The expression of emotions in humans is closely linked to similar instinctive behaviors, (Darwin, 1872; James, 1884; Scherer, 2005). Therefore, it is reasonable to use a reactive approach to determine the emotion to express. The proposed solution is to make use of the concept of empathy. The emotion expressed by the robot, e^*, is the one assigned the highest probability by the classifier

$$e^* : \max_i \{P(s_{[k,k+n]}|e_i)\} \tag{4}$$

This is a straightforward solution and does not require knowledge of the emotion semantics or context. A similar method is used in (Takeda et al., 2007), where the dance step is selected based on the ratio between the two highest probabilities. If it is above some threshold the step associated to the highest probability is executed, otherwise the robot wheels are stopped. This method is not suitable for expressing emotions through motion because stopping can be perceived as an emotion, fear for instance. It is also not robust to classification errors and does not enable robots to take the initiative. The latter is an important aspect in general HRI problems. Since most humans are not familiar with robots, they may not expect an autonomous behaviors from these machines. The original solution can be improved by minimizing a decision cost

$$e*(\gamma) : \min_j \left\{ \sum_i c_{ij}(\gamma) P(s_{[k,k+n]}|e_i) P(e_i;\gamma) \right\} \tag{5}$$

where $c_{ij}(.)$ is the cost of expressing emotion e_i instead of emotion e_j and $P(e_i;\gamma)$ is the *a priori* probability of observing emotion e_i. The discrete parameter γ is used to define the context of the mobile robot application. For instance, in surveillance applications it is reasonable to expect humans to behave aggressively. The probability of observing anger is then greater than that of happiness and the robot should also prefer also to express hostility over happiness.

5 EXPERIMENTAL RESULTS

A set of experiments where conducted to evaluate the emotion classifier, with results presented in this section. The interface was implemented in C++ language using a standard, of-the-shelve webcam mounted on top of a Pioneer P3-AT robot. The purpose is to evaluate the classifier from the robot world perspective view. The experiments where performed with a human wearing a bright, green colored vest to facilitate the detection of color blobs. The webcam was calibrated to measure the distance and angle of the human under the assumption that the height of the hip is constant. The blob detection is affected by high frequency noise because the vest surface is wrinkled and is not perceived with an uniform color. A median filter was applied the values of the blob centroid, to remove some of the noise. The maximum sampling rate was approximately 6 samples per second, much slower than the rate in (Takeda et al., 2007) for instance.

The parameters for the vector quantization procedure where determined by hand, based on the social distances discussed in (Becheiraz and Thalmann, 1996; Pacchierotti et al., 2006). The values of the relative position and angle are quantized in $\{1.0, 2.0, 3.5, 4.5, 5.5\}[m]$ and $\{-40^o, 0^o, 40^o\}$, respectively. The norm of the relative velocity is quantized in $\{0.5, 1.5, 2.0, 3.0, 4.5\}[m/s]$, where $0.5\,m/s$ roughly corresponds to the human being stopped. With respect to the use of clustering algorithms, this approach does not require a large amount of data to be properly trained. Also, the quantization values determined by the algorithms may not reflect the social distances used by humans.

The emotions considered in the experiments where: (i) anger, (ii) fear, (iii) friendliness and (iv) apathy, which can be understood as the agent not expressing any emotion. Their expression was exemplified by a human in front of the robot. The human ran towards the robot to express anger, while friendliness was expressed with a normal pace. In order to express fear, the human moved toward the robot but at halfway stopped and moved away. The expression of apathy was exemplified with the human moving parallel to the webcam image plane or away from

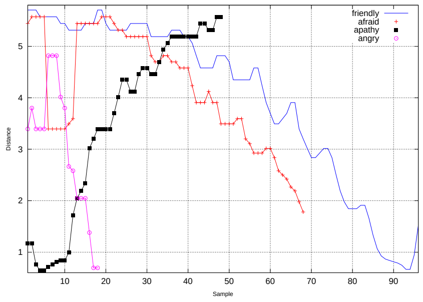


Figure 4: Example of feature $\|p(t_k)\|$ for each emotion.

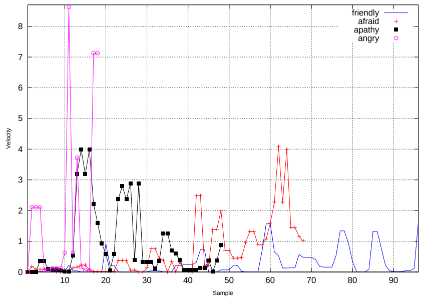


Figure 5: Example of feature $\|v_h(t_{k-1})\|$ for each emotion.

the robot. A total of twenty videos where recorded, with five examples for each emotion. The features $\|p(t_k)\|$, $\|v_h(t_{k-1})\|$ and $\theta(t_k)$ are plotted in Figures 4 to 6, taken from an example of each emotion. In these figures is visible that the distance and angle features produced distinctive sequences for each of the emotions emotion. For example, friendliness and fear produce similar distance sequences but clearly distinct sequences of angles. The estimation of the values for the velocity feature is sensitive to the noise and detection failures of the vision system. Another source of error is the height of the hip which as small variations during the motion. Although some patterns are visible in Figure 5 for each emotion, the sequences of values are very irregular and with also abnormally high values.

The HMM of each emotion was trained using the quantized sequence of features from all the videos. Each HMM is composed with five states and a left-right transition structure. After the training phase, the emotion classifier was evaluated using all of the emotion examples. The prior probabilities of each emotion are equal, $P(e_i;\gamma) = 0.25$, and the elements of the decision cost matrix are all unitary. In Table 1 are summarized the classification for each set of videos. The numbers between parenthesis in the first column represent the total number of sequences, $s_{[k,k+n]}$ per set of videos. The numbers in the other columns represent the number of times the corresponding emotion

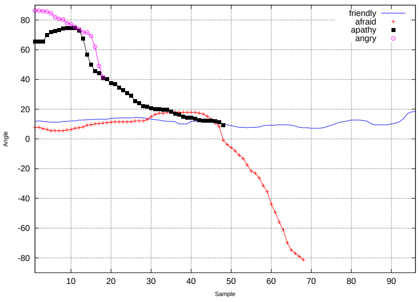


Figure 6: Example of feature $\theta(t_k)$ for each emotion.

was perceived. From Table 1, the emotions of friendliness and fear had the highest number of correct classifications but not anger and apathy. The main reason is that these emotions have sequences with much smaller dimensions, see Figure 4 for instance. Nevertheless, the data in Table 1, is useful to determine the values for c_{ij} and $P(e_i;\gamma)$, which reduce the classification errors. These can also, to some degree, be handled by the selection and expression of emotions in the robot. For instance, the first action of the robot can be to stop and observe the human in order to reduce classification errors. This is a common trait in the expression of fear in humans, for example.

6 CONCLUSIONS

An HRI interface for the expression of emotions through body motion was presented. The approach was implemented in practice with a standard vision system. Despite the simplicity of the implementation, acceptable results where obtained. In addition, the bottlenecks of the system performance where identified. Thus, given more efficient feature estimation methods, it is reasonable to assert the feasibility of the proposed HRI interface. Since human body motion is emotionally charged, (Atkinson et al., 2004), no prior training in robotics or specially designed hardware is required in this HRI interface. Thus, it can be used in HRI applications where humans are un-skilled in mobile robotics.

The interface is valuable also to other mobile robot applications. As argued by António Damásio, in (Damasio, 2006) and elsewhere, emotions are fundamental to successful decision making in humans. Thus the ability to express them without the need for additional hardware is by itself a feature of interest. Since any movement of the agent can be perceived as an emotion, knowledge of the application context is required for disambiguation purposes. The discrete parameter γ was introduced to account for the appli-

Table 1: Emotion classifier results with $c_{ij} = 1.0$ and $P(e_i;\gamma) = 0.25$.

Video Set \ Emotion	*Friendly*	*Afraid*	*Apathy*	*Anger*
Friendly (23)	10	4	5	4
Afraid (31)	9	11	8	3
Apathy (11)	3	3	1	4
Anger (13)	3	3	7	1

cation context.

The vision system low frame rate and the detection failures had a negative impact on the system performance. Therefore, future work is aimed at increasing the frame rate and the robustness of the human detection. For instance, through better the use of hardware and a detection algorithms, such as a face detector. Also, the approach must be evaluated using groups of humans with different backgrounds in mobile robotics.

ACKNOWLEDGEMENTS

This work was supported by European Project FP6-2005-IST-6-045062-URUS, and Fundação para a Ciência e a Tecnologia (ISR/IST pluriannual funding) through the POSConhecimento Program that includes FEDER funds. Nelson Gonçalves is working under grant SFRH/BD/23804/2005, from Fundação para a Ciência e a Tecnologia.

REFERENCES

Atkinson, A. P., Dittrich, W. H., Gemmell, A. J., and Young, A. W. (2004). Emotion perception from dynamic and static body expressions in point-light and full-light displays. *Perception*, 33:717–746.

Becheiraz, P. and Thalmann, D. (1996). A model of nonverbal communication and interpersonal relationship between virtual actors. In *Computer Animation '96. Proceedings*, pages 58–67.

Breazeal, C. (2003). Emotion and sociable humanoid robots. *International Journal of Human-Computer Studies*, 59(1-2):119–155.

Damasio, A. (2006). *Descartes' Error*. VINTAGE RAND.

Darwin, C. (1872). *Expression of the Emotions in Man and Animals, The*. Oxford University Press Inc, 3 sub edition.

de Gelder, B., Snyder, J., Greve, D., Gerard, G., and Hadjikhani, N. (2004). Fear fosters flight: a mechanism for fear contagion when perceiving emotion expressed by a whole body. *Proc Natl Acad Sci U S A*, 101(47):16701–16706.

den Stock, J. V., Grezes, J., and de Gelder, B. d. (2008). Human and animal sounds influence recognition of body language. *Brain Research*, 1242:185–190.

Finke, M., Koay, K. L., Dautenhahn, K., Nehaniv, C. L., Walters, M. L., and Saunders, J. (2005). Hey, i'm over here - how can a robot attract people's attention? In *Robot and Human Interactive Communication, 2005. ROMAN 2005. IEEE International Workshop on*, pages 7–12.

Fong, T., Nourbakhsh, I., and Dautenhahn, K. (2003). A survey of socially interactive robots. *Robotics and Autonomous Systems*, 42(3-4):143–166.

Goodrich, M. A. and Schultz, A. C. (2007). Human-robot interaction: a survey. *Found. Trends Hum.-Comput. Interact.*, 1(3):203–275.

James, W. (1884). What is an emotion ? *Mind*, 9:188–205.

Nakamura, T., Kiyono, K., Yoshiuchi, K., Nakahara, R., Struzik, Z. R., and Yamamoto, Y. (2007). Universal scaling law in human behavioral organization. *Physical Review Letters*, 99(13):138103.

Neff, M. and Fiume, E. (2006). Methods for exploring expressive stance. *Graphical Models*, 68(2):133–157.

Pacchierotti, E., Christensen, H. I., and Jensfelt, P. (2006). Evaluation of passing distance for social robots. In *Robot and Human Interactive Communication, 2006. ROMAN 2006. The 15th IEEE International Symposium on*, pages 315–320.

Russell, B. (2004). *History of Western Philosophy (Routledge Classics)*. Routledge, 2 edition.

Scherer, K. R. (2005). What are emotions? and how can they be measured? *Social Science Information*, 44(4):695–729.

Takeda, T., Hirata, Y., and Kosuge, K. (2007). Dance step estimation method based on hmm for dance partner robot. *Industrial Electronics, IEEE Transactions on*, 54(2):699–706.

VISUAL ATTENTION IN 3D SPACE
Using a Virtual Reality Framework as Spatial Memory

M. Zaheer Aziz and Bärbel Mertsching
GET Lab, Universität Paderborn, 33098 Paderborn, Germany
aziz@get.upb.de, mertsching@get.upb.de

Keywords: Visual attention, Virtual reality simulator, 3D space, Spatial memory, Robotic vision.

Abstract: This paper presents a conceptual framework to integrate a spatial memory, derived from a 3D simulator, with a visual attention model. The proposed system is inspired from brain research that explicitly accounts for the use of spatial memory structures in intelligent object recognition and navigation by humans in the three dimensional space. The experiments presented here extend the capability of visual attention modeling to work in 3D space by connecting it to simulated maneuvers in virtual reality. The introduction of this concept opens new directions for work to reach the goal of intelligent machine vision, especially by mobile vision systems.

1 INTRODUCTION

Spatial memory is an important part of the human brain that is responsible to store three dimensional structures of environments, landmarks, and objects (Moscovitch et al., 2005). The stored environments in this memory help during navigation through known routes and maps like walking through corridors or driving through streets of everyday routine. The object data in this memory is also useful for collision avoidance in an automatic way, for example during car driving a decision to overtake a long vehicle is made quite involuntarily after estimating the vehicle's length using its memorized 3D model recalled by looking at its rear only.

Construction of scenes and objects in the spatial memory has a close relation with visual attention. Research on vision systems of primates reveals that the natural vision views and recognizes objects (especially large ones) by fixating on their constituent parts rather than perceiving them as a whole. This is managed by the visual attention mechanism that selects salient portions of objects (or scenes) and focuses upon them one after the other. In artificial vision systems, such selective viewing can help to filter out redundant and non-relevant data.

This paper presents design of a memory driven vision system that integrates artificial visual attention with a 3D spatial memory. A robotic vision system able to perform overt visual attention will focus on salient objects or their visible parts and use this visual information to activate the complete 3D model of the object from the spatial memory. Utilization of a virtual reality simulation framework is proposed as storage mechanism for the learned environments and objects. Such a proposal not only leads to knowledge driven machine vision but introduces a very useful utility of 3D simulation engines as well.

Literature in psychophysics has described the role of spatial memory and its role in navigation, object recognition, self localization, and intelligent maneuvers. The work presented in (Oman et al., 2000) shows capability of learning three dimensional structure not only by looking at the target object or environment from different directions but by imagining to view it from these orientations as well. The ability of the brain to visualize a scene from an orientation in space that was not actually experienced by it is shown in (Shelton and Mcnamara, 2004). A relation between visual attention and spatial memory is established in (Aivar et al., 2005) with a conclusion that a detailed representation of the spatial structure of the environment is typically retained across fixations and it is also used to guide eye movements while looking at already learnt objects. Formation of object representation by human vision through snapshots taken from different view angles is discussed in (Hoshino et al., 2008) and it is suggested that such procedure is followed only for the objects under visual attention while the unattended scene may be processed as a 2-D representation bound to the background scene as a texture.

The natural visual attention mechanism rapidly analyzes the visual features in the viewed scene to determine salient locations or objects (Treisman and

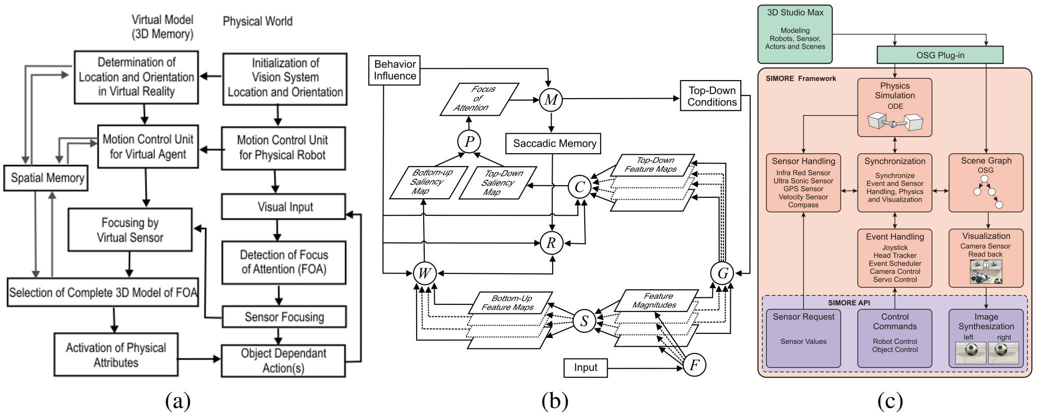


Figure 1: (a) Proposed model for integrating visual attention with 3D spatial memory. (b) Architecture of the region-based attention model used as the selection mechanism for regions of interest in real world (Aziz and Mertsching, 2007). (c) Architecture of the simulation framework SIMORE that is used as spatial memory of the mobile robot (Kutter et al., 2008).

Gelade, 1980)(Wolfe and Horowitz, 2004). After attending the current focus of attention a process of inhibition of return (IOR) (Cutzu and Tsotsos, 2003) suppresses the attended location so that other less salient objects may also get a chance to be attended. Existing attention models have shown success in selection and inhibition of return in two dimensional view frames. The natural visual attention, on the other hand, works in three dimensional world despite its perception of a two dimensional projection on the retina. In order to make advancement in the state-of-the-art, the model proposed in this paper attempts to integrate a spatial memory with the visual attention process in order to extend the scope of attention and IOR towards 3D.

2 PROPOSED MODEL

The objective of the current status of the proposed model is to associate a spatial memory to the visual attention process and activate the three dimensional structure of the attended object for use in decision making procedures. Figure 1(a) shows the architecture of the proposed model. As this model involves visual attention and a spatial memory to perform its task, the design of the two involved components is also discussed here.

The architecture of the attention model is presented in figure 1(b). The primary feature extraction function F produces a set of regions (Aziz and Mertsching, 2006) and associates feature magnitudes of color, orientation, eccentricity, symmetry, and size with each region. Computation of the bottom-up saliency using rarity criteria and bottom-up contrast of region features with respect to its neighborhood is performed by the group of processes S (see (Aziz and Mertsching, 2008a) for details) whose output is combined by the procedure W. The function G considers the given top-down conditions to produce fine grain saliency maps that are combined by the function C. The function P combines the saliency maps into a master conspicuity map and applies a peak selection mechanism. Inhibition of return (IOR), denoted by R, suppresses the already attended location(s) using a saccadic memory. Explanation of the internal steps and functions of this attention model can be seen in (Aziz and Mertsching, 2007) and (Aziz and Mertsching, 2008b).

The simulation system used as spatial memory is a 3D robot simulation framework SIMORE (SImulation of MObile Robots and Environments) developed in our group (Mertsching et al., 2005) (Kutter et al., 2008). Figure 1(c) shows its architecture with its major components exposed. The core of simulator is based upon the open source library Open Scene Graph (OSG) (Burns and Osfield, 2004) which is a hierarchical graph consisting of drawable meshes in a forward kinematic order. The physics simulation component represents the dynamic engine for collision detection and force based physics using Open Dynamics Engine (ODE) which is an open source library for simulating rigid body dynamics (Smith, 2009). Extensions for sensor and meta information have been done using specialized nodes for these purposes. These enhancements of the existing scene graph allows to rely on an existing library and enables the system to import and export from available 3D modeling software such as 3D Studio Max.

The simulation framework SIMORE has the ability to maneuver a simulated robot in the virtual environment by driving, taking turns, rotating its camera head, and turning other movable sensors by control commands from an external computer program.

The master control program, for example a computational model of visual attention, manipulates the physical robot in the real world and maneuvers the virtual agent in the simulation framework. Therefore the simulated robot can act as an agent of the real platform. The readings from the simulated sensors are obtained according to their position and direction in the virtual environment while they are aligned with the physical ones. Such a mechanism allows the vision system to recall a complete 3D model of an attended object even by looking at only a part visible from the current view angle.

According to the proposed model, the vision system selects an object to attend from its viewed scene and performs overt attention using its pan-tilt camera. Whenever the vision system finds an object of interest (or its part) it consults the spatial memory by looking at the virtual scene through its simulated camera. The visible features of the attended object in the real camera view, information about location of robot (in real and virtual environment), and angles of camera direction can guide to pick the right object from the virtual scene matching with the object under the focus of attention. Knowing the identity of the modeled object its complete set of attributes will be loaded into the working memory. Using the current status of our experimental platforms we demonstrate a spatial inhibition of return on the previously focused object(s) so that they remain inhibited even after the robot motion in 3D space.

3 CURRENT SYSTEM STATUS

In the current status, the interface between the visual attention module and the simulation engine is successfully established and work is underway to enable the synchronized selection of the attended objects from the spatial memory. We are able to present here the expected results from the proposed model with manual configurations in the synchronization part.

Figure 2 shows the arrangement in which the vision system appearing at the right side of the subfigure (a) drives forward while searching for red objects (the ball and the robot in the left-bottom corner). Figure 2(b) shows the global camera view of the arrangement in the simulation framework.

Results of the first attempt of attentional search are provided in figure 3 in which subfigure (a) shows camera view of real robot and (b) is the view from the aligned simulated camera. Figure 3(c) shows the region selected by the attention mechanism and (d) shows the camera view after bringing the ball into center of view frame. Figure 3(e) shows the virtual camera view after aligning it with the current status of the real camera. Figure 3(f) demonstrates the selected 3D model from the spatial memory whose activation not only exposes its hidden portion to the vision system but its volume information as well.

Figure 4 shows results of the second attempt of attention after moving ahead subsequent to attending the first target (the ball). The first focus of attention remains under inhibition of return during this attempt (shown by dark (blue) rectangle). In subfigure (a) the vision system moves ahead but no new object of interest comes into view, whereas the already attended ball

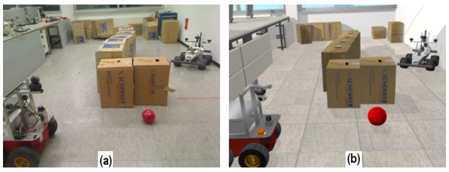


Figure 2: Initialization of the robotic platform and alignment of the agent in terms of location and orientation. (a) Real robot at initialization (b) Global view in virtual reality.

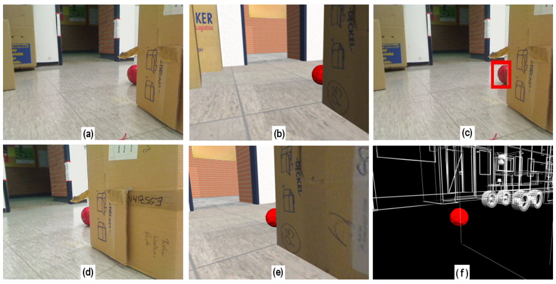


Figure 3: Results of first attempt of attention. (a) Camera view of real robot (b) Camera view of virtual robot (c) Focus of attention detected by real robot (d) Overt attention to object of interest (e) View of the simulated environment (spatial memory) after synchronized rotation of the simulated camera (f) 3D model of first FOA activated (inactive objects are shown in wireframe).

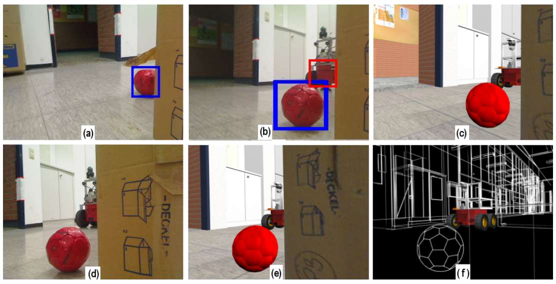


Figure 4: Results of second attempt of attention. (a) Camera view of real robot after moving ahead (previous FOA inhibited). (b) After moving further ahead the second target found (previous FOA still remains inhibited). (c) Aligned camera view of virtual robot. (d) Overt attention to second object of interest. (e) Orientation synchronization of simulated camera. (f) 3D model of second FOA activated.

remains under inhibition of return due to the use of spatial memory even when its 2D location in the view frame and size has changed with respect to its last attended instance. Figures 4(b) shows attention on the second target whereas the ball is still under inhibition. The subfigure (c) shows the view in the spatial memory after aligning its sensors to the real world. Figures 4 (d) and (e) demonstrate views in the real world and the simulation framework after overt attention on the second target while the activated model of the attended object (the robot) can be seen in figure 4(f).

4 DISCUSSION

A conceptual framework of integrating a spatial memory with the vision procedures has been presented here and the feasibility of using a 3D simulator as a spatial memory is introduced. The area of integration of vision and spatial memory, their interaction, and cooperation needs to be explored further as there are many issues to be resolved. For example the physical system can gain error of localization and orientation over time due to inaccuracy in its sensors and wheel slippages that lead to synchronization problem between the real robot and its agent.

Using the spatial memory can increase the potentials of vision in 3D world and intelligence in autonomous decision making. Work needs to be done for handling further complexities in the scenario. For example, activation of the 3D models of objects will be more useful when positions of movable objects are not known in advance. Using the visual information from the camera, the robot could recognize an object and activate its whole model there. This can be helpful in navigation planning while roaming in known environments in which a bunch of known objects are moving around or located at arbitrary locations, for example 3D models of different types of vehicles could be used for intelligent autonomous drive on a known road map.

ACKNOWLEDGEMENTS

We gratefully acknowledge the funding of this work by the German Research Foundation (DFG) under grant Me 1289/12-1(AVRAM).

REFERENCES

Aivar, M. P., Hayhoe, M. M., Chizk, C. L., and Mruczek, R. E. B. (2005). Spatial memory and saccadic targeting in a natural task. *Journal of Vision*, 5:177–193.

Aziz, M. Z. and Mertsching, B. (2006). Color segmentation for a region-based attention model. In *Workshop Farbbildverarbeitung (FWS06)*, pages 74–83, Ilmenau - Germany.

Aziz, M. Z. and Mertsching, B. (2007). Color saliency and inhibition using static and dynamic scenes in region based visual attention. *Attention in Cognitive Systems, LNAI 4840*, pages 234–250.

Aziz, M. Z. and Mertsching, B. (2008a). Fast and robust generation of feature maps for region-based visual attention. *Transactions on Image Processing*, 17:633–644.

Aziz, M. Z. and Mertsching, B. (2008b). Visual search in static and dynamic scenes using fine-grain top-down visual attention. In *ICVS 08, LNCS 5008*, pages 3–12, Santorini - Greece. Springer.

Burns, D. and Osfield, R. (2004). Tutorial: Open scene graph. In *Proceedings Virtual Reality*, pages 265–265.

Cutzu, F. and Tsotsos, J. K. (2003). The selective tuning model of attention: Psychophysical evidence for a suppressive annulus around an attended item. *Vision Research*, pages 205–219.

Hoshino, E., Taya, F., and Mogi, K. (2008). Memory formation of object representation: Natural scenes. *R. Wang et al. (eds.), Advances in Cognitive Neurodynamics*, pages 457–462.

Kutter, O., Hilker, C., Simon, A., and Mertsching, B. (2008). Modeling and simulating mobile robots environments. In *3rd International Conference on Computer Graphics Theory and Applications (GRAPP 2008)*, Funchal - Portugal.

Mertsching, B., Aziz, M. Z., and Stemmer, R. (2005). Design of a simulation framework for evaluation of robot vision and manipulation algorithms. In *International Conference on System Simulation and Scientific Computing*, Beijing-China.

Moscovitch, M., Rosenbaum, R. S., Gilboa, A., Addis, D. R., Westmacott, R., Grady, C., McAndrews, M. P., Levine, B., Black, S., Winocur1, G., and Nadel, L. (2005). Functional neuroanatomy of remote episodic, semantic and spatial memory: a unified account based on multiple trace theory. *Journal of Anatomy*, pages 35–66.

Oman, C. M., Shebilske, W. L., Richards, J. T., Tubré, T. C., Bealli, A. C., and Natapoffi, A. (2000). Three dimensional spatial memory and learning in real and virtual environments. *Spatial Cognition and Computation*, 2:355–372.

Shelton, A. L. and Mcnamara, T. P. (2004). Spatial memory and perspective taking. *Memory & Cognition*, 32:416–426.

Smith, R. (last accessed March 2009). Open Dynamics Engine, Version 0.8. http://www.ode.org.

Treisman, A. M. and Gelade, G. (1980). A feature-integration theory of attention. *Congnitive Psychology*, 12:97–136.

Wolfe, J. M. and Horowitz, T. S. (2004). What attributes guide the deployment of visual attention and how do they do it? *Nature Reviews, Neuroscience*, 5:1–7.

ROBOT AUDITORY SYSTEM BASED ON CIRCULAR MICROPHONE ARRAY FOR HOME SERVICE ROBOTS

Keun-Chang Kwak
Dept. of Control, Instrumentation, and Robot Engineering, Chosun University
375 Seosuk-dong Dong-gu, Gwangju, 501-759, Korea
kwak@chosun.ac.kr

Keywords: Robot auditory system, Circular microphone array, Home service robots, Sound source localization, Speaker recognition, Speech enhancement.

Abstract: In this paper, we develop robot auditory system including speaker recognition, sound source localization, and speech enhancement based on circular microphone array for home service robots. These techniques are concerned with audio-based Human-Robot Interaction (HRI) that can naturally interact between human and robot through audio information obtained from microphone array and multi-channel sound board. The robot platform used in this study is wever, which is a network-based intelligent home service robot. The experimental results show the effectiveness of the presented audio-based HRI components from the constructed speaker and sound localization database.

1 INTRODUCTION

During the past few years, we have witnessed a rapid growth in the number and variety of applications of robots, ranging from conventional industrial robots to intelligent service robots. Conventional industrial robots perform jobs and simple tasks by following pre-programmed instructions for humans in factories. On the other hand, the main objective of the intelligent service robot is to adapt for the necessities of life as accessibility to human life increases. While industrial robots have been widely used in many manufacturing industries, intelligent service robots are still in elementary standard. Although the intelligent robots have been brought to public attention, the development of intelligent service robots remains as a matter to be researched further. Recently, there has been a renewal of interest in Human-Robot Interaction (HRI) for intelligent service robots. Among various HRI components, we especially focus on audio-based HRI including speech enhancement, speech recognition, speaker recognition, sound source localization, sound source separation, and gender/age classification. We shall deal with some of audio-based HRI components. The robot platform used in this paper is wever, which is a network-based intelligent home service robot equipped with multi-channel sound board and three low-cost condenser microphones. Finally, we shall show the performance of the developed techniques such as speaker recognition, sound localization, and speech enhancement among audio-based HRI components from the databases constructed in u-robot test bed.

2 AUDIO-BASED HRI

In this section, we present text-independent speaker recognition based on MFCC (Mel-Frequency Cepstral Coefficients) and GMM (Gaussian Mixture Model), sound source localization based on ESI (Excitation Source Information), and speech enhancement based on PBF (Phase-error Based Filter) with circular microphone array equipped with intelligent service robot.

2.1 Speaker Recognition

Firstly the EPD (Endpoint Detection) algorithm is performed to analyze speech signal obtained from speaker. Here the speech signal is detected by log energy and zero crossing. After detecting signal, the feature extraction step is performed by six stages to obtain MFCC. These stages consist of pre-emphasis, frame blocking, hamming window, FFT (Fast Fourier Transform), triangular bandpass filter, and cosine transform. For simplicity, we use 11 MFCC parameters except for the first order. In what follows, we construct GMM (Reynolds and Rose, 1995) frequently used in conjunction with text-independent speaker recognition to represent speaker's individual model in robot environments. Figure 1 shows the signal detected by log energy and zero crossing rate. Figure 2 shows the block diagram for feature extraction.

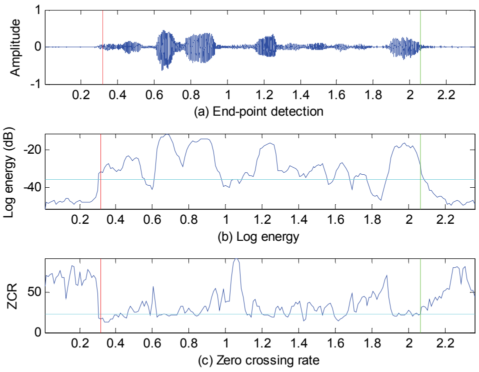


Figure 1: Endpoint detection.

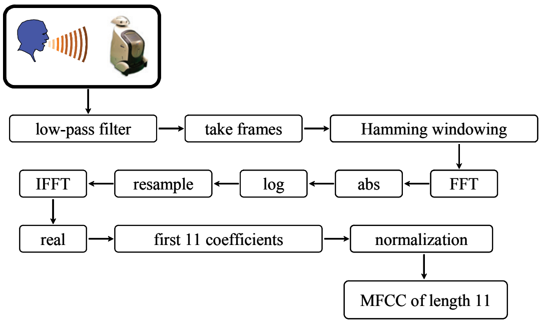


Figure 2: Block diagram for feature extraction.

2.2 Sound Source Localization

Sound source localization is performed by excitation source information and reliable angel estimation to determine the time-delay between each two microphones from speech source when robot's name is called. The time-delay based on excitation source information is comprised on two main stages. The first stage is to estimate time-delay from speech signals collected by three microphones. For this, the segmented signals by the endpoint detection should be detected. Here the speech signal of robot's name used in this study is wever. In order to perform time-delay estimation based on excitation source information, we firstly need to obtain linear prediction residual. This error includes the important information about excitation source during speech production. The linear prediction residual has a large value around the instants of glottal closure for voiced speech. However, these residuals should be transformed to derive critical information from short segments of linear prediction residual due to large fluctuations in amplitude. In the second stage, the values of linear prediction residual are transformed by computing the Hilbert envelop of linear prediction residual signal (Raykar and Yegnanarayana, 2005). Figure 3 shows linear prediction residual and Hilbert envelope. Figure 4 shows time-delay estimation from transformed signals obtained by Hilbert envelope.

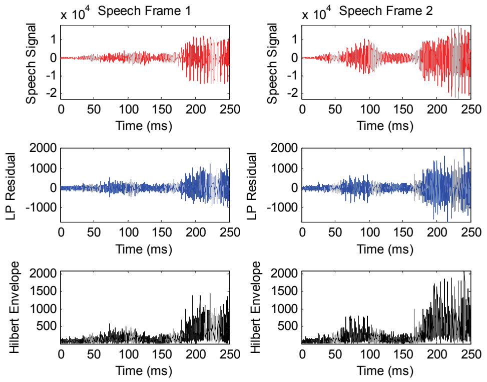


Figure 3: Linear prediction residual and Hilbert envelope.

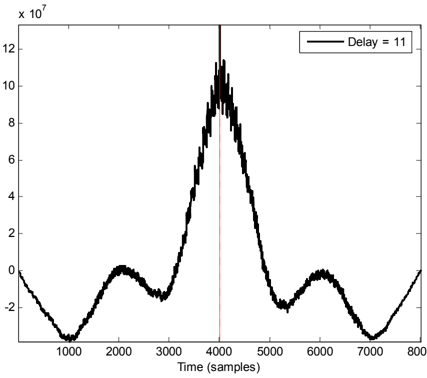


Figure 4: Time-delay estimation.

2.3 Speech Enhancement

This section present multi-microphone signal processing for speech recognition based on phase-error based filtering (Aarabi and Shi, 2004). This filtering performs time-frequency masking in the STFT (Short-time Fourier Transform) domain. For each pair of input frames, their phase-error spectrum is computed and used to modulate the amplitude spectrum. High error yields lower masking values. This has the effect of reducing time-alignment mismatch for each frequency bin, which is supposed to be related to reverberation and noise. Therefore, this method involves obtaining time-varying, or alternatively, time-frequency, phase-error filters based on prior knowledge regarding the time difference of arrival of the speech source of interest and the phase of the signals recorded by the microphones. Figure 5 shows the signals obtained from three microphones and enhanced signal, respectively.

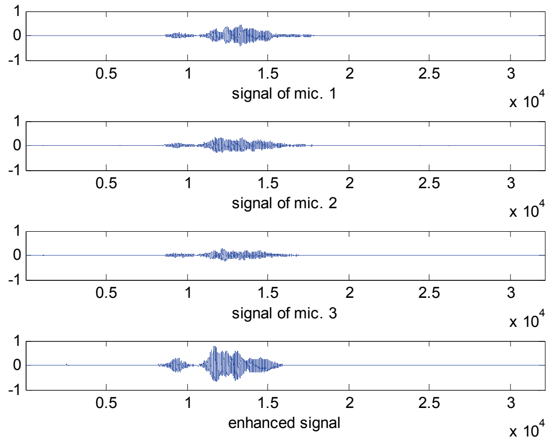


Figure 5: Speech enhancement by three microphones.

3 EXPERIMENTAL RESULTS

In this section, we use speaker database to evaluation the performance of the presented speaker recognition system (Kwak et al., 2007). Figure 6 shows "wever" equipped with three microphones and sound boards shown in Figure 7. The database is constructed by audio recording of 20 speakers. The data set consists of 30 sentences for each speaker and channel. For simplicity, we use only single microphone and 2200 sentences. The recording was done in u-robot test bed. The audio is stored as a mono, 16bit, 16kHz, and WAV file. The experimental results within 3 meter showed a good recognition performance of 94.5% recognition rate. However, the recognition performance at 4 and 5 meter showed 87.5% and 83%, respectively (see Figure 8).

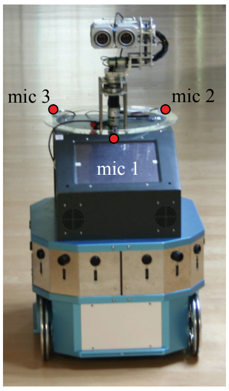


Figure 6: Robot platform-"wever".

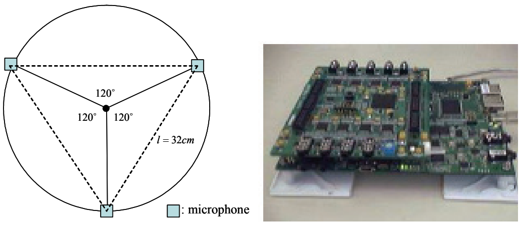


Figure 7: Arrangement of microphones and sound board.

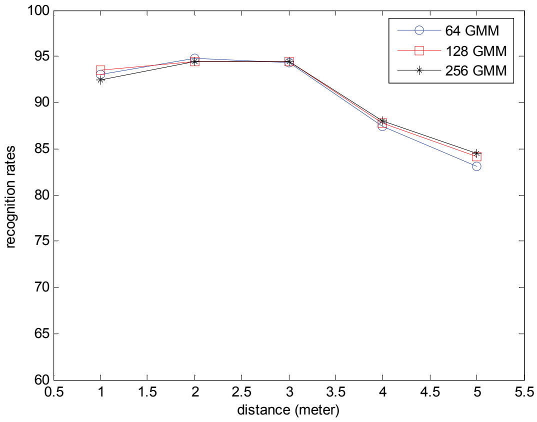


Figure 8: Recognition performance.

On the other hand, the localization success rate is considered as performance measure for sound localization. The localization success rate is computed by FOV (Field of View) of robot camera because sound localization is used with face detection when robot moves toward caller. The database used in this study was constructed in u-robot test bed environment that is similar with home environment to evaluate the sound localization algorithm (Kwak et al., 2008). The data set (M1 and M2) consists of 72 speeches at each meter from 1 meter to 3 meter. The localization success rate of the presented method is 92.3%. The presented method showed a better localization performance (about

20%) in comparison to that of TDOA and GCC-PHAT (see Figure 9 and 10).

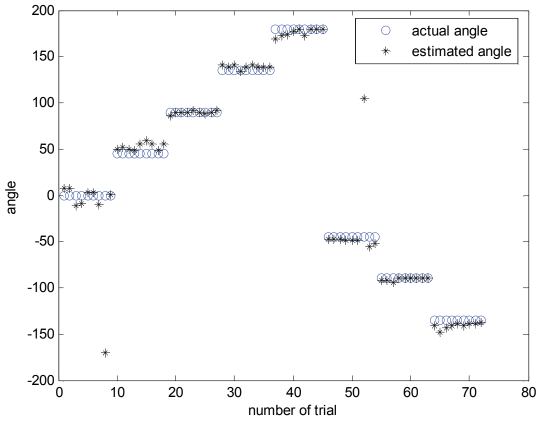


Figure 9: Actual angles and estimated angles (M1 set).

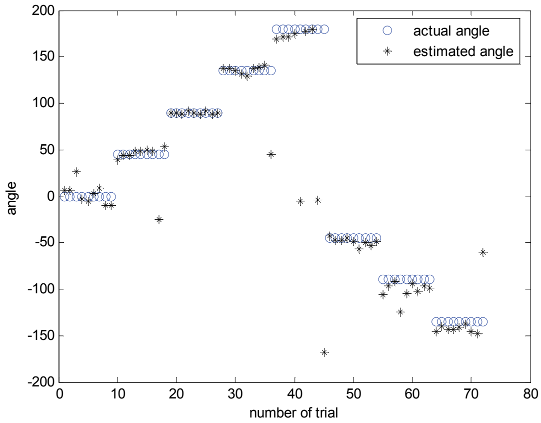


Figure 10: Actual angles and estimated angles (M2 set).

4 CONCLUSIONS

We have developed some of audio-based HRI components for intelligent home service robots. These components are composed of speaker recognition based on MFCC-GMM, sound source localization based on ESI and reliable angle estimation, and speech enhancement based on PBF. We have showed the usefulness and effectiveness of the developed techniques through the performance obtained from the constructed databases. On the basis of these components, we shall continuously develop other techniques such as sound source separation, gender/age classification, and fusion of information obtained from multi-microphones for humanlike robot auditory system.

REFERENCES

Reynolds, D. A., Rose, R. C., 1995. Robust text-independent speaker identification using Gaussian mixture speaker models. *IEEE Trans. on Speech and Audio Processing*, vol. 3, no. 1, pp. 72-83.

Kwak, K. C., Kim, H. J., Bae, K. S., Yoon, H. S., 2007. Speaker identification and verification for intelligent service robots. In *International Conference on Artificial Intelligence (ICAI2007)*, Las Vegas, May.

Raykar, V. C., Yegnanarayana, B., Prasanna, S. R. M., Duraiswami R., 2005. Speaker localization using excitation source information in speech. *IEEE Trans. on Acoustic, Speech, and Signal Processing*, vol. 13, no. 5, pp. 751-761.

Kwak, K. C., Kim, S. S., 2008. Sound source localization with the aid of excitation source information in home robot environments. *IEEE Trans. on Consumer Electronics*, vol. 54, no. 2, pp. 852-856.

Aarabi, P., Shi, G., 2004. Phase-based dual-microphone robust speech enhancement. *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 34, no. 4, pp. 1763-1773.

PATH PLANNING WITH MARKOVIAN PROCESSES

Istvan Szoke, Gheorghe Lazea, Levente Tamas, Mircea Popa and Andras Majdik
Technical University of Cluj-Napoca, Daicoviciu Street, Cluj-Napoca, Romania
{istvan.szoke, gheorghe.lazea, levente.tamas, mircea.popa, andras.majdik}@aut.utcluj.ro

Keywords: Path planning, Navigation algorithms, Mapping, Mobile robots, Markovian processes.

Abstract: This paper describes the path planning for the mobile robots, based on the Markov Decision Problems. The presented algorithms are developed for resolving problems with partially observable states. The algorithm is applied in an office environment and tested with a skid-steered robot. The created map combines two mapping theory, the topological respectively the metric method. The main goal of the robot is to reach from the home point to the door of the indoor environment using algorithms which are based on Markovian decisions.

1 INTRODUCTION

The first step in mobile robot navigation is to create or to use a map and to localize itself in it (Thrun, 2003). An autonomous agent has to have the following abilities: map learning or map creating, localization and path planning. The map representation can be metric or topological (Borenstein, 1996). In the case of the metric representation, the objects are replaced with precise coordinates, the disadvantage of this representation is that the precise distances can be very hard calculated, the map inaccuracies and the dead-reckoning errors are appearing often. The topological representation only considers places and the relations between them, its disadvantages would be the unreliable sensors which can not detect landmarks and perceptual aliasing. The second step in an agent's navigation process is the localization, which is strongly dependent to the map learning phase. This problem is common known as, Simultaneous localization and mapping (SLAM). SLAM is of one of the most important researched subfields of robotics (Fox, 2003). To plan a route to a goal location, the agent must be able to estimate its position. The most well known methods to do this, are the relative and absolute position measurements (Thrun, 2004). For the relative position measurements the most used methods are the odometry and inertial navigation, respectively for the absolute position estimation, the active beacons, artificial and natural landmark recognition and map-based positioning (Thrun, 2003). Path planning is defined as follows: is the art of deciding which route to take, based on and expressed in terms of the current internal representation of the terrain.

The definition of the path finding: the execution of this theoretical route, by translating the plan from the internal representation in terms of physical movement in the environment.

2 PATH PLANNING PROCESS

The effectiveness of a search can be measured in three ways. Does it bring a solution at all, it is a good solution (the one with a low path cost), and what is the search cost associated with the time and memory required to find a solution. The total cost of a search is defined as the sum of the path cost and the search cost. Route finding algorithms are used in a variety of applications, such as airline travel planning or routing in computer networks. In the case of the robot navigation, the agent can move in a continuous space with an infinite set of possible actions and states. In case of a circular robot which is moving on a flat surface, the space is two-dimensional, but in case of a robot that has arms and legs, the search space will be many-dimensional.

2.1 Markovian Processes

These kinds of processes integrate topological and metric representation as well, utilizing both action and sensor data in determining the robot position (Cassandra, 1996). Bayes rule is used to update the position distribution after each action and sensor reading (Koenig, 1996). In a Markov model actions occur in discrete time. To solve a Markov Decision Problem requires calculating an optimal policy in a stochastic environment with a transition model which satisfies the Markovian property.

2.2 Markov Decision Problem

The problem of calculating an optimal policy in a stochastic environment with a known transition model is called a **Markov decision problem** (MDP). It can be expressed, that a problem which has the **Markov property**, its transition model from any given state depend only on the state and not on previous history. Knowledge of the current state is all that is required in making a decision. A transition model is one which gives, for each state *s* and action *a* the resulting distribution of states if the action *a* was executed in *s*. A Markov decision process is a mathematical model of a discrete-time sequential decision problem (Littman, 2009). MDP is defined as a four tuple $\langle S, A, P, R\rangle$ (Regan, 2005), where S is the finite set of environment states, A is the set of actions, P is the set of action dependent transition probabilities, R is the reward function. $R: S \times A \rightarrow R$ is the expected reward for taking each action in each state. To maximize the expected reward over a sequence of decisions is the main goal of this problem. In an ideal case, the agent should take actions that maximize future rewards. In an MDP, the expected future reward is dependent only on the current state and action, so it must exist a stationary policy which will guarantee that the maximum expected rewards are received, if taken starting from the current state. The goal of any method to solve a MDP is to identify the optimal policy. The optimal policy is one that will maximize the expected reward starting from any state. If we would enumerate all of the possible policies for a state space, and then pick the one with the maximum expected value function, the method would be intractable, because the number of policies is exponential in the size of the state space. The traditional approach to solving sequential decision problems is dynamic programming. For applying this type of programming we need precise information about P, the transition probabilities and R, the reward function. Unfortunately, dynamic programming is computationally expensive in large state spaces.

2.2.1 Value Iteration Algorithm

This algorithm is used to calculate the optimal policy for the given environment. The main idea of the algorithm is to calculate the utility for each of the states, note with *U(state)*. These utility values are used for select an optimal action in each state.

$$U_{t+1}(i) \leftarrow R(i) + \max_a \sum M_{ij}^a \cdot U_t(j) \qquad (1)$$

Where *R* is called the reward function, M_{ij}^a is the transition model, the probability of reaching state *j* if action *a* is taken in state i, and *U* is the utility estimate. It’s an iterative algorithm, as $t->\infty$, the utility values will converge to stable values. Equation (1) is the basis for dynamic programming, which was developed by Richard Bellman (Russell, 1995). Using this type of programming sequential decision problems can be solved.

```
function VALUE_ITERATION (MDP)
   input: MDP with states S,
          transition model T,
          reward function R
    repeat
          U = U’
          for each state s do
          $U_{t+1}(i) \leftarrow R(i) + \max_a \sum M_{ij}^a \cdot U_t(j)$
    until close enough(U,U’)
    return U.
```

2.2.2 Policy Iteration Algorithm

After the utility values are calculated for all the states, the corresponding policy is calculated using the equation (2).

$$policy\,(i) = \arg\max_a \sum_j M_{ij}^a \cdot U(j) \qquad (2)$$

The basic idea of the algorithm is that by picking a policy, then calculate the utility of each state given that policy. The policy is updated after the new utility is inserted in the equation (3). The RMS (Root Mean Square) (Russell, 1995) method is used, to know how many iterations has to be done. The RMS error of the utility values are compared to the correct values.

```
function POLICY_ITERATION (MDP)
 input: MDP with states S,
        transition model T,
        reward function R
```

```
repeat
   U  =  Policy_Evaluation(π,U,MDP)
   unchanged?=true
for each state s do
   $policy(i) = \arg\max_{a} \sum_{j} M_{ij}^{a} \cdot U(j)$
   unchanged?=false
until unchanged?
return π.
```

2.3 Partially Observable Markov Decision Problem (POMDP)

This problem like the simple MDP is part of the sequential decision problems family. This can occur if the agent's environment is an inaccessible one, which means that, there is not enough information in order to determine the state or the associated transition probabilities, which means that the agent cannot directly observe that state. As a solution to this problem a new MDP has to be constructed, where the probability distribution plays the role of the state variable. It will end in a new state space, which will have real value probabilities, but they are infinite. For real-time applications even MDPs are hard to compute, and POMDP need approximations in order to obtain the optimal policy. Due to the fact that the agent doesn't have direct access to the current state, the POMDP algorithms need the whole history of the process, which means that it will lose it's Markovian property. This step is replaced by maintaining a probability distribution over all of the states, it gives us the same result as if we would keep the entire history. Between decisions the state can change, unlike in an MDP where state changes occur just due decisions. A variety of algorithms have been developed for solving POMDP. The Whitness algorithm (Littman, 1994) finds the solution using value iteration, it has been used with 16 states. In case of a more complex state space this algorithm would not be efficient. Another approach by (Littman, 1995), is a hybrid one, which is able to determine high quality policies with approximately 100 states. The realistic problems usually require thousands of states, so questions to this field remain open (Regan, 2005).

3 EXPERIMENTAL RESULTS

The described algorithm in this paper was tested using MobileSim simulator. The Pioneer AT robot's starting point is the Home Point situated in the first cell of the map. The robot has to reach the goal point which is the door of the room. The first step was to create the map of the room, where to robot will navigate. The size of the room is 7m wide and 8.5m long. The area of the room was divided in cells with area of $1m^2$. In Figure 1, the created map can be seen, with the most representative points, like the Home Point and the Goal Point. The other visible objects from the map, are obstacles like desks, shelves and supportive walls.

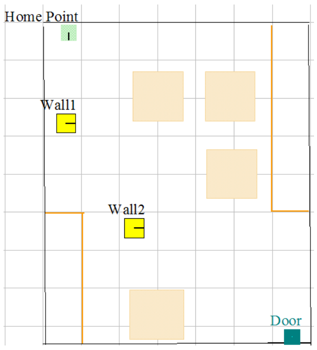


Figure 1: The map of the office environment.

The cells which represent an obstacle, their utility is zero, for the other cells, the iterative algorithm computes until the value between two consecutive utility value, for the same cell, is not more than 0.015. The initial value for a cell is -0.04, the utility value for the Goal Point is 1, and for a state which should be avoided, it has a value of -1. In a real world environment a state with utility -1, can be a whole in the ground or a state, from where the robot would get easily lost and use all of its power, due to this fact the agent must be able to reach his goal. The path shown in Figure 2, is the most shortest, what the robot can have. In order to avoid to get close to the state which has a utility of -1, the presented algorithm was applied to obtain another path. The scope of the path which will be obtained is to not have immediate neighbours to the cell which has utility of -1.

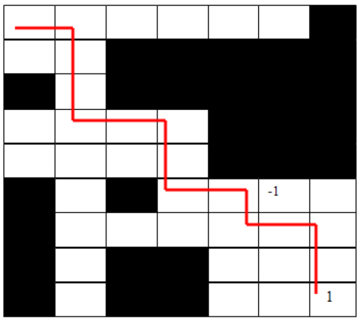
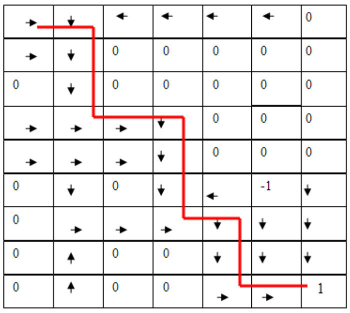


Figure 2: The shortest path.

After 20 iterations of the Value Iterative algorithm, the final utility values were obtained. In Figure 3, the policies are represented, which were obtained after applying the Policy Algorithm.



→	↓	←	←	←	←	0
→	↓	0	0	0	0	0
0	↓	0	0	0	0	0
→	→	→	↓	0	0	0
→	→	→	↓	0	0	0
0	↓	0	↓	←	-1	↓
0	→	→	→	↓	↓	↓
0	↑	0	0	↓	↓	↓
0	↑	0	0	→	→	1

Figure 3: The optimal policy values.

The cells with a higher utility value are more valuable for this algorithm than the states with a lower utility value. As shown in Figure 3, the optimal path has been found, the robot achieving to the door without getting close to the cell with utility -1, state which gets the lowest global reward.
In Figure 4, is represented the found path after testing with MobileSim.

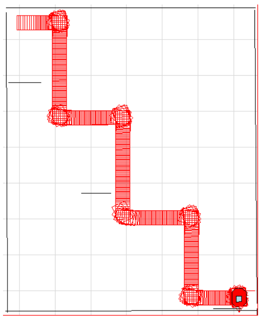


Figure 4: The found path using the presented algorithms.

The agent obtained an autonomous behaviour as presented in Figure 4 successfully avoids obstacles and the state with utility -1, it reach to the goal point which is the door of the office environment. As shown in Figure 2, the shortest path is not the path with the least cost (LaValle, 2006).

4 CONCLUSIONS

The main focus of this paper was to present the algorithms which are solving MDPs for path planning in mobile robots navigation. In a well known environment and with all the states observable, the agent is able to avoid obstacles and to navigate to the desired point.

5 FUTURE WORK

It is intended to create the map of the environment in a dynamic way. By the advance of the robot, the map would be created using the fusion of the data obtained by multiple sensors. The map will contain dynamic objects as well, human beings, which will have to be recognized and inserted in the map and to determine the projected positions for them.

REFERENCES

Cassandra, T., Kaelbing L., Kurien, J., 1996. *Acting under uncertainty: Discrete bayesian models for mobile-robot navigation*. IEEE/RSJ International Conference on Intelligent Robots and Systems.

Fox, D., Hightower J., 2003. Bayesian Filtering for Location Estimation. IEEE Pervasive Computing, vol. 2, no. 3, pp. 24-33, doi:10.1109/MPRV.2003.1228524.

LaValle, S., 2006. *Planning algorithms*. Cambridge University Press.

Littman, M,. 1994. *The Whitness algorithm: Solving partially observable Markov decision processes.* Tech. Rep. CS-94-04

Littman, M. L., 2009. *A tutorial on partially observable Markov decision processes.* Journal of Mathematical Psychology.

Regan, K., 2005. *The advisor-POMDP: A principled Approach to Ttrust through Reputation in Electronic Markets.* Proceedings of the third annual conference on privacy, security and trust, PST..

Thrun, S., 2003. *Learning Occupancy Grid Maps with Forward Sensor Models.* Autonomous Robots, v.15 n.2, p.111-127.

Thrun, S., Burgard, W., 2004. *A Probabilistic Approach to Concurrent Mapping and Localization for Mobile Robots.* Springer Netherlands.

AUTHOR INDEX

AUTHOR INDEX (CONT.)

AUTHOR INDEX (CONT.)

Proceedings of ICINCO 2009
6th International Conference on Informatics in Control, Automation and Robotics - Volume 2
ISBN: 978-989-674-000-9
http://www.icinco.org