ICINCO 2010

7th International Conference on
Informatics in Control, Automation and Robotics

Proceedings

Volume 2

Funchal, Madeira - Portugal · 15 - 18 June, 2010

Sponsored by:

INSTICC

Co-sponsored by:

IFAC

In Cooperation with:

aaai

Member Workflow Management Coalition WfMC

acm SIGART

ICINCO 2010

Proceedings of the
7th International Conference on
Informatics in Control, Automation and Robotics

Volume 2

Funchal, Madeira, Portugal

June 15 - 18, 2010

Co-Sponsored by
INSTICC – Institute for Systems and Technologies of Information, Control and Communication
IFAC – International Federation of Automatic Control

In Cooperation with
AAAI – Association for the Advancement of Artificial Intelligence
WfMC – Workflow Management Coalition
APCA – Associação Portuguesa de Controlo Automático
ACM SIGART – Association for Computing Machinery / Special Interest Group on Artificial Intelligence

Copyright © 2010 SciTePress – Science and Technology Publications
All rights reserved

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

Printed in Portugal

ISBN: 978-989-8425-01-0

Depósito Legal: 311149/10

http://www.icinco.org

icinco.secretariat@insticc.org

BRIEF CONTENTS

INVITED SPEAKERS

José Santos-Victor

Instituto Superior Técnico

Portugal

Alicia Casals

Institute for Bioengineering of Catalonia.IBEC and Universitat Politècnica de Catalunya.UPC

Spain

Bradley Nelson

Robotics and Intelligent Systems at ETH-Zürich

Switzerland

Wissama Khalil

Ecole Centrale de Nantes, IRCCyN

France

Oleg Gusikhin

Ford Research & Adv. Engineering

U.S.A.

John Hollerbach

University of Utah

U.S.A.

Organizing and Steering Committees

Conference Chair

Joaquim Filipe, Polytechnic Institute of Setúbal / INSTICC, Portugal

Program Co-chairs

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

Jean-Louis Ferrier, University of Angers, France

Proceedings Production

Patrícia Alves, INSTICC, Portugal

Helder Coelhas, INSTICC, Portugal

Vera Coelho, INSTICC, Portugal

Andreia Costa, INSTICC, Portugal

Patricia Duarte, INSTICC, Portugal

Bruno Encarnação, INSTICC, Portugal

Mauro Graça, INSTICC, Portugal

Raquel Martins, INSTICC, Portugal

Liliana Medina, INSTICC, Portugal

Carla Mota, INSTICC, Portugal

Vitor Pedrosa, INSTICC, Portugal

Filipa Rosa, INSTICC, Portugal

José Varela, INSTICC, Portugal

CD-ROM Production

Elton Mendes, INSTICC, Portugal

Pedro Varela, INSTICC, Portugal

Graphics Production and Webdesigner

Daniel Pereira, INSTICC, Portugal

Secretariat

Marina Carvalho, INSTICC, Portugal

Webmaster

Sérgio Brissos, INSTICC, Portugal

PROGRAM COMMITTEE

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

Alexande Poznyak, CINVESTAV-IPN, Mexico

Andrew Adamatzky, University of the West of England, U.K.

Eugenio Aguirre, University of Granada, Spain

Hesham Alfares, King Fahd University of Petroleum and Minerals, Saudi Arabia

Adel Al-Jumaily, University of Technology, Sydney, Australia

Francesco Amigoni, Politecnico di Milano, Italy

Peter Arato, Budapest University of Technology and Economics, Hungary

Alejandro Hernandez Arieta, University of Zurich, Switzerland

Marco Antonio Arteaga, Universidad Nacional Autonoma de Mexico, Mexico

Vijanth Sagayan Asirvadam, Universiti Teknologi PETRONAS, Malaysia

T. Asokan, Indian Institute of Technology Madras, India

Ruth Bars, Budapest University of Technology and Economics, Hungary

Adil Baykasoglu, University of Gaziantep, Turkey

Karsten Berns, University of Kaiserslautern, Germany

Mauro Birattari, IRIDIA-CoDE, Université Libre de Bruxelles, Belgium

Christian Blum, Universitat Politècnica de Catalunya, Spain

Patrick Boucher, Supélec, France

Bernard Brogliato, INRIA, France

Kevin Burn, University of Sunderland, U.K.

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

Dídac Busquets, Universitat de Girona, Spain

Javier Fernandez de Canete, University of Malaga, 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

Ratchatin Chanchareon, Chulalongkorn University, Thailand

Antonio Chella, Università di Palermo, Italy

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

Graziano Chesi, University of Hong Kong, China

Sung-Bae Cho, Yonsei University, Korea, Republic of

Ryszard S. Choras, University of Technology & Life Sciences, Poland

Carlos Coello Coello, CINVESTAV-IPN, Mexico

António Dourado Correia, University of Coimbra, Portugal

José Boaventura Cunha, University of Trás-os-montes and Alto Douro, Portugal

Mingcong Deng, Okayama University, Japan

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

Denis Dochain, Université Catholique de Louvain, Belgium

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

Venky Dubey, Bournemouth University, U.K.

Frederick Ducatelle, Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA), Switzerland

Ashish Dutta, Indian Institute of Technology Kanpur, India

Petr Ekel, Pontifical Catholic University of Minas Gerais, Brazil

Atilla Elci, Middle East Technical University, Turkey

Ali Eydgahi, University of Maryland Eastern Shore, U.S.A.

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

Paolo Fiorini, Università degli Studi di Verona, Italy

PROGRAM COMMITTEE (CONT.)

Georg Frey, Saarland University, Germany

Wai-Keung Fung, University of Manitoba, Canada

Dragan Gamberger, Rudjer Boskovic Institute, Croatia

Andrea Garulli, Universita' di Siena, Italy

Ryszard Gessing, Silesian University of Technology, Poland

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.

Kevin Guelton, University of Reims Champagne-Ardenne, France

Maki K. Habib, The American University in Cairo, Egypt

Wolfgang Halang, Fernuniversitaet, Germany

Onur Hamsici, Qualcomm, U.S.A.

John Harris, University of Florida, U.S.A.

Inman Harvey, University of Sussex, U.K.

Dominik Henrich, University of Bayreuth, Germany

Suranga Hettiarachchi, Indiana University Southeast, U.S.A.

Victor Hinostroza, University of Ciudad Juarez, Mexico

Wladyslaw Homenda, Warsaw University of Technology, Poland

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

Joris Hulstijn, Vrije Universiteit, Amsterdam, The Netherlands

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

Atsushi Imiya, IMIT Chiba University, Japan

Giovanni Indiveri, University of Salento, Italy

Mirjana Ivanovic, Faculty of Science, University of Novi Sad, Serbia

Sarangapani Jagannathan, Missouri University of Science and Technology, U.S.A.

Masoud Jamei, Simcyp Ltd, U.K.

Ping Jiang, The University of Bradford, U.K.

Graham Kendall, University of Nottingham, U.K.

DaeEun Kim, Yonsei University, Korea, Republic of

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

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

Gerhard K. Kraetzschmar, Bonn-Rhein-Sieg University of Applied Sciences, Germany

Mianowski Krzysztof, Politechnika Warszawska, Poland

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

Alexander Lanzon, University of Manchester, U.K.

Kathryn J. De Laurentis, University of South Florida, U.S.A.

Kauko Leiviskä, University of Oulu, Finland

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

Guoping Liu, University of Glamorgan, U.K.

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

José Tenreiro Machado, Institute of Engineering of Porto, Portugal

Anthony Maciejewski, Colorado State University, U.S.A.

Frederic Maire, Queensland University of Technology, Australia

Om Malik, University of Calgary, Canada

Hervé Marchand, INRIA, France

Philippe Martinet, Lasmea, France

Rene V. Mayorga, University of Regina, Canada

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

PROGRAM COMMITTEE (CONT.)

Prashant Mehta, University of Illinois at Urbana-Champaign, U.S.A.

Carlo Menon, Simon Fraser University, Canada

António Paulo Moreira, INESC Porto / FEUP, Portugal

Vladimir Mostyn, VSB - Technical University of Ostrava, Czech Republic

Rafael Muñoz-salinas, University of Cordoba, Spain

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

Andreas Nearchou, University of Patras, Greece

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

Sergiu Nedevschi, Technical University of Cluj-Napoca, Romania

Monica N. Nicolescu, University of Nevada, Reno, U.S.A.

Klas Nilsson, Lund University, Sweden

Urbano Nunes, University of Coimbra, Portugal

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

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

Igor Paromtchik, INRIA, France

Mario Pavone, University of Catania, Italy

Claudio de Persis, Sapienza University of Rome, Italy

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

Michael Piovoso, The Pennsylvania State University, U.S.A.

Raul Marin Prades, Jaume I University, Spain

José Ragot, Centre de Recherche en Automatique de Nancy, France

Jerzy Respondek, Silesian University of Technology, Poland

A. Fernando Ribeiro, Universidade do Minho, Portugal

Robert Richardson, University of Leeds, U.K.

Mihailo Ristic, Imperial College London, U.K.

Juha Röning, University of Oulu, Finland

Agostinho Rosa, IST, Portugal

Danilo De Rossi, University of Pisa, Italy

Fariba Sadri, Imperial College London, U.K.

Mehmet Sahinkaya, University of Bath, U.K.

Priti Srinivas Sajja, Sardar Patel University, India

Abdel-badeeh Salem, Ain Shams University, Egypt

Marcello Sanguineti, University of Genova, Italy

Jurek Sasiadek, Carleton University, Canada

Sergio M. Savaresi, Politecnico di Milano, Italy

Carla Seatzu, University of Cagliari, Italy

Michael Short, Teesside University, U.K.

Silvio Simani, University of Ferrara, Italy

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

Olivier Simonin, INRIA - LORIA, France

Joaquin Sitte, Queensland University of Technology, Australia

Adam Slowik, Koszalin University of Technology, Poland

Andrzej Sluzek, Nanyang Technological University/Nicolaus Copernicus University, Singapore

Safeeullah Soomro, Yanbu University College, Saudi Arabia

Stefano Squartini, Polytechnic University of Marche, Italy

Burkhard Stadlmann, University of Applied Sciences Wels, Austria

Chun-Yi Su, Concordia University, Canada

Ryszard Tadeusiewicz, AGH University of Science and Technology, Poland

Kazuya Takeda, Nagoya University, Japan

PROGRAM COMMITTEE (CONT.)

Daniel Thalmann, VR Lab EPFL, Switzerland

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

Avgoustos Tsinakos, University of Kavala Institute of Technology, Greece

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

Angel Valera, Universidad Politécnica de Valencia, Spain

Eloisa Vargiu, University of Cagliari, Italy

Annamaria R. Varkonyi-koczy, Obuda University, Hungary

Ramiro Velazquez, Universidad Panamericana, Mexico

Damir Vrancic, Jožef Stefan Institute, Slovenia

Bernardo Wagner, Leibniz Universität Hannover, Germany

Dianhui Wang, La Trobe University, Australia

James Whidborne, Cranfield University, U.K.

Sangchul Won, Pohang University of Science and Technology, Korea, Republic of

Qishi Wu, University of Memphis, U.S.A.

Marek Zaremba, Université du Québec (UQO), Canada

Janan Zaytoon, University of Reims Champagne Ardennes, France

Primo Zingaretti, Università Politecnica delle Marche, Italy

AUXILIARY REVIEWERS

Lounis Adouane, LASMEA, UMR CNRS 6602, France

Iman Awaad, Bonn-Rhein-Sieg University, Germany

Yao Chen, Institute of Systems Sciences, Chinese Academy of Sciences, China

Rohit Chintala, Texas A and M University, India

Jonathan Courbon, LASMEA, France

Kun Deng, University of Illinois at Urbana Champaign, U.S.A.

Anibal Ferreira, University of Porto, Portugal

Leonardo Fischer, Universidade Federal do Rio Grande do Sul - UFRGS, Brazil

Fernando Fontes, Faculty of Engineering, University of Porto, Portugal

Ronny Hartanto, Bonn-Rhein-Sieg University of Applied Sciences, Germany

Laurent Harter, -, France

Nico Hochgeschwender, ESG GmbH, Germany

Ana Lopes, Institute of Systems and Robotics - University of Coimbra, Portugal

Jan Paulus, Bonn-Rhein-Sieg University oAS, Germany

Cristiano Presmebida, Institute of System and Robotics, Portugal

Michael Reckhaus, Bonn-Rhein-Sieg University, Germany

Lynda Seddiki, Paris 8 University, France

Renato Silveira, Universidade Federal do Rio Grande do Sul (UFRGS), Brazil

Yu Sun, University of Illinois, Urbana-Champaign, U.S.A.

C. Renato Vázquez, Universidad de Zaragoza, Spain

Huibing Yin, University of Illinois at Urbana-champaign, U.S.A.

SELECTED PAPERS BOOK

A number of selected papers presented at ICINCO 2010 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 2010 Program Committee members.

Foreword

This book contains the proceedings of the 7th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2010) which was sponsored by the Institute for Systems and Technologies of Information, Control and Communication (INSTICC) and held in Funchal, Madeira - Portugal. ICINCO 2010 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 Workflow Management Coalition (WfMC), the Portuguese Association for Automatic Control (APCA) and the Association for Computing Machinery (ACM SIGART).

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 six plenary keynote lectures, given by internationally recognized researchers, namely - José Santos-Victor (Instituto Superior Técnico, Portugal), Alícia Casals (Institute for Bioengineering of Catalonia.IBEC and Universitat Politècnica de Catalunya.UPC, Spain), Bradley Nelson (Institute of Robotics and Intelligent Systems at ETH-Zürich, Switzerland), Wisama Khalil (Ecole Centrale de Nantes, IRCCyN, France), Oleg Gusikhin (Ford Research & Adv. Engineering, U.S.A.) and John Hollerbach (University of Utah, U.S.A.).

The meeting is complemented with one satellite workshop, the International Workshop on Artificial Neural Networks and Intelligent Information Processing (ANNIIP), and one Special Session on Intelligent Vehicle Controls & Intelligent Transportation Systems (IVC & ITS).

ICINCO received 320 paper submissions, not including those of the workshop or the special session, from 57 countries, in all continents. To evaluate each submission, a double blind paper review was performed by the Program Committee. Finally, only 142 papers are published in these proceedings and presented at the conference. Of these, 94 papers were selected for oral presentation (27 full papers and 67 short papers) and 48 papers were selected for poster presentation. The full paper acceptance ratio was 8%, and the oral acceptance ratio (including full papers and short papers) was 29%. 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 2010.

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 Workshop Social Event & Banquet in the evening of June 17 (Thursday).

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 INSTICC team; 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 2011.

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

BIOINSPIRED ROBOTICS AND VISION WITH HUMANOID ROBOTS

José Santos-Victor
Instituto Superior Técnico, Lisboa, Portugal

Abstract: In this talk, I will describe recent results on exploring recent results from neurophysiology and developmental psychology for the design of humanoid robot technologies. The outcome of this research is twofold: (i) using biology as an inspiration for more flexible and sophisticated robotic technologies and (ii) contribute to the understanding of human cognition by developing biologically plausible (embodied) models and systems.

One application area is the domain of video surveillance and human activity recognition. We will see how recent findings in neurophysiology (the discovery of the mirror neurons) suggest that both action understanding and execution are performed by the same brain circuitry. This might explain how humans can so easily (apparently) understand the actions of other individuals, which constitutes the building block of non-verbal communication first and then, language acquisition and social learning.

The second aspect to be addressed is the use of development as a methodological approach for building complex humanoid robots. This line of research is inspired after the human cognitive and motor development, a pathway that allows newborns to progressively acquire new skills and develop new learning strategies. In engineering terms, this may be a way not only to structure the sensed data but also to master the complexity of the interaction with the physical world with a sophisticated body (sensing and actuation).

During the talk, I will provide examples with several humanoid platforms used for this research: Baltazar is a humanoid torso we developed to study sensorimotor coordination and cognition; the latest results are implemented in the iCub humanoid robot, for which we designed the head, face and body covers as well as the attention and affordance learning system.

BRIEF BIOGRAPHY

José Santos-Victor received the PhD degree in Electrical and Computer Engineering in 1995 from Instituto Superior Técnico (IST - Lisbon, Portugal), in the area of Computer Vision and Robotics. He is an Associate Professor with "Aggregation" at the Department of Electrical and Computer Engineering of IST and a researcher of the Institute of Systems and Robotics (ISR) and heads the Computer and Robot Vision Lab - VisLab.

He is the scientific responsible for the participation of IST/ISR in various European and National research projects in the areas of Computer Vision and Robotics. His research interests are in the areas of Computer and Robot Vision, particularly in the relationship between visual perception and the control of action, biologically inspired vision and robotics, cognitive vision and visual controlled (land, air and underwater) mobile robots.

Prof. Santos-Victor was an Associated Editor of the IEEE Transactions on Robotics and the Journal of Robotics and Autonomous Systems.

HUMAN
Robot Cooperation Techniques in Surgery

Alícia Casals
Institute for Bioengineering of Catalonia (IBEC), Universitat Politècnica de Catalunya (UPC), Barcelona, Spain
alicia.casals@upc.edu

Keywords: Medical Robotics, Human Robot Interaction, Human Machine Interfaces, Surgical Robots.

Abstract: The growth of robotics in the surgical field is consequence of the progress in all its related areas, as: perception, instrumentation, actuators, materials, computers, and so. However, the lack of intelligence of current robots makes teleoperation an essential means for robotizing the Operating Room (OR), helping in the improvement of surgical procedures and making the best of the human-robot couple, as it already happens in other robotic application fields. The assistance a teleoperated system can provide is the result of the control strategies that can combine the high performance of computers with the surgeon knowledge, expertise and will. In this lecture, an overview of teleoperation techniques and operating modes suitable in the OR is presented, considering different cooperation levels. A special emphasis will be put on the selection of the most adequate interfaces currently available, able to operate in such quite special environments.

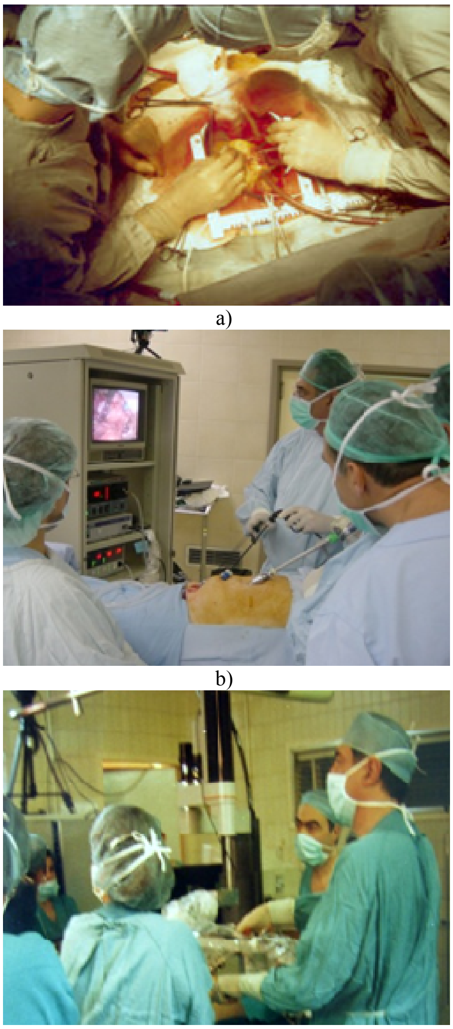
1 INTRODUCTION

Technological evolution has continuously been introducing new equipments and changes in the Operating Room. Technology does not only affect real surgical interventions, but it also has a bear on all diagnosis and planning strategies, according to the diagnosed pathology. The history of surgery has suffered a continuous evolution through which three significant phases can be identified. They can be summarized in fig.1. From the practice of open surgery, fig. 1 a), in which surgeons get in touch directly with the patient organs or corresponding body parts, as purely manual actuation, the advent of new instruments and visualization techniques opened the era of minimally invasive surgery. Instruments with long handles allow entering the body through natural holes and small incisions over the patient fig. 1.b). This image shows the common scenario in laparoscopic surgery. At this stage, surgeons rely on instruments and specific equipment to perform surgery. The era of surgical robotics emerges as these instruments acquire new performances, or others appear in the scene, fig. 1.c).

New surgical procedures, not conceivable several decades ago, are more complex and require much higher performances. To face the challenges this kind of surgery relies on robots cooperating with humans, so as to extract the best of both of them. Humans provide intelligence and decision making while robots contribute with their precision, computing capabilities and no tiredness. In this context, with human and machines sharing the working scene, and the task itself, more powerful interfaces and interaction means become necessary.

Both, cooperation and interface requirements will depend on the typology of surgery. Speaking in terms of robotics technology and considering that technological needs vary enormously with the kind of surgery, surgery can be classified in: microsurgery, neurosurgery, intracavity and orthopedic, and percutaneous and transcutaneous interventions. As significant distinction among them, some characteristics, as the kind of tissue (hard or soft) or the body parts undergoing surgery, are to be considered.

Since hard tissues are able to maintain its shape, when they can be immobilized some techniques applied in industry can be exported to surgery, otherwise, a tracking system to dynamically determine the changing reference frames is required. Soft tissues present the problem of deformability, making robot operation more complex. In this case, teleoperation is an alternative solution.



a)

b)

c)

Figure 1: Evolution of surgical procedures. a) open surgery, b) minimally invasive surgery, c) robot assisted surgery.

2 HUMAN ROBOT COOPERATION MODES

Human robot cooperation is implemented by means of teleoperation, therefore, a master device in the surgeon side controls a slave arm, a teleoperated robot, patient side. In between, a computer implements the required assistive functions that enhance human capabilities, resulting in a "super surgeon". Such assistive functions can be a change of scale, defining constrains within the working space, tremor reduction and movement compensation (breathing or heart beating) and so. The surgeon can be located in a close position, or in any other location, a few meters or some kilometers away. Fig. 2 shows a schema of such system.

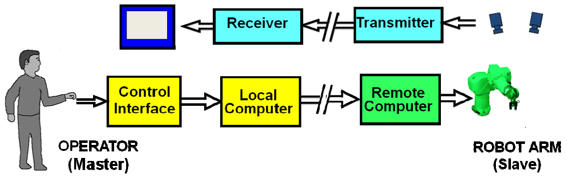


Figure 2: Schema of a teleoperation system.

However, the surgeon could also be in close contact with the robot, which guides and supervises his or her actions. In this cooperative mode, comanipulation, the surgeon hand and the robot end effector move simultaneously holding the surgical instrument. Working in these conditions, a change of scale or movement compensation is not an issue, but assisted teleoperation allows establishing constraints, virtual fixtures, operating over reference frames either fixed or floating over the patient anatomy. Comanipulation also allows directly perceiving both, images and the operating environment, what is especially useful in orthopedic surgery. The definition of virtual fixtures during the surgical planning facilitates a safer operation reducing the surgeon stress in critical interventions. Working with this configuration the robot itself behaves as a *haptic* device, Fig. 3.

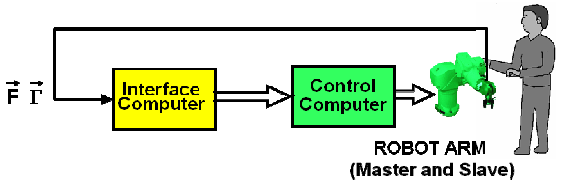


Figure 3: Schema of a comanipulation system.

The level of cooperation can also vary with the typology of the surgical interventions since the level of preplanning and programming varies depending on the predictability of the intervention. Three levels can be considered: manual guidance, supervised guidance and autonomous control.

The role of the interface in such cooperation systems is crucial as the surgeon cannot pay much attention to the robotic system, but to the patient and the own surgical procedure. A schema of the characteristics an interface should provide is shown in fig. 4.

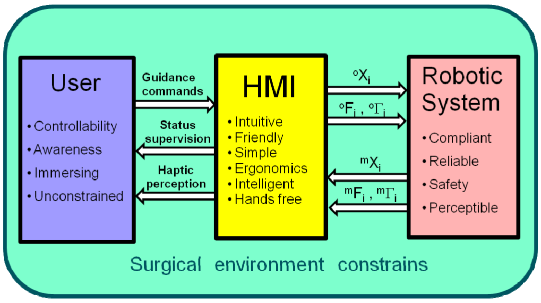


Figure 4: Schema of the characteristics of a teleoperation interface.

3 ROBOT ARCHITECTURES

Different surgical specialties work along different scale ranges and with different working requirements. Considering the variety of working conditions and from the robotics point of view, surgical procedures can be classified as follows: microsurgery, neurosurgery, transcutaneous, percutaneous, intracavities interventions and orthopedics. In what follows the main characteristics of each of them are described.

3.1 Microsurgery

Most interventions requiring microsurgery performances are related to sewing nerves in transplants or to ophthalmology. In this specialty, one or more high precision and high accessibility 6 Degrees of Freedom (DoF) arms are necessary. Additional DoF might be required to increase accessibility. As teleoperation assistance functions, a change of scale between the master and the slave increases the achievable precision as the application requires. Compared to former manipulators or holding devices used in classical manual practice, robots substitute them with advantage

3.2 Neurosurgery

Interventions in the skull also require high precision. However, its accessibility requirements are no so demanding since insertions are applied through incisions done just over the target area. In this case, movement compensation or floating reference frames are not needed as the skull can be fixed, with stereotaxis devices.

3.3 Transcutaneous

Some interventions can be performed through the skin and soft tissues using radiation focused over the target area. This radiation can be of different types: RX, Gama, or High Intensity focused US (HIFU). In such minimally invasive technique, 5DoF are enough to focus the therapeutic beam over a point in a 3D space, with any orientation.

3.4 Percutaneous

Relatively simple interventions as biopsy, aspiration, ablations or releasing therapeutic payload, are more and more used due to the few invasiveness of the technique. This technique implies inserting needles with high precision, to produce the advance and drilling movements when the needle is manually oriented into the insertion point. 5 DoF are necessary to place and orient the needle with a robot.

Taking advantage of this minimally invasive surgery or intervention, deflection and guided probes can be used to reach areas not easily reachable. Operating through natural ways, as the arteries, this technique is being used successfully to treat brain aneurisms.

3.5 Intracavity Interventions

When the intervention cannot be carried out by means of needles and more versatile instruments are needed with two, three or four DoF, endoscopic techniques are required. There are two endoscopic techniques, one based on the use of natural orifices (NOTES) and the second based on small incisions to access the abdominal and thoracic cavity (laparoscopy) or the joints between bones (arthroscopy).

These techniques were introduced in the seventies operating the instruments manually. At present, these instruments can be guided by teleoperated robots, so as they can take advantage of assisted teleoperation techniques. These robots should be multiarm (2, 3 or 4), each with several DoF, not only for tool positioning but also to increase accessibility and to make the pose of each arm compatible with the patient in the operating table.

NOTES are used in intra vaginal interventions, ear, laparoscopy for abdominal, prostate, heart, gynecology or arthroscopy, knee specially.

3.6 Orthopedics

Orthopedic surgery uses as end effectors drilling, cutting or milling tools to operate over bone tissues. These techniques are oriented to bone repair either with prosthesis implants, subjecting or immobilizing boards or to reconstruct bones with grafts after oncologic surgery, for example.

For these procedures CAD/CAM techniques can be used, with similar methods than those used in industry. Reference frames registration between anatomical elements in the operating table and the CAD model previously obtained from CT images are used to make task planning possible.

4 SENSOR REQUIREMENTS

Based on these different scenarios, robotics requires different kind of sensors to be able to implement the required control strategies. Two kinds of sensors are needed: 3D geometrical positioning sensors (navigators) and physical interaction characteristics (force and torque sensors).

Positioning anatomical parts in the 3D space is not simple, especially when dealing with not immobilized rigid elements, soft, deformable or rhythmically moving tissues. The success of surgical robots rely on their capability for adequately sensing positions either using physical contact sensors (optical or magnetic techniques) or remote sensing, specially vision. Current limitations of computer vision strongly condition its advances.

Apart from being able to control the robot, not only geometrical strategies are necessary to generate trajectories, but additional force control techniques are required to avoid injures, as for instance, necrosis. On the other hand, force sensors provide the information required to generate *haptic* information to be feedback to the surgeon. .

5 INTERFACES REQUIREMENTS

The requirements of an interface for surgical applications does not imply uniquely the interpretation of human will to control the teleoperated robot, but also to provide some feedback of the task going on to the surgeon. Thus, an interface constitutes a complete system, fig. 5, consisting of master devices adequate for every specific kind of surgery, actuators to feedback information to the surgeon, monitoring devices, and the computing power to process the information coming both, from the teleoperated system to provide the adequate information and from the human operator to provide the adequate control orders.

The schema of fig.5 shows that an interface can be a complete system that in some environments should provide certain intelligence level and, as indicated in the schema, even generate synthetic information to improve human operator's perception.

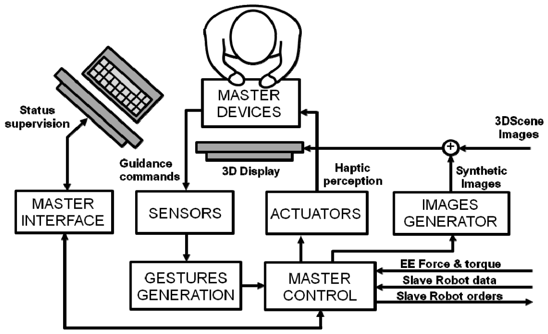


Figure 5: Schema of the master station in a teleoperated system.

Besides considering the best kind of master devices, either classical manual devices or hand free systems, the information coming from the robot side are key to achieve an efficient and smooth interaction. Feedback information can be either visual: direct images, augmented images or virtual, besides other kind of numerical and graphic data; or haptic, based on the force data measured in the robot working environment.

6 CONCLUSIONS

Interfaces are key components in teleoperated systems, or in robotic systems that work in close cooperation with humans. In the field of surgery, the surgeon faces the problem of dealing with complex systems while they are performing their own job, surgery. On the other side, their specialty is far from informatics and mechatronic systems. Thus, the design of a human robot interface in such fields should consider the three parts that compose the working environment, as shown in fig. 4, the user, the interface and the robotic system. While the surgeon has to pay complete attention to the intervention itself, the interface should provide the means of reacting to the humans' will, to interpret their needs, and to supply any kind of information that can help them to take decisions.

Teleoperation and its interface with the human operator can take different configurations according to the application needs, considering both, the distance from the master to the slave and the typology of the application. In each case, the availability of the required teleoperation assistance functions to improve human and system performances is essential. Together with such assistance, the information fed back to the user, visual, haptic or even sound, can be intelligently processed to constitute a significant help for the whole process.

BRIEF BIOGRAPHY

Alicia Casals is professor at the Technical University of Catalonia (UPC), in the Automatic Control and Computer Engineering Department. She is currently leading the research group on Robotics and Medical imaging Program of the Institute for Biomedical Engineering of Catalonia, and is member of the research group GRINS: Intelligent Robotics and Systems at UPC. The research is oriented to improve human robot interaction through multimodal perception, focused mainly in the area of medical robotics. In this field she is working both in rehabilitation, assistance and surgical applications. Her background is in Electrical and Electronic Engineering and PhD in Computer Vision. From 2001 to 2008 she was the coordinator of the Education and Training key area within Euron, the Network of Excellence: European Robotics Network, and RAS Vice President for Membership in the period 2008-2009. From the developed research projects she won different awards, Award to a social invention (Mundo Electrónico), International Award *Barcelona'92* (Barcelona City_Hall), *Ciutat de Barcelona* Award 1998 (Barcelona City_Hall), and *Narcis Monturiol* Medal from the Catalan Government as recognition of the research trajectory 1999. From 2007 Prof. Casals is member of the *Institut d'Estudis Catalans*, the Academy of Catalonia.

MAKING MICROROBOTS MOVE

Bradley J. Nelson
Institute of Robotics and Intelligent Systems, ETH Zurich, Zurich, Switzerland

Keywords: Nanorobotics, Microrobotics, Nanocoils, Magnetic actuation.

Abstract: Our group has recently demonstrated three distinct types of microrobots of progressively smaller size that are wirelessly powered and controlled by magnetic fields. For larger scale microrobots, from 1mm to 500 µm, we microassemble three dimensional devices that precisely respond to torques and forces generated by magnetic fields and field gradients. In the 500 µm to 200 µm range, we have developed a process for microfabricating robots that harvest magnetic energy from an oscillating field using a resonance technique. At even smaller scales, down to micron dimensions, we have developed microrobots we call Artificial Bacterial Flagella (ABF) that are of a similar size and shape as natural bacterial flagella, and that swim using a similar low Reynolds number helical swimming strategy. ABF are made from a thin-film self-scrolling process. In this paper I describe why we want to do this, how each microrobot works, as well as the benefits of each strategy.

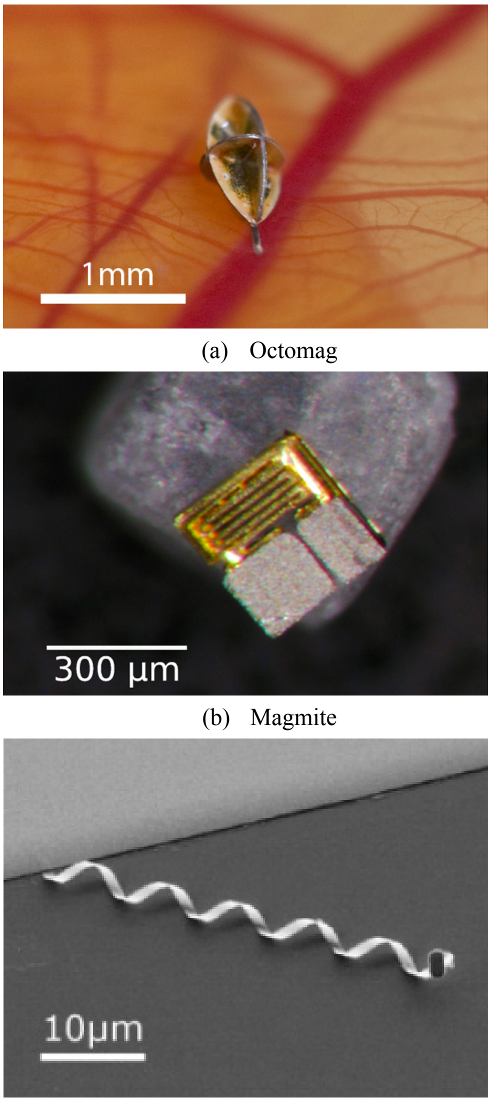
1 INTRODUCTION

Micro and nanorobotics have the potential to dramatically change many aspects of medicine by navigating bodily fluids to perform targeted diagnosis and therapy and by manipulating cells and molecules. In the past few years, we have developed three new approaches to wirelessly controlling microscale structures with high precision over long distances in liquid environments (Figure 1). Because the distance from which these structures can be controlled is relatively large, the structures can not only be used as tools for manipulating other micro and nanoscale structures, similar to particle trapping techniques, but can also serve as vehicles for targeted delivery to locations deep within the human body. The microrobots we have developed are non-spherical. Therefore, both their position and orientation can be precisely controlled, removing another limitation of particle trapping. Unprecedented control in multiple degrees of freedom has been achieved with field strengths as low as 1 mT.

2 MEDICAL MICROROBOTS

Minimally invasive medical techniques are linked with a variety of patient oriented benefits ranging from reduction of recovery time, medical complications, infection risks, and post-operative pain, to lower hospitalization costs, shorter hospital stays, and increased quality of care [1-4]. Microrobotic devices have the potential for improved accessibility compared to current clinical tools, and medical tasks performed by them can even become practically noninvasive. They will perform tasks that are either difficult or impossible with current methods. Rather than acting as autonomous agents that navigate the body diagnosing and solving problems, microrobots will more likely act as new technical tools for clinicians, continuing to capitalize on the clinicians cognitive skill, which is their greatest asset.

In recent years there has been significant progress on robot-assisted colonoscopy and on wireless miniature robots for use in the GI tract (Kazzim et al., 2006). Motivated by capsule endoscopes that are already in clinical use, a number of technologies have been explored to expand the capabilities of these devices, ranging from wireless GI pressure monitoring systems and lab-on-a-chip devices equipped with pH and temperature sensors (Johannessen et al., 2006) to the addition of legs and other mechanisms for controlled locomotion (Menciassi et al., 2007). The size of these devices approaches a few centimeters, capitalizing on the relatively large size of the GI tract. By further reducing device size and creating microrobots with a



(a) Octomag

(b) Magmite

(c) ABF

Figure 1: Three different types of microrobots at varying orders of magnitude. a) The Octomag robot (shown puncturing a small vein) is controlled using magnetic fields and field gradients. b) The Magmite (sitting on a grain of salt) is powered by oscillating magnetic fields that excite a spring mass system to resonance that harvests the impact energy. c) Artificial Bacterial Flagella (ABF) are propelled through fluid by rotating a magnetic field to generate torques on the magnetic metal head of the device.

maximum dimension of only a few millimeters or less, additional locations in the human body becomavailable for wireless intervention. Natural pathways such as the circulatory system, the urinary system, and the central nervous system become available, enabling intervention with minimal trauma.

As we downscale robots to submillimeter dimensions, the relative importance of physical effects changes (Wautelet, 2001). As device size is reduced, surface effects and fluid viscosity dominate over inertia and other volumetric effects, and power storage becomes a key issue. Furthermore, microrobots, like microorganisms, swim in a low-Reynolds-number regime, requiring swimming methods that differ from macroscale swimmers (Purcell, 1977). This places strong constraints on the development of medical microrobots. In traditional robotics, it is often easy to compartmentalize aspects of robot design such as kinematics, power, and control. In the design of wireless microrobots, fabrication is fundamentally limited by scaling issues, and power and control are often inextricably linked. Engineers must give up intuition gained from observing and designing in the macroscale physical world, and instead rely on analysis and simulation to explore microrobot design. Even then, only experimental results will demonstrate the efficacy of a given microrobot strategy, as the world experienced by the microrobot may be quite difficult to accurately model.

3 OCTOMAG MICROROBOTS

The Octomag microrobot (Yesin et al., 2006), shown in Figure 1, was the first microrobot we developed and is primarily intended for ophthalmic surgery. An external magnetic field acts to align the robot along the long (“easy”) axis, and a field gradient is generated to pull or push the microrobot. The winged shape acts to reduce the side-ways drift of the microrobot by increasing the fluid drag along the axes perpendicular to the long axis (Abbott et al., 2007). The relatively large size of the device is compatible with using gradient fields for propulsion at distances suitable for use within the body and is targeted at controlling the device in a viscous medium.

The robot is a three-dimensional structure built by microassembling individual parts, which allows for the combination of incompatible materials and processes for the integration of MEMS based sensors and actuators. The principle advantage of the hybrid design is that the individual parts of the assembly can be produced with standard MEMS manufacturing processes that create planar geometries. In this way, different subsystems of the robot can be manufactured using the most suitable process for the purpose. Robot parts have been made with electroplated nickel, single crystal silicon, polymer, and laser cut steel.

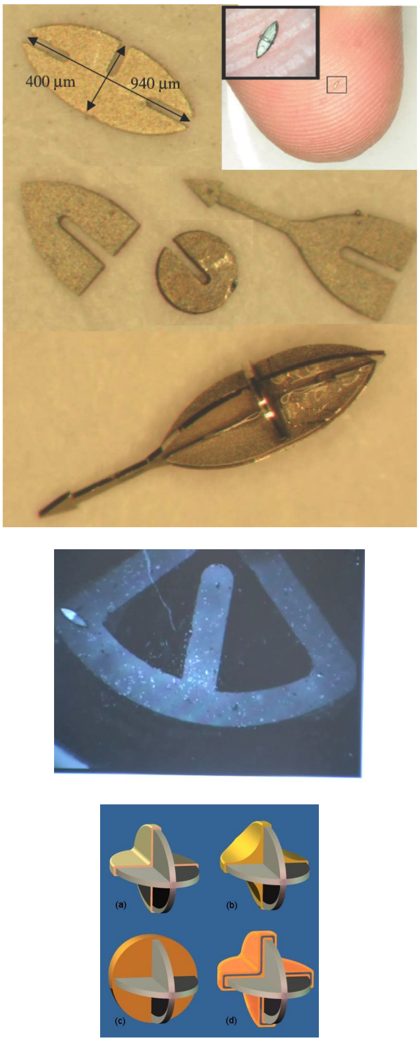


Figure 2: (a) Octomag microrobots are microassembled from 50µm thick electroplated nickel. (b) The sub-mm devices can be precisely controlled to navigate through tiny mazes, and (c) can be coated to ensure biocompatibility and for transport of cargo, such as drugs to be unloaded by diffusion (Yesin et al., 2006).

4 MAGMITES

With decreasing size, gradient propulsion becomes infeasible due to the force generated being related to the magnetic medium's volume, which decreases rapidly with size. To overcome this limitation, we have developed a second propulsion mechanism that harnesses the interactive forces between small magnetic bodies in a uniform magnetic field to drive a spring mechanism to resonance(Vollmers et al., 2008). This energy is then rectified to move the robot through its environment.

The resonant nature of the actuator enables the device to move with fields below 2 mT which is roughly 50x that of the Earth's magnetic field. This locomotion mechanism has been demonstrated on both structured and unstructured surfaces and is controllable enough to repeatedly and precisely follow trajectories. The frequency selectivity of the spring mass resonating structure allows multiple robotic agents to be used on the same substrate to perform tasks. Although this propulsion method was initially designed for operation in air, it has demonstrated its ability to perform in aqueous environments and manipulate glass microspheres on the order of 50 µm.

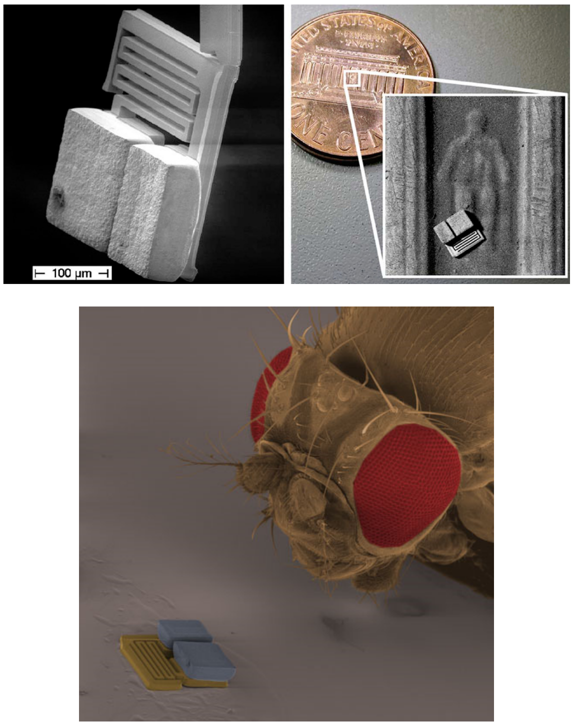


Figure 3: Magmite robots consist of two Ni masses separated by a gold spring. The robot shown measures 300µm square, 70µm thick, and is dwarfed by Drosophila melanogaster (Vollmers et al., 2008).

5 ARTIFICIAL BACTERIAL FLAGELLA

As sizes decrease further, the resonant frequencies of the mechanical structures required for the resonant magnetic actuator increase to tens of kHz and become difficult to generate at sufficient strength. At this scale, torque on the magnetic bodies becomes one of the predominant forces that can be generated.

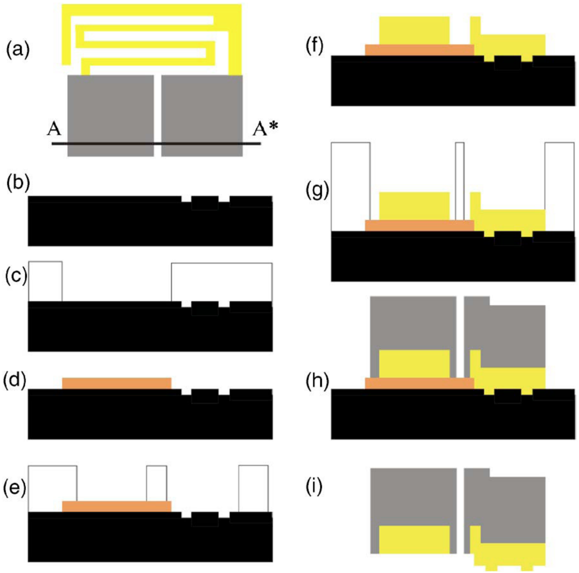


Figure 4: Fabrication sequence with cross section shown along line A-A* (a). Holes for dimples (b) are etched in a wafer before a Ti/Cu adhesion/seed layer is evaporated onto the surface. Photoresist is applied (c) to define thick electroplated copper islands (d). The springs and frame are defined (e) and plated (f) before a final layer of photoresist (g) defines the nickel bodies (h). The device is released from the wafer by etching the sacrificial copper layer (i).

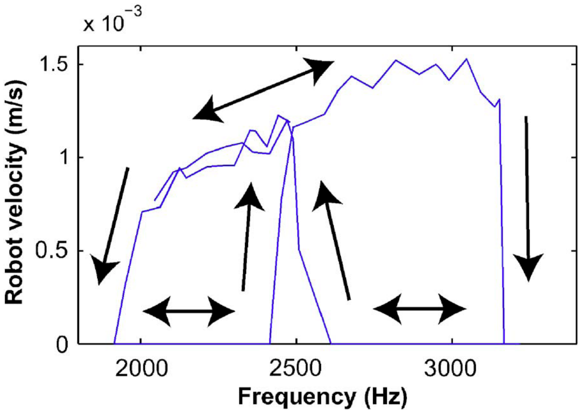


Figure 5: Robot velocity as a function of frequency with a driving field of 2.2 mT. When moving, the robot never comes to a complete rest to allow static friction can take effect. Driving with a frequency too far from resonance reduces the system energy and at some point the robot sticks to the substrate. Moving back toward resonance increases the absorbed energy, allowing the robot to begin moving again.

Taking inspiration from nature, this torque can be leveraged to create artificial bacterial flagella (Zhang et al., 2009).

The helical swimming robot consists of two parts: a helical tail and a magnetic metal head. The tails are 27 to 42 nm thick, less than 2 µm wide, and coil into diameters smaller than 3 µm. The robots are fabricated by a self-scrolling technique (Zhang et al., 2006). The helical swimming microrobots are propelled and steered precisely in water by a rotating magnetic field on the order of 1 to 2 mT. As the robot's principle dimensions approach those of individual cells, many of the experimental methods used with the robots parallel those used by their biological counterparts.

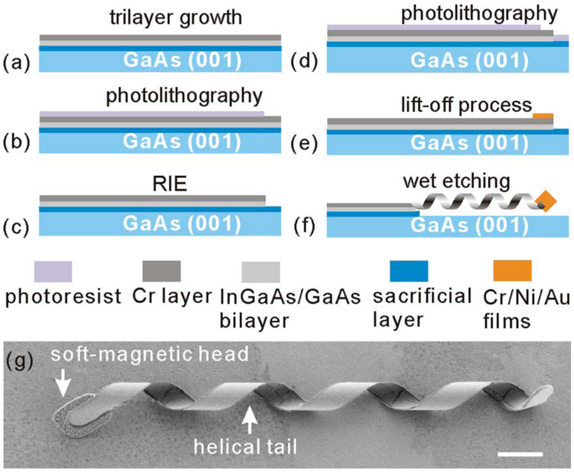


Figure 6: [(a)–(f)] Fabrication procedure of the ABF with InGaAs/GaAs/Cr helical tail. (g) FESEM image of an untethered ABF. The scale bar is 4 µm (Zhang et al., 2009).

6 SUMMARY

Recent advances in microbotics have demonstrated new capabilities in wirelessly controlling microscale structures with high precision over long distances in liquid environments. These breakthroughs make it possible to experimentally investigate the use of these microrobots for manipulating micro and nano size structures in as many as six degrees-of-freedom. Applications to nanomedicine in areas related to targeted medical therapies and molecular manipulation are clear, though many challenges must be addressed. To functionalize these devices and to improve their performance capabilities, fundamental issues in the role surface forces play must be addressed; biocompatibility must be ensured; loading and diffusion of biomolecules must be investigated; and interactions with and manipulation of tissue and macromolecules must be considered. There is a lot yet to do.

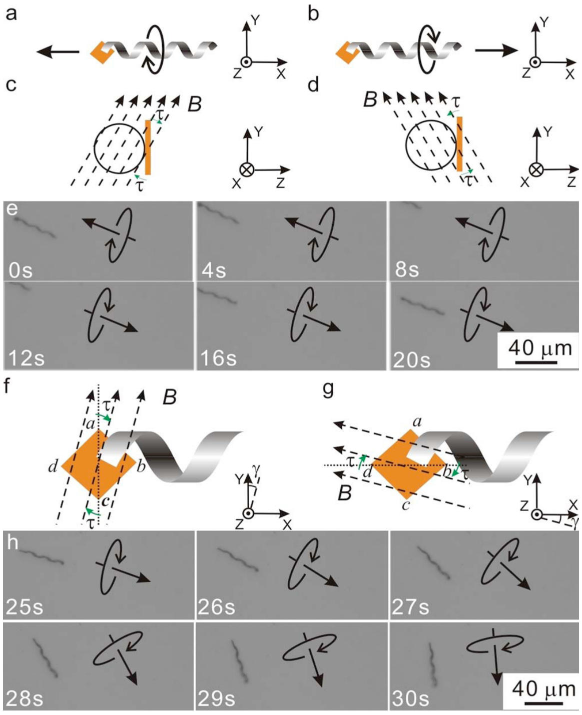


Figure 7: ABF swimming motion controlled by magnetic fields with field strength of 2.0 mT. [(a)-(d)] Schematic of a left-handed ABF swimming forward and backward. With the field B continuously rotating perpendicular to the X axis of the ABF, a misalignment angle between the field and the thin magnetic head will induce a magnetic torque (t) that attempts to align the ABF head with the field, resulting in rotation and propulsion of the ABF. (e) Optical microscope images of the forward/ backward motion of an ABF controlled by magnetic fields. The commanded translation and rotation directions of the ABF are indicated by the arrows. (f) If the field is rotated about the Z axis by an angle |γ|<45° with respect to the easy axis ac of the head, then the ABF is steered as it is propelled, as the easy axis ac attempts to align with the field. This is the steering principle used during normal operation of the ABF. (g) If the field is rotated about the Z axis by an angle |γ|<45° with respect to the easy axis bd, the ABF will instantaneously attempt to rotate perpendicular to the helix axis. However, steering using the bd easy axis is not possible simultaneously with forward/backward propulsion. (h) Optical microscope images of the turning motion of an ABF controlled by magnetic fields. The commanded translation and rotation directions of the ABF are indicated by the arrows.

REFERENCES

F. Tendick, S. S. Sastry, R. S. Fearing, and M. Cohn., "Applications of micromechatronics in minimally invasive surgery," *IEEE/ASME Transactions on Mechatronics,* vol. 3, pp. 34-42, 1998.

P. Dario, M. C. Carrozza, A. Benvenuto, and A. Menciassi, "Micro-systems in biomedical applications," *Journal of Micromechanics and Microengineering,* vol. 10, pp. 235-244, 2000.

M. J. Mack, "Minimally invasive and robotic surgery," *Journal of American Medical Association,* vol. 285, pp. 568-572, 2001.

M. C. Carrozza, P. Dario, and L. P. S. Jay, "Micromechatronics in surgery," *Transactions of the Institute of Measurement and Control,* vol. 25, pp. 309-327, 2003.

I. Kassim, L. Phee, W. S. Ng, F. Gong, P. Dario, and C. A. Mosse, "Locomotion techniques for robotic colonoscopy," *IEEE Engineering in Medicine and Biology Magazine,* pp. 49-56, May/June 2006.

E. A. Johannessen, L. Wang, S. W. J. Reid, D. R. S. Cumming, and J. M. and Cooper, "Implementation of radiotelemetry in a lab-in-a-pill format," *Lab on a Chip,* vol. 6, pp. 39-45, 2006.

A. Menciassi, D. Accoto, S. Gorini, and P. Dario, "Development of a biomimetic miniature robotic crawler," *Autonomous Robots,* vol. 21, pp. 155-163, 2007.

M. Wautelet, "Scaling laws in the macro-, micro- and nanoworlds," *European Journal of Physics,* vol. 22, pp. 601-611, Nov 2001.

E. M. Purcell, "Life at low Reynolds numbers," *American Journal of Physics,* vol. 45, pp. 3-11, 1977.

K. B. Yesin, K. Vollmers and B.J. Nelson, "Modeling and control of untethered biomicrorobots in a fluidic environment using electromagnetic fields," *International Journal of Robotics Research,* vol. 25, pp. 527-536, 2006.

J. J. Abbott, O. Ergeneman, M. Kummer, A. M. Hirt, and B. J. Nelson, " Modeling magnetic torque and force for controlled manipulation of soft-magnetic bodies," *IEEE Transactions on Robotics,* vol. 23, pp. 1247-1252, December 2007.

K. Vollmers, D. Frutiger, B. E. Kratochvil, and B. J. Nelson, "Wireless resonant magnetic actuation for untethered microrobots," *Applied Physics Letters,* vol. 92, April 2008.

L. Zhang, J.J. Abbott, L.X. Dong, B.E. Kratochvil, D.J. Bell, D.J. and B.J. Nelson, "Artificial bacterial flagella: Fabrication and magnetic control," *Applied Physics Letters,* vol. 94, February 2009.

L. Zhang, L. X. Dong, D. J. Bell, B. J. Nelson, C. Schoenenberger, and D. Gruetzmacher, "Fabrication and characterization of freestanding Si/Cr micro- and nanospirals," *Microelectron. Eng.,* vol. 83, pp. 1237-1240, Apr-Sep 2006.

BRIEF BIOGRAPHY

Brad Nelson is the Professor of Robotics and Intelligent Systems at ETH Zürich. His primary research focus is on microrobotics and nanorobotics with an emphasis on applications in biology and medicine. He received a B.S.M.E. from the University of Illinois at Urbana-Champaign and an M.S.M.E. from the University of Minnesota. He has worked as an engineer at Honeywell and Motorola and served as a United States Peace Corps Volunteer in Botswana, Africa, before obtaining a Ph.D. in Robotics from Carnegie Mellon University in 1995. He was an Assistant Professor at the University of Illinois at Chicago (1995-1998) and an Associate Professor at the University of Minnesota (1998-2002). He became a Full Professor at ETH Zürich in 2002.

Prof. Nelson has been awarded a McKnight Land-Grant Professorship and is a recipient of the Office of Naval Research Young Investigator Award, the National Science Foundation Faculty Early Career Development (CAREER) Award, the McKnight Presidential Fellows Award, and the Bronze Tablet. He was elected as a Robotics and Automation Society Distinguished Lecturer in 2003 and 2008 and won Best Paper Awards at major robotics conferences and journals in 2004, 2005, 2006, 2007, 2008 and 2009. He was named to the 2005 “Scientific American 50,” Scientific American magazine’s annual list recognizing fifty outstanding acts of leadership in science and technology from the past year for his efforts in nanotube manufacturing. His laboratory won the 2007 and 2009 RoboCup Nanogram Competition, both times the event has been held. He serves on the editorial boards of several journals, has served as the head of the Department of Mechanical and Process Engineering from 2005–2007, and is currently the Chairman of the ETH Electron Microscopy Center (EMEZ).

DYNAMIC MODELING OF ROBOTS USING RECURSIVE NEWTON-EULER TECHNIQUES

Wisama Khalil
Ecole Centrale de Nantes, IRCCyN UMR CNRS 6597, 1 Rue de la Noë, 44321 Nantes, France
Wisama.khalil@irccyn.ec-nantes.fr

Keywords: Dynamic Modelling, Newton-Euler, Recursive Calculation, Tree Structure, Parallel Robots, Flexible Joints, Mobile Robots.

Abstract: This paper present the use of recursive Newton-Euler to model different robotics systems. The main advantages of this technique are the facility of implementation by numerical or symbolical programming and providing models with reduced number of operations. In this paper the inverse and direct dynamic models of different robotics systems will be presented. At first we start by rigid tree structure robots, then these algorithms will be generalized for closed loop robots, parallel robots, and robots with lumped elasticity. At the end the case of robots with moving base will be treated.

1 INTRODUCTION

The dynamic modelling of robots is an important topic for the design, simulation, and control of robots. Different techniques have been proposed and used by the robotics community. In this paper we show that the use of Newton-Euler recursive technique for different robotics systems is easy to develop and programme. The proposed algorithm can be extended to many types of structures; serial, tree structure, closed, parallel, with a fixed base or with moving platform. The same technique can be used for robots with lumped elasticity or flexible links.

In section 2 we will recall the method used to describe the kinematics of the structure, and then in section 3 we present the inverse and the direct dynamic modeling of tree structure rigid robots which are considered as the base methods. The following sections present the generalization to the other systems.

2 DESCRIPTION OF THE KINEMATICS OF ROBOTS

The geometry of the structures will be described using the Modified Denavit and Hartenberg method as proposed in (Khalil and Kleinfinger, 1986). This method can take into account tree structures and closed loop robots. Its use facilitates the calculation of the base inertial parameters of robots (Gautier and Khalil, 1988, Khalil W., Bennis F., 1994, Khalil and Bennis, 1995).

2.1 Geometric Description of Tree Structure Robots

A tree structure robot is composed of n+1 links and n joints. Link 0 is the base and link n is a terminal link. The joints are either revolute or prismatic, rigid or elastic. The links are numbered consecutively from the base, link 0, to the terminal links. Joint j connects link j to link a(j), where a(j) denotes the link antecedent to link j. A frame R_i is attached to each link i such that (Figure 1):

- $\mathbf{z}_i$ is along the axis of joint i;
- $\mathbf{x}_i$ is taken along the common normal between $\mathbf{z}_i$ and one of the succeeding joint axes, which are fixed on link i.

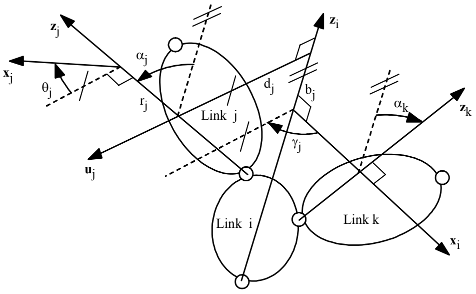


Figure 1: Geometric parameters for a link i.

In general the homogeneous transformation matrix ${}^{i}\mathbf{T}_j$, which defines the frame R_j relative to frame R_i is obtained as a function of six geometric parameters (γ_j , b_j, α_j, d_j, θ_j, r_j). Thus ${}^{i}\mathbf{T}_j$ is obtained as:

$$ {}^{i}\mathbf{T}_j = \mathbf{Rot}(\mathbf{z}, \gamma_j)\ \mathbf{Tran}(\mathbf{z}, b_j)\ \mathbf{Rot}(\mathbf{x}, \alpha_j)\ \mathbf{Tran}(\mathbf{x}, d_j)\ \mathbf{Rot}(\mathbf{z}, \theta_j)\ \mathbf{Tran}(\mathbf{z}, r_j) $$

After developing, this matrix can be partitioned as follows:

$$ {}^{i}\mathbf{T}_j = \begin{bmatrix} {}^{i}\mathbf{R}_j & {}^{i}\mathbf{P}_j \\ \mathbf{0}_{1\mathrm{x}3} & 0 \end{bmatrix} \qquad (1) $$

Where **R** defines the (3×3) rotation matrix and P defines the (3×1) vector defining the position of the origin of frame j with respect to frame i.

If $\mathbf{x}_i$ is along the common normal between $\mathbf{z}_i$ and $\mathbf{z}_j$, the parameters γ_j and b_j will be equal to zero.

The joint variable of joint j is denoted by:

$$ q_j = \bar{\sigma}_j\theta_j + \sigma_j r_j $$

where $\sigma_j = 0$ if joint j is revolute, $\sigma_j = 1$ if joint j is prismatic, and $\bar{\sigma}_j = 1 - \sigma$. We set $\sigma_j = 2$ to define a frame R_j fixed with respect to frame a(j). In this case, q_j and $\bar{\sigma}_j$ are not defined.

The serial structure is a special case of a tree structure where a(j)=j-1, γ_j =0, and b_j =0 for all j=1,..,n.

2.2 Description of Closed Loop Structure

The system is composed of L joints and n + 1 links, where link 0 is the fixed base and L > n. The number of independent closed loops is equal to:

$$ B = L - n $$

The joints are either active (motorized) or passive. The number of active joints is denoted N. The position and orientation of all the links can be determined as a function of the active joint variables.

To determine the geometric parameters of a mechanism with closed chains, we proceed as follows:

a) Construct an equivalent tree structure having n joints by virtually cutting each closed chain at one of its passive joints. Define the geometric parameters of the tree structure as given in section 2.1.

b) For each cut joint define two supplementary frames on one of the links connected by this joint. Assuming that a cut joint is numbered k (where k=n+1,…, L) and that the links connected by joint k are numbered i and j (where i and j <n) the frames will be defined as follows (Figure 2):

- frame R_k is defined fixed on link j such that a(k)=i, the axis $\mathbf{z}_k$ is along the axis of joint k, and $\mathbf{x}_k$ is along the common normal between $\mathbf{z}_k$ and $\mathbf{z}_j$. The matrix ${}^{i}\mathbf{T}_k$ will be determined using the general parameters γ_k, b_k, α_k, d_k, θ_k, r_k.

- frame R_{k+B} is aligned with R_k that is to say it is fixed on link j, but a(k+B)= j. The geometric parameters defining R_{k+B} are constant, we note that r_{k+B} and θ_{k+B} are zero since $\mathbf{x}_{k+B}$ is normal to $\mathbf{z}_j$.

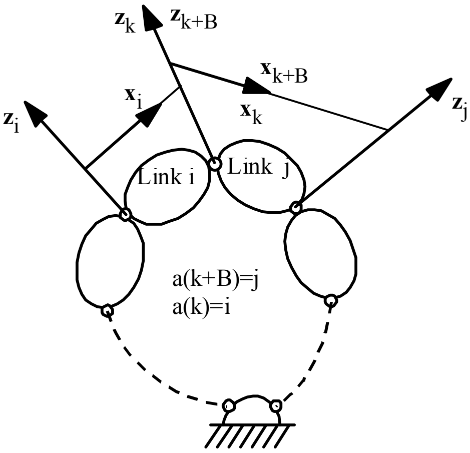


Figure 2: Frames of a cut joint k.

The joint variables are denotes as:

$$ \mathbf{q} = \begin{bmatrix} \mathbf{q}_{tr} \\ \mathbf{q}_{c} \end{bmatrix}, \mathbf{q}_{tr} = \begin{bmatrix} \mathbf{q}_{a} \\ \mathbf{q}_{p} \end{bmatrix} \qquad (2) $$

- $\mathbf{q}_{tr}$ vector containing the tree structure joint variables;
- $\mathbf{q}_{a}$ vector containing the N active joint variables;
- $\mathbf{q}_{p}$ vector containing the p=n−N passive joint variables of the equivalent tree structure;
- $\mathbf{q}_{c}$ vector containing the B variables of the cut joints.

Only the N active variables $\mathbf{q}_a$ are independent. Since R_k and R_{k+B} are aligned, the geometric constraint equations for each loop, which can be used to calculate the passive joint variables in terms of the active joint variables, can be written as:

$$ {}^{k+B}\mathbf{T}_j \ldots {}^{i}\mathbf{T}_k = \mathbf{I}_4 \qquad (3) $$

The kinematic constraint equations are obtained by using the fact that the screw of frame k is equal to that of frame k+B:

$$\begin{gathered} {}^0\mathbb{V}_k = {}^0\mathbb{V}_{k+B} \\ \mathbf{J}_k\,\dot{\mathbf{q}}_{\mathbf{b1}} = \mathbf{J}_{k+B}\dot{\mathbf{q}}_{\mathbf{b2}} \end{gathered} \tag{4}$$

$\mathbb{V}_j$ (6×1) kinematic screw vector of frame j, given by:

$$\mathbb{V}_j = \begin{bmatrix} \mathbf{V}_j^T & \boldsymbol{\omega}_j^{\ T} \end{bmatrix}^T \tag{5}$$

$\mathbf{V}_j$ linear velocity of the origin of frame R_j,

$\boldsymbol{\omega}_j$ angular velocity of frame j,

$\dot{\mathbf{q}}_{\mathbf{b1}}$ joint velocities from the base to frame k, through branch 1,

$\dot{\mathbf{q}}_{\mathbf{b2}}$ joint velocities from the base to frame k through branch 2.

3 DYNAMIC MODELING OF TREE STRUCTURE ROBOTS

3.1 Introduction

The most common in use methods to calculate the dynamic models are the Lagrange equations and the Newton Euler Equations (Craig 1986, Khalil and Dombre 2002, Angeles 2006).

The Lagrange equation is given as:

$$\boldsymbol{\Gamma} = \frac{\mathrm{d}}{\mathrm{dt}}\left[\frac{\partial L}{\partial \dot{\mathbf{q}}}\right]^T - \left[\frac{\partial L}{\partial \mathbf{q}}\right]^T \tag{6}$$

where Γ is the joint torques and forces, L is the Lagrangian of the robot defined as the difference between the kinetic energy E and the potential energy U of the system : L = E – U. After developing we obtain:

$$\boldsymbol{\Gamma} = \mathbf{A}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{H}(\mathbf{q},\dot{\mathbf{q}}) \tag{7}$$

where **A** is the inertia matrix of the robot and **H** is the Coriolis, Centrifuge and gravity torques.

Solving the previous equation to find Γ in terms of $(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}})$ is known as the inverse dynamic problem, and solving it to obtain $\ddot{\mathbf{q}}$ in terms of $(\mathbf{q},\dot{\mathbf{q}},\boldsymbol{\Gamma})$ is known as the direct dynamic model. The inverse dynamic model is obtained by substituting $(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}})$ into (7), whereas the direct model needs to inverse the inertia matrix.

$$\ddot{\mathbf{q}} = -(\mathbf{A})^{-1}(\boldsymbol{\Gamma} - \mathbf{H}) \tag{8}$$

The calculation of the Lagrange equations for systems with big number of degrees of freedom using developed symbolic methods is time consuming, and the obtained model will need more time to execute with respect to that of recursive methods. The recursive Newton-Euler algorithms have been shown to be an excellent tool to model rigid robots (Khalil and Kleinfinger, 1987, Khalil and Creusot, 1987, Khosla, 1987). In (Hollerbach, 1980) an efficient recursive Lagrange algorithm is presented but without achieving better performances than that of Newton-Euler.

The Newton-Euler equations giving the external forces and moments on a link j about the origin of frame j are written as:

$${}^j\mathbb{F}_j = {}^j\mathbb{J}_j\,{}^j\dot{\mathbb{V}}_j + \begin{bmatrix} {}^j\boldsymbol{\omega}_j \times \left({}^j\boldsymbol{\omega}_j \times {}^j\mathbf{MS}_j\right) \\ {}^j\boldsymbol{\omega}_j \times \left({}^j\mathbf{J}_j\,{}^j\boldsymbol{\omega}_j\right) \end{bmatrix} \tag{9}$$

where

$${}^j\mathbb{F}_j = \begin{bmatrix} {}^j\mathbf{F}_j \\ {}^j\mathbf{M}_j \end{bmatrix} \tag{10}$$

$\boldsymbol{\omega}_j$ the angular velocity of link j;

$\dot{\mathbf{V}}_j$ the linear acceleration of the origin of frame j;

$\mathbb{F}_j$ total external wrench on link j;

$\mathbf{F}_j$ total external forces on link j;

$\mathbf{M}_j$ total external moments on link j about O_j;

$\mathbb{J}_j$ (6×6) inertia matrix of link j:

$${}^j\mathbb{J}_j = \begin{bmatrix} M_j\mathbf{I}_3 & -{}^j\mathbf{M}\hat{\mathbf{S}}_j \\ {}^j\mathbf{M}\hat{\mathbf{S}}_j & {}^j\mathbf{J}_j \end{bmatrix} \tag{11}$$

Where M_j, $\mathbf{MS_j}$ and $\mathbf{J_j}$ are the standard inertial parameters of link j. They are respectively, the mass, the first moments, and the inertia matrix about the origin.

3.2 Calculation of the Inverse Dynamics using Recursive NE Algorithm

The algorithm consists of two recursive computations (Luh, Walker and Paul, 1980): forward recursion and backward recursion. The forward equations, from link 1 to link n, compute the link velocities and accelerations and consequently the dynamic wrench on each link. The backward equations, from link n to the base, provide the reaction wrenches on the links and consequently the joint torques.

This method gives the joint torques in terms of the joint positions, velocities and accelerations without explicitly computing the matrices **A** and **H**. That is to say the algorithm will be denoted by:

$$\boldsymbol{\Gamma} = \mathrm{NE}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}},\mathbf{f}_e,\mathbf{m}_e) \tag{12}$$

Where $\mathbf{f}_e$ and $\mathbf{m}_e$ are the external forces and moments of the links of the robot on the environment.

The forward recursive equations are based on the following equations (Khalil and Dombre 2002):

$$^{j}\mathbb{V}_{j} = {}^{j}\mathbb{T}_{i}\,{}^{i}\mathbb{V}_{i} + \dot{q}_{j}\,{}^{j}\mathbb{a}_{j} \qquad (13)$$

$$^{j}\dot{\mathbb{V}}_{j} = {}^{j}\mathbb{T}_{i}\,{}^{i}\dot{\mathbb{V}}_{i} + {}^{j}\boldsymbol{\gamma}_{j} + \ddot{q}_{j}\,{}^{j}\mathbb{a}_{j} \qquad (14)$$

$$^{j}\boldsymbol{\gamma}_{j} = \begin{bmatrix} {}^{j}\mathbf{R}_{i}\left[{}^{i}\boldsymbol{\omega}_{i}\times\left({}^{i}\boldsymbol{\omega}_{i}\times{}^{i}\mathbf{P}_{j}\right)\right]+2\sigma_{j}({}^{j}\boldsymbol{\omega}_{i}\times\dot{q}_{j}\,{}^{j}\mathbf{a}_{j}) \\ \bar{\sigma}_{j}\,{}^{j}\boldsymbol{\omega}_{i}\times\dot{q}_{j}\,{}^{j}\mathbf{a}_{j} \end{bmatrix} \qquad (15)$$

where

$^{j}\mathbb{a}_{j}$ is the (6x1) column matrix given as :

$$^{j}\mathbb{a}_{j} = [\,0\;\;0\;\;\sigma_{j}\;\;0\;\;0\;\;\bar{\sigma}_{j}\,]^{T} \qquad (16)$$

$^{j}\mathbb{T}_{i}$ and the screw transformation matrix is:

$$^{j}\mathbb{T}_{i} = \begin{bmatrix} {}^{j}\mathbf{R}_{i} & -{}^{j}\mathbf{R}_{i}\,{}^{i}\hat{\mathbf{P}}_{j} \\ \mathbf{0}_{3x3} & {}^{j}\mathbf{R}_{i} \end{bmatrix} \qquad (17)$$

The forward algorithm is given for j=1,…, n, with i = a(j), as follows:

$$^{j}\omega_{i} = {}^{j}\mathbf{R}_{i}\,{}^{i}\omega_{i} \qquad (18)$$

$$^{j}\omega_{j} = {}^{j}\omega_{i} + \bar{\sigma}_{j}\,\dot{q}_{j}\,{}^{j}\mathbf{a}_{j} \qquad (19)$$

$$^{j}\dot{\omega}_{j} = {}^{j}\mathbf{R}_{i}\,{}^{i}\dot{\omega}_{i} + \bar{\sigma}_{j}\,(\ddot{q}_{j}\,{}^{j}\mathbf{a}_{j} + {}^{j}\omega_{i}\times\dot{q}_{j}\,{}^{j}\mathbf{a}_{j}) \qquad (20)$$

$$^{j}\dot{\mathbf{V}}_{j} = {}^{j}\mathbf{R}_{i}\,({}^{i}\dot{\mathbf{V}}_{i} + {}^{i}\mathbf{U}_{i}\,{}^{i}\mathbf{P}_{j}) + \sigma_{j}\,(\ddot{q}_{j}\,{}^{j}\mathbf{a}_{j} + 2\,{}^{j}\omega_{i}\times\dot{q}_{j}\,{}^{j}\mathbf{a}_{j}) \qquad (21)$$

$$^{j}\mathbf{F}_{j} = \mathrm{M}_{j}\,{}^{j}\dot{\mathbf{V}}_{j} + {}^{j}\mathbf{U}_{j}\,{}^{j}\mathbf{MS}_{j} \qquad (22)$$

$$^{j}\mathbf{M}_{j} = {}^{j}\mathbf{J}_{j}\,{}^{j}\dot{\omega}_{j} + {}^{j}\omega_{j}\times({}^{j}\mathbf{J}_{j}\,{}^{j}\omega_{j}) + {}^{j}\mathbf{MS}_{j}\times{}^{j}\dot{\mathbf{V}}_{j} \qquad (23)$$

with

$$^{j}\mathbf{U}_{j} = {}^{j}\hat{\dot{\omega}}_{j} + {}^{j}\hat{\omega}_{j}\,{}^{j}\hat{\omega}_{j}$$

and where $\mathbf{a}_j$ is the unit vector along the $\mathbf{z}_j$ axis which is the axis of joint j.

The matrix $\hat{\mathbf{w}}$ defines the 3×3 vector product matrix associated to the (3×1) vector $\mathbf{W}$ such that:

$$\hat{\mathbf{w}} = \begin{bmatrix} 0 & -w_z & w_y \\ w_z & 0 & -w_x \\ -w_y & w_x & 0 \end{bmatrix} \qquad (24)$$

$$\mathbf{w}\times\mathbf{v} = \hat{\mathbf{w}}\;\mathbf{v}$$

These equations are initialized by $\omega_0 = \mathbf{0}$, $\dot{\omega}_0 = \mathbf{0}$, $\dot{\mathbf{V}}_0 = -\mathbf{g}$, $^{0}\mathbf{U}_0 = 0$, with $\mathbf{g}$ is the acceleration of gravity.

Initialising the linear acceleration $\dot{\mathbf{V}}_0$ by $-\mathbf{g}$ will take automatically the effect of gravity forces on all the links of the structure.

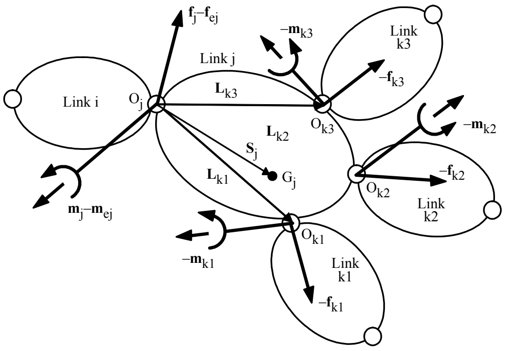


Figure 3: Forces and moments acting on a link j.

The backward recursive equations are deduced from the resultant forces and moments on link j around the origin of link j (Figure 3).

$$^{j}\mathbb{f}_{j} = {}^{j}\mathbb{F}_{j} + \sum_{k}\,{}^{k}\mathbb{T}_{j}^{T}\,{}^{k}\mathbb{f}_{k} + {}^{j}\mathbb{f}_{ej}$$

(25)

Where a(k)=j,

The backward equations can be calculated for j=n,…,1:

$$^{j}\mathbf{f}_{j} = {}^{j}\mathbf{F}_{j} + {}^{j}\mathbf{f}_{ej} \qquad (26)$$

$$^{j}\mathbf{m}_{j} = {}^{j}\mathbf{M}_{j} + {}^{j}\mathbf{m}_{ej} \qquad (27)$$

$$^{i}\mathbf{f}_{j} = {}^{i}\mathbf{R}_{j}\,{}^{j}\mathbf{f}_{j} \qquad (28)$$

$$^{i}\mathbf{f}_{ei} = {}^{i}\mathbf{f}_{ei} + {}^{i}\mathbf{f}_{j} \qquad (29)$$

$$^{i}\mathbf{m}_{ei} = {}^{i}\mathbf{m}_{ei} + {}^{i}\mathbf{R}_{j}\,{}^{j}\mathbf{m}_{j} + {}^{i}\mathbf{P}_{j}\ \mathrm{x}\ {}^{i}\mathbf{f}_{j} \qquad (30)$$

$$\Gamma_j = (\sigma_j\,{}^{j}\mathbf{f}_j + \bar{\sigma}_j\,{}^{j}\mathbf{m}_j)^{T\,j}\mathbf{a}_j + \mathrm{Ia}_j\,\ddot{q}_j + \mathrm{F}_{sj}\,\mathrm{sign}(\dot{q}_j) + \mathrm{F}_{vj}\,\dot{q}_j \qquad (31)$$

Where:

$\mathbf{f}_j$ and $\mathbf{m}_j$ are the reaction forces and moments of link a(j) on link j respectively, Ia_j is the inertia of the rotor and transmission gears of the motor of joint j, F_{sj} and F_{vj} are the coulomb and viscous friction parameters respectively, $^{j}\mathbf{f}_{ej}$ and $^{j}\mathbf{m}_{ej}$ are the external forces and moments of link j on the environment.

This algorithm is easy to program numerically or symbolically. The computational cost is linear with the number of degrees of freedom of the robot. To reduce the number of operations of the calculation of this model the base inertial parameters can be used instead of the standard inertial parameters and the technique of customized symbolic method can be applied (Khosla 1986 , Khalil and Kleinfinger 1987, Khalil and Creusot 1997).

3.3 Computation of the Direct Dynamic Model

The computation of the direct dynamic model is employed to carry out simulations for the purpose of testing the robot performances and studying the control laws. During simulation, the dynamic equations are solved for the joint accelerations given the input torques and the state of the robot (joint positions and velocities). Through integration of the joint accelerations, the robot trajectory is then determined.

The direct dynamic model can be obtained from Lagrange equation (8) as follows:

$$\ddot{\mathbf{q}} = \mathbf{A}^{-1}\,[\boldsymbol{\Gamma} - \mathbf{H}(\mathbf{q}, \dot{\mathbf{q}})]$$

Two methods based on Newton-Euler methods can be used to obtain the dynamic model: the first is based on calculating the **A** and **H** matrices using Newton-Euler inverse dynamic model in order to calculate the joint accelerations by (8); the second method is based on a recursive Newton-Euler algorithm that does not explicitly calculate the matrix **A** and has a computational cost that varies linearly with the number of degrees of freedom of the robot. For tree structure robots, the second method is more efficient, but the first method can be used for closed loop robots and other complicated systems. That is why we will present both methods.

3.3.1 Using the Inverse Dynamic Model to Calculate the Direct Dynamic Model

In this method the matrices $\mathbf{H}(\mathbf{q},\dot{\mathbf{q}})$ and $\mathbf{A}(\mathbf{q})$ are calculated using the inverse model by giving special values for the joint accelerations, joint velocities, external forces, friction, gravity (Walker and Orin 1982).

By comparing equations (7) and (12) we deduce that $\mathbf{H}(\mathbf{q}, \dot{\mathbf{q}})$ is equal to $\boldsymbol{\Gamma}$ if $\ddot{\mathbf{q}} = \mathbf{0}$, and that the ith column of **A** is equal to $\boldsymbol{\Gamma}$ if:

$$\ddot{\mathbf{q}} = \mathbf{u}_i,\ \dot{\mathbf{q}} = \mathbf{0},\ \mathbf{g} = \mathbf{0},\ \mathbf{f}_{ej} = \mathbf{0},\ \mathbf{m}_{ej} = \mathbf{0}$$

where $\mathbf{u}_i$ is the (nx1) unit vector whose ith element is equal to 1, and the other elements are zeros. Iterating the procedure for i = 1,…, n leads to the construction of the entire inertia matrix.

To reduce the computational complexity of this algorithm, we can make use of the base inertial parameters and the customized symbolic techniques. Moreover, we can take advantage of the fact that the inertia matrix **A** is symmetric.

3.3.2 Recursive NE Computation of the Direct Dynamic Model

This method is based on the recursive Newton-Euler equations and does not use explicitly the inertia matrix of the robot (Armstrong 1979, (Featherstone 1983, (Brandl, Johanni and Otter, 1986).

Using (9) and (25) the equilibrium equations of link j can be written as:

$$ {}^{j}\mathbb{J}_j\,{}^{j}\dot{\mathbb{V}}_j = {}^{j}\mathbb{f}_j + {}^{j}\boldsymbol{\beta}_j - \sum_{\mathbf{k}} {}^{k}\mathbb{T}_j^{T}\,{}^{k}\mathbb{f}_k \qquad (32)$$

where k denote the links articulated on link j such that a(k)=j, and

$$ {}^{j}\boldsymbol{\beta}_j = -{}^{j}\mathbb{f}_{ej} - \begin{bmatrix} {}^{j}\boldsymbol{\omega}_j \times \left({}^{j}\boldsymbol{\omega}_j \times {}^{j}\mathbf{MS}_j\right) \\ {}^{j}\boldsymbol{\omega}_j \times \left({}^{j}\mathbf{J}_j\,{}^{j}\boldsymbol{\omega}_j\right) \end{bmatrix} \qquad (33)$$

The joint accelerations are obtained as a result of three recursive computations:

i) first forward computations for j = 1, ..., n: in this step, we compute the screw transformation matrices ${}^{j}\mathbb{T}_i$, the link angular velocities ${}^{j}\omega_j$ as well as ${}^{j}\gamma_j$ and ${}^{j}\beta_j$ vectors, which appear in the link accelerations and the link wrenches equations respectively when $\ddot{\mathbf{q}} = \mathbf{0}$;

$$ {}^{j}\boldsymbol{\gamma}_j = \begin{bmatrix} {}^{j}\mathbf{R}_i\left[{}^{i}\boldsymbol{\omega}_i \times \left({}^{i}\boldsymbol{\omega}_i \times {}^{i}\mathbf{P}_j\right)\right] + 2\sigma_j({}^{j}\boldsymbol{\omega}_i \times \dot{q}_j\,{}^{j}\mathbf{a}_j) \\ \bar{\sigma}_j\,{}^{j}\boldsymbol{\omega}_i \times \dot{q}_j\,{}^{j}\mathbf{a}_j \end{bmatrix} \qquad (34)$$

$$ {}^{j}\boldsymbol{\beta}_j = -{}^{j}\mathbb{f}_{ej} - \begin{bmatrix} {}^{j}\boldsymbol{\omega}_j \times \left({}^{j}\boldsymbol{\omega}_j \times {}^{j}\mathbf{MS}_j\right) \\ {}^{j}\boldsymbol{\omega}_j \times \left({}^{j}\mathbf{J}_j\,{}^{j}\boldsymbol{\omega}_j\right) \end{bmatrix} \qquad (35)$$

ii) backward recursive computation: in this step we calculate the elements H_j, ${}^{j}\mathbb{J}_j^{*}$, ${}^{j}\beta_j^{*}$, ${}^{j}\mathbb{K}_j$, ${}^{j}\alpha_j$ which express $\ddot{q}_j$ and ${}^{j}\mathbb{f}_j$ in terms of ${}^{i}\dot{\mathbb{V}}_i$ in the third recursive equations. These equations are demonstrated in the following sub-section.

For j = n ... 1, compute:

$$H_j = ({}^{j}\mathbb{a}_j^{T}\,{}^{j}\mathbb{J}_j^{*}\,{}^{j}\mathbb{a}_j + Ia_j) \qquad (36)$$

$$ {}^{j}\mathbb{K}_j = {}^{j}\mathbb{J}_j^{*} - {}^{j}\mathbb{J}_j^{*}\,{}^{j}\mathbb{a}_j\,H_j^{-1}\,{}^{j}\mathbb{a}_j^{T}\,{}^{j}\mathbb{J}_j^{*} \qquad (37)$$

$$ {}^{j}\alpha_j = {}^{j}\mathbb{K}_j\,{}^{j}\gamma_j + {}^{j}\mathbb{J}_j^{*}\,{}^{j}\mathbb{a}_j\,H_j^{-1}\,(\tau_j + {}^{j}\mathbb{a}_j^{T}\,{}^{j}\beta_j^{*}) - {}^{j}\beta_j^{*} \qquad (38)$$

If a(j) ≠ 0, calculate also:

$$ {}^{i}\beta_i^{*} = {}^{i}\beta_i^{*} - {}^{j}\mathbb{T}_i^{T}\,{}^{j}\alpha_j \qquad (39)$$

$$ {}^{i}\mathbb{J}^{*}_{i} = {}^{i}\mathbb{J}^{*}_{i} + {}^{j}\mathbb{T}^{T}_{i}\,{}^{j}\mathbb{K}_{j}\,{}^{j}\mathbb{T}_{i} \tag{40} $$

These equations are initialized by

$$ {}^{j}\mathbb{J}^{*}_{j} = {}^{j}\mathbb{J}_{j} \text{ and } {}^{j}\boldsymbol{\beta}^{*}_{j} = {}^{j}\boldsymbol{\beta}_{j}. $$

iii) second forward recursive computations. Since the acceleration of the base is known ($\dot{\mathbf{V}}_0 = -\mathbf{g}$, $\dot{\boldsymbol{\omega}}_0 = \mathbf{0}$ for fixed base), the third recursive computation gives $\ddot{q}_j$ and ${}^{j}\mathbb{f}_{j}$ (if needed) for j = 1… n. as follows:

$$ \ddot{q}_j = H^{-1}_{j}\ [-{}^{j}\mathbb{a}^{T}_{j}\,{}^{j}\mathbb{J}^{*}_{j}\,({}^{j}\mathbb{T}_{i}\,{}^{i}\dot{\mathbb{V}}_{i} + {}^{j}\boldsymbol{\gamma}_{j}) + \tau_j + {}^{j}\mathbb{a}^{T}_{j}\,{}^{j}\boldsymbol{\beta}^{*}_{j}] \tag{41} $$

$$ {}^{j}\mathbb{f}_{j} = \begin{bmatrix} {}^{j}\mathbf{f}_{j} \\ {}^{j}\mathbf{m}_{j} \end{bmatrix} = {}^{j}\mathbb{K}_{j}\,{}^{j}\mathbb{T}_{i}\,{}^{i}\dot{\mathbb{V}}_{i} + {}^{j}\boldsymbol{\alpha}_{j} \tag{42} $$

$$ {}^{j}\dot{\mathbb{V}}_{j} = {}^{j}\mathbb{T}_{i}\,{}^{i}\dot{\mathbb{V}}_{i} + {}^{j}\mathbb{a}_{j}\,\ddot{q}_j + {}^{j}\boldsymbol{\gamma}_{j} \tag{43} $$

where

$$ \tau_j = \Gamma_j - F_{sj}\,\mathrm{sign}(\dot{q}_j) - F_{vj}\,\dot{q}_j \tag{44} $$

Calculation of the elements of the backward recursive equations

To simplify the notations, we consider the case of a serial structure of n joints. Expressing the acceleration of link n in terms of the acceleration of link n-1, and since ${}^{n+1}\mathbb{f}_{n+1} = \mathbf{0}$, we obtain:

$$ {}^{n}\mathbb{J}_{n}\,({}^{n}\mathbb{T}_{n-1}\,{}^{n-1}\dot{\mathbb{V}}_{n-1} + \ddot{q}_n\,{}^{n}\mathbb{a}_{n} + {}^{n}\boldsymbol{\gamma}_{n}) = {}^{n}\mathbb{f}_{n} + {}^{n}\boldsymbol{\beta}_{n} \tag{45} $$

Since:

$$ {}^{j}\mathbb{a}^{T}_{j}\,{}^{j}\mathbb{f}_{j} = \tau_j - Ia_j\,\ddot{q}_j $$

$$ \tau_j = \Gamma_j - F_{sj}\,\mathrm{sign}(\dot{q}_j) - F_{vj}\,\dot{q}_j $$

We obtain the joint acceleration of joint n:

$$ \ddot{q}_n = H^{-1}_{n} $$

$$ (-{}^{n}\mathbb{a}^{T}_{n}\,{}^{n}\mathbb{J}_{n}\,({}^{n}\mathbb{T}_{n-1}\,{}^{n-1}\dot{\mathbb{V}}_{n-1} + {}^{n}\boldsymbol{\gamma}_{n}) + \tau_n + {}^{n}\mathbb{a}^{T}_{n}\,{}^{n}\boldsymbol{\beta}_{n}) \tag{46} $$

where H_n is a scalar given as:

$$ H_n = ({}^{n}\mathbb{a}_{n}{}^{T}\,{}^{n}\mathbb{J}_{n}\,{}^{n}\mathbb{a}_{n} + Ia_n) \tag{47} $$

Substituting for $\ddot{q}_n$ from (46) and (45), we obtain the dynamic wrench ${}^{n}\mathbb{f}_{n}$ as:

$$ {}^{n}\mathbb{f}_{n} = \begin{bmatrix} {}^{n}\mathbf{f}_{n} \\ {}^{n}\mathbf{m}_{n} \end{bmatrix} = {}^{n}\mathbb{K}_{n}\,{}^{n}\mathbb{T}_{n-1}\,{}^{n-1}\dot{\mathbb{V}}_{n-1} + {}^{n}\boldsymbol{\alpha}_{n} \tag{48} $$

where:

$$ {}^{n}\mathbb{K}_{n} = {}^{n}\mathbb{J}_{n} - {}^{n}\mathbb{J}_{n}\,{}^{n}\mathbb{a}_{n}\,H^{-1}_{n}\,{}^{n}\mathbb{a}^{T}_{n}\,{}^{n}\mathbb{J}_{n} \tag{49} $$

$$ {}^{n}\boldsymbol{\alpha}_{n} = {}^{n}\mathbb{K}_{n}\,{}^{n}\boldsymbol{\gamma}_{n} + {}^{n}\mathbb{J}_{n}\,{}^{n}\mathbb{a}_{n}\,H^{-1}_{n}\,(\tau_n + {}^{n}\mathbb{a}^{T}_{n}\,{}^{n}\boldsymbol{\beta}_{n}) - {}^{n}\boldsymbol{\beta}_{n} \tag{50} $$

We now have $\ddot{q}_n$ and ${}^{n}\mathbb{f}_{n}$ in terms of ${}^{n-1}\dot{\mathbb{V}}_{n-1}$. Iterating the procedure for j = n – 1, we obtain:

$$ {}^{n-1}\mathbb{J}_{n-1}\,{}^{n-1}\dot{\mathbb{V}}_{n-1} = {}^{n-1}\mathbb{f}_{n-1} + {}^{n}\mathbb{T}^{T}_{n-1}\,{}^{n}\mathbb{f}_{n} + {}^{n-1}\boldsymbol{\beta}_{n-1} \tag{51} $$

which can be rewritten as:

$$ {}^{n-1}\mathbb{J}^{*}_{n-1}\,({}^{n-1}\mathbb{T}_{n-2}\,{}^{n-2}\dot{\mathbb{V}}_{n-2} + \ddot{q}_{n-1}\,{}^{n-1}\mathbb{a}_{n-1} + {}^{n-1}\boldsymbol{\gamma}_{n-1}) = {}^{n-1}\mathbb{f}_{n-1} + {}^{n-1}\boldsymbol{\beta}^{*}_{n-1} \tag{52} $$

where:

$$ {}^{n-1}\mathbb{J}^{*}_{n-1} = {}^{n-1}\mathbb{J}_{n-1} + {}^{n}\mathbb{T}^{T}_{n-1}\,{}^{n}\mathbb{K}_{n}\,{}^{n}\mathbb{T}_{n-1} \tag{53} $$

$$ {}^{n-1}\boldsymbol{\beta}^{*}_{n-1} = {}^{n-1}\boldsymbol{\beta}_{n-1} - {}^{n}\mathbb{T}^{T}_{n-1}\,{}^{n}\boldsymbol{\alpha}_{n} \tag{54} $$

Equation (52) has the same form as (45). Thus, we can express $\ddot{q}_{n-1}$ and ${}^{n-1}\mathbb{f}_{n-1}$ in terms of ${}^{n-2}\dot{\mathbb{V}}_{n-2}$. Iterating this procedure for j = n – 2, …, 1, we obtain $\ddot{q}_j$ and ${}^{j}\mathbb{f}_{j}$ in terms of ${}^{j-1}\dot{\mathbb{V}}_{j-1}$ for j = n – 1, ..., 1 as given by equations (41) and (43) which represent the general case.

4 INVERSE DYNAMIC MODELING OF CLOSED LOOP ROBOTS

The computation of the Inverse dynamic model of closed loop robots can be obtained by first calculating the inverse dynamic model of the equivalent tree structure robot, in which the joint variables satisfy the constraints of the loop. Then the closed loop torques of the active joints $\boldsymbol{\Gamma}_c$ are obtained by projecting the tree structure torques $\boldsymbol{\Gamma}_{tr}$ on the motorized joints using the transpose of the Jacobian matrix of the tree structure variables (or velocities) in terms of the active joint variables (or velocities).

$$ \boldsymbol{\Gamma}_c = \mathbf{G}^T\,\boldsymbol{\Gamma}_{tr}(\mathbf{q}_{tr}, \dot{\mathbf{q}}_{tr}, \ddot{\mathbf{q}}_{tr}) \tag{55} $$

where:

$$ \mathbf{G} = \frac{\partial \mathbf{q}_{tr}}{\partial \mathbf{q}_{a}} = \frac{\partial \dot{\mathbf{q}}_{tr}}{\partial \dot{\mathbf{q}}_{a}} \tag{56} $$

It can be written also as:

$$ \boldsymbol{\Gamma}_c = \boldsymbol{\Gamma}_a + \frac{\partial \dot{\mathbf{q}}_p}{\partial \dot{\mathbf{q}}_a}\boldsymbol{\Gamma}_p \tag{57} $$

Where:
$\boldsymbol{\Gamma}_a$ and $\boldsymbol{\Gamma}_p$ are the torque of actuated and passive joints of the tree structure.

The kinematics Jacobian matrix can be obtained from (4) representing the kinematics closed loop constraints.

There is no recursive method to obtain the direct dynamic model of closed loop robots. It can be computed using the inverse dynamic model by a procedure similar to that given in section (3.3.1) in order to obtain the matrices $\mathbf{A}_c$ and $\mathbf{H}_c$ of the following relation:

$$\Gamma_c = \mathbf{A}_c(\mathbf{q}_{tr})\ddot{\mathbf{q}}_a + \mathbf{H}_c(\mathbf{q}_{tr},\dot{\mathbf{q}}_{tr}) \quad (58)$$

5 INVERSE DYNAMIC MODELING OF PARALLEL ROBOTS

A parallel robot is a complex multi-body system having several closed loops. It is composed of a moving platform connected to a fixed base by parallel legs. The dynamic model can be obtained as described in the previous section, but in this section we present a method that takes into account the parallel structure. To simplify the notations we will present her the case of parallel robots with six degrees of freedom. Examples concerning reduced mobility robots are given in (Khalil and Ibrahim 2007).

The robot is composed of a fixed base and a mobile platform. They are connected using m parallel legs.

The inverse dynamic model gives the forces and torques of motorized joints as a function of the desired trajectory of the mobile platform.

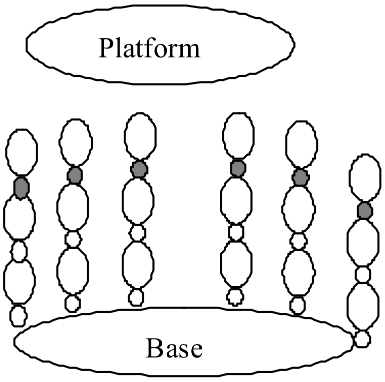


Figure 4: Parallel structure after separating the platform.

To obtain the dynamic models of parallel robots, we exploit their structural characteristics by decomposing the system into two subsystems: the platform and the legs.

The dynamics of the platform is calculated as a function of the Cartesian variables (spatial Cartesian position, velocity and acceleration of the platform), whereas the dynamics of the legs are calculated as a function of the joint variables of the legs $\left(\mathbf{q}_i,\dot{\mathbf{q}}_i,\ddot{\mathbf{q}}_i\right)$ for i=1,…,m. The active joint torques are obtained by the sum of these dynamics and projecting them on the active joint axes.

To project the dynamics of the platform on the active joint space we multiply it by the transpose of the robot Jacobian matrix, which gives the platform screw $\mathbb{V}_p$ in terms of the motorized joint velocities $\dot{\mathbf{q}}_a$, and to project the leg dynamics on the active joint space we use the Jacobian between these two spaces. Thus the dynamic model of the parallel structure is given by the following equation:

$$\mathbf{\Gamma} = \mathbf{J}_\mathbf{P}^\mathbf{T}\mathbb{F}_\mathbf{P} + \sum_{\mathbf{i=1}}^{\mathbf{m}}\left(\frac{\partial\dot{\mathbf{q}}_\mathbf{i}}{\partial\dot{\mathbf{q}}_\mathbf{a}}\right)^\mathbf{T}\mathbf{\Gamma}_\mathbf{i} \quad (59)$$

where

$\mathbb{F}_\mathbf{P}$ is the total forces and moments on the platform,

$\mathbf{J}_\mathbf{P}$ is the $(6\times n)$ kinematics Jacobian matrix of the robot, which gives the platform velocity $\mathbf{V}_\mathbf{P}$ (translational and angular) as a function of the active joint velocities:

$$\mathbb{V}_\mathbf{P} = \mathbf{J}_\mathbf{P}\dot{\mathbf{q}}_\mathbf{a} \quad (60)$$

$\mathbf{\Gamma}_\mathbf{i}$ is the inverse dynamic model of leg i, it is a function of $\left(\mathbf{q}_i,\dot{\mathbf{q}}_i,\ddot{\mathbf{q}}_i\right)$, which can be obtained in terms of the platform location, velocity and acceleration, using the inverse kinematic models of the legs. We note that $\mathbf{q}_\mathbf{i}$ does not include the passive joint variables connecting the legs to the platform.

In this section we suppose n=6, thus $\mathbf{J}_\mathbf{P}$ is (6×6) matrix.

The calculation of $\mathbf{J}_\mathrm{p}$ is obtained by inverting $\mathbf{J}^{\mathbf{-1}}_\mathrm{p}$, which is easy to obtain for most parallel structures.

$\mathbb{F}_\mathbf{P}$ is calculated by the Newton-Euler equation (9).

The calculation of $\partial\dot{\mathbf{q}}_\mathbf{i}/\partial\dot{\mathbf{q}}_\mathbf{a}$ is carried out by the following relation, which exploits the parallel structure of the robot:

$$\frac{\partial\dot{\mathbf{q}}_\mathbf{i}}{\partial\dot{\mathbf{q}}_\mathbf{a}} = \frac{\partial\dot{\mathbf{q}}_\mathbf{i}}{\partial\mathbf{v}_\mathbf{i}}\frac{\partial\mathbf{v}_\mathbf{i}}{\partial\mathbb{V}_\mathbf{P}}\frac{\partial\mathbb{V}_\mathbf{P}}{\partial\dot{\mathbf{q}}_\mathbf{a}} \quad (61)$$

with: $\mathbf{v}_\mathbf{i}$ is the Cartesian velocity transferred from leg i to the platform.

We can rewrite (61) as:

$$\frac{\partial\dot{\mathbf{q}}_\mathbf{i}}{\partial\dot{\mathbf{q}}_\mathbf{a}} = \mathbf{J}_\mathbf{i}^{\mathbf{-1}}\,\mathbf{J}_{\mathbf{vi}}\,\mathbf{J}_\mathbf{r} \quad (62)$$

$\mathbf{J}_\mathbf{i}$ is the kinematic Jacobian matrix of leg i such that:

$$\mathbf{v}_\mathbf{i} = \mathbf{J}_\mathbf{i}\,\dot{\mathbf{q}}_\mathbf{i} \quad (63)$$

$\mathbf{J}_{\mathbf{vi}}$ gives $\mathbf{v}_{\mathbf{i}}$ as a function of $\mathbb{V}_{\mathbf{P}}$:

$$\mathbf{v}_{\mathbf{i}} = \mathbf{J}_{\mathbf{vi}}\mathbb{V}_{\mathbf{P}} \tag{64}$$

For the Gough-Stewart platform (where the mobile platform is connected to the legs using spherical joints), we obtain:

$$\mathbf{J}_{\mathrm{vi}} = \frac{\partial \mathbf{v}_{\mathbf{i}}}{\partial \mathbb{V}_{\mathbf{P}}} = \begin{bmatrix} \mathbf{I}_3 & -\hat{\mathbf{P}}_{\mathrm{i}} \end{bmatrix} \tag{65}$$

Where $\mathrm{P_i}$ is the vector between the origin of the platform frame and the centre of the spherical joint linking the platform with leg i.

Finally the inverse dynamic model of the robot is given by the following form:

$$\mathbf{\Gamma} = \mathbf{J}_{\mathbf{p}}^{\mathbf{T}} \left[\mathbb{F}_{\mathbf{P}} + \sum_{\mathbf{i=1}}^{\mathbf{m}} \mathbf{J}_{\mathbf{vi}}^{\mathbf{T}} \mathbf{J}_{\mathbf{i}}^{\mathbf{-T}} \mathbf{\Gamma}_{\mathbf{i}} \right] \tag{66}$$

We note that the term between the brackets in (66) represents the dynamic model of the robot expressed in the Cartesian space of the platform frame (Khalil and Guegain, 2002).

6 INVERSE DYNAMIC MODELING OF ROBOTS WITH ELASTIC JOINTS

In this section we tree structure robots with lumped elasticity or flexible joints. The system can be described using Modified Denavit and Hartenberg method presented in section 2. Each joint could be either elastic or rigid (Khalil and Gautier, 2000).

6.1 Lagrange Dynamic Form

The general form of the dynamic model of a system with flexible joints has the same form as (7). It can be rewritten as:

$$\mathbf{\Gamma} = \mathbf{A(q)}\,\ddot{\mathbf{q}} + \mathbf{H(q}, \dot{\mathbf{q}}\,) \tag{67}$$

It can be partitioned as follows:

$$\Gamma = \begin{bmatrix} \Gamma_r \\ \Gamma_f \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{12}^{\mathrm{T}} & \mathbf{A}_{22} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_r \\ \ddot{\mathbf{q}}_f \end{bmatrix} + \begin{bmatrix} \mathbf{H}_r \\ \mathbf{H}_f \end{bmatrix} \tag{68}$$

Where $\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}$ are the (n×1) vectors of positions, velocities, and accelerations of rigid and elastic joints;

$\mathbf{H(q}, \dot{\mathbf{q}})$ is the (n×1) vector of Coriolis, centrifugal and gravity forces,

$\mathbf{A(q)}$ is the (n×n) inertia matrix of the system,

Γ_r is the vector of rigid joint torques,

Γ_f is the vector of elastic joint torques.

If joint j is flexible:

$$\Gamma_j = -\,\Delta q_j\, K_j \tag{69}$$

where K_j is the stiffness of the elastic joint,

$$\Delta q_j = q_j - q_{0j} \tag{70}$$

q_{0j} is the joint position corresponding to zero elasticity force.

In the case of a system with elasticity, the direct dynamic model has the same outputs as in the case of rigid bodies; it gives the joint accelerations as a function of the joint torques and of the system state variables ($\mathbf{q}$, $\dot{\mathbf{q}}$). It can be calculated using (68) by calculation the inverse of **A**.

In the case of a system with elasticity, the inverse dynamic model calculates the input torques and the elastic accelerations as a function of the joint positions, velocities and rigid joint accelerations. It is to be noted that the accelerations of the elastic variables cannot be specified independently. Using (68) to calculate the inverse model, we have first to calculate the acceleration elastic accelerations from the second row:

$$\Gamma_f = \begin{bmatrix} \mathbf{A}_{12}^{\mathrm{T}} & \mathbf{A}_{22} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_r \\ \ddot{\mathbf{q}}_f \end{bmatrix} + \mathbf{H}_f \tag{71}$$

then we can calculate the rigid joint torques from the first row.

6.2 Direct Dynamics of Systems with Flexible Joints using Recursive NE

The direct dynamic model of system with flexible joints can be calculated using the recursive direct dynamic model algorithm of rigid joints presented in section (3.3) after putting $\Gamma_j = -\,\Delta q_j\, K_j$. for the elastic joints.

Remark: We note that in case of rigid non motorized joint, the same algorithm can be used after putting $\Gamma_j = 0$.

6.3 Inverse Dynamics of Systems with Flexible Joints using Recursive NE

The recursive inverse dynamic algorithm of rigid links cannot be used for system with flexible joints since the accelerations of the flexible joints are unknown. On the contrary it can be used to obtain the **A** and **H** matrices as explained in section (3.3.1), then we can proceed as explained in 6.1 for the calculation of $\ddot{\mathbf{q}}_f$ and $\mathbf{\Gamma}_r$.

We propose here a recursive algorithm to solve this problem (Khalil and Gautier 2000). This algorithm consists of three recursive steps.

i) The first forward iteration is exactly the same as that of the direct dynamic model (section 3.3).

ii) The second backward recursive equations calculate the matrices giving the elastic accelerations $\ddot{q}_j$ and ${}^{j}\mathbb{f}_j$ as a function of ${}^{a(j)}\dot{\mathbb{V}}_{a(j)}$. These matrices can be defined using a similar procedure as in section (3.3). They can be calculated for j =n, ...,1, as follows:

- If joint j is elastic:

$$H_j = {}^{j}\mathbb{a}_j^{T}\,{}^{j}\mathbb{J}_j^{*}\,{}^{j}\mathbb{a}_j \tag{72}$$

$${}^{j}\mathbb{K}_j = {}^{j}\mathbb{J}_j^{*} - {}^{j}\mathbb{J}_j^{*}\,{}^{j}\mathbb{a}_j\,H_j^{-1}\,{}^{j}\mathbb{a}_j^{T}\,{}^{j}\mathbb{J}_j^{*} \tag{73}$$

$${}^{j}\alpha_j = {}^{j}\mathbb{K}_j\,{}^{j}\gamma_j + {}^{j}\mathbb{J}_j^{*}\,{}^{j}\mathbb{a}_j\,H_j^{-1}\,(-K_j\,\Delta q_j + {}^{j}\mathbb{a}_j^{T}\,{}^{j}\boldsymbol{\beta}_j^{*}) - {}^{j}\boldsymbol{\beta}_j^{*} \tag{74}$$

- If joint j is rigid:

$${}^{j}\mathbb{K}_j = {}^{j}\mathbb{J}_j^{*} \tag{75}$$

$${}^{j}\alpha_j = {}^{j}\mathbb{K}_j\,{}^{j}\gamma_j + {}^{j}\mathbb{J}_j^{*}\,{}^{j}\mathbb{a}_j\,\ddot{q}_j - {}^{j}\boldsymbol{\beta}_j^{*} \tag{76}$$

if a(j) ≠ 0, calculate:

$${}^{i}\boldsymbol{\beta}_i^{*} = {}^{i}\boldsymbol{\beta}_i^{*} - {}^{j}\mathbb{T}_i^{T}\,{}^{j}\alpha_j \tag{77}$$

$${}^{i}\mathbb{J}_i^{*} = {}^{i}\mathbb{J}_i^{*} + {}^{j}\mathbb{T}_i^{T}\,{}^{j}\mathbb{K}_j\,{}^{j}\mathbb{T}_i \tag{78}$$

The previous equations are initialized by:

$${}^{j}\mathbb{J}_j^{*} = {}^{j}\mathbb{J}_j, \text{ and } {}^{j}\boldsymbol{\beta}_j^{*} = {}^{j}\boldsymbol{\beta}_j.$$

The third recursive equations (for j = 1, ..., n) calculate $\ddot{q}_j$ for the elastic joints and the joint torques for the rigid joints using the following equation:

$${}^{j}\mathbb{f}_j = \begin{bmatrix} {}^{j}\mathbf{f}_j \\ {}^{j}\mathbf{m}_j \end{bmatrix} = {}^{j}\mathbb{K}_j\,{}^{j}\mathbb{T}_i\,{}^{i}\dot{\mathbb{V}}_i + {}^{j}\alpha_j \tag{79}$$

- if j is elastic:

$$\ddot{q}_j = H_j^{-1}[-{}^{j}\mathbb{a}_j^{T}\,{}^{j}\mathbb{J}_j^{*}\,({}^{j}\mathbb{T}_i\,{}^{i}\dot{\mathbb{V}}_i + {}^{j}\gamma_j) - K_j\,\Delta q_j + {}^{j}\mathbb{a}_j^{T}\,{}^{j}\boldsymbol{\beta}_j^{*} \tag{80}$$

$${}^{j}\dot{\mathbb{V}}_j = {}^{j}\mathbb{T}_i\,{}^{i}\dot{\mathbb{V}}_i + {}^{j}\mathbb{a}_j\,\ddot{q}_j + {}^{j}\gamma_j \tag{81}$$

- if j is rigid

$$\Gamma_j = (\sigma_j\,{}^{j}\mathbf{f}_j + \bar{\sigma}_j\,{}^{j}\mathbf{m}_j)^{T}\,{}^{j}\mathbf{a}_j + Ia_j\,\ddot{q}_j \tag{82}$$

7 DYNAMIC MODELING OF ROBOTS WITH MOVING BASE

The structure treated in this section includes a big number of systems such as: cars, mobile robots, mobile manipulators, walking robots, Humanoid robots, eel like robots (Khalil W., G. Gallot G., Boyer F., 2007), snakes like robots, flying robots, spatial vehicle, etc. The difference between all of these systems will be in the calculation of the interaction forces with the environment. In the previous sections the base is fixed thus the acceleration of the base is equal to zero, whereas in the case of a mobile base system the acceleration of the base must be determined in both direct and inverse dynamic models. The proposed recursive dynamic models are easy to implement and calculate using numerical calculation. The inverse dynamic model, which is used in general in the control problems, can be used in simulation too when the objective is to study the evolution of the base giving joint positions, velocities and accelerations of the other joints. The direct dynamic model can be used in simulation when the joint torques are specified.

We use the same notations of section 2 to describe the structure. The base fixed frame R_0 is defined wrt the world fixed frame R_w by the transformation matrix ${}^{w}\mathbf{T}_0$. This matrix is supposed known at t = 0, it will be updated by integrating the base acceleration. The velocity and acceleration of the base are represented by the (6x1) vectors $\mathbb{V}_0$ and $\dot{\mathbb{V}}_0$ respectively.

The Cartesian velocities and accelerations of the links are calculated using the recursive equations (13)-(17).

7.1 General form of the Dynamic Models

The dynamic model of a robot with moving base can be represented by the following relation:

$$\begin{bmatrix} \mathbf{0}_{6x1} \\ \Gamma \end{bmatrix} = \mathbf{A} \begin{bmatrix} {}^{0}\dot{\mathbb{V}}_0 \\ \ddot{\mathbf{q}} \end{bmatrix} + \mathbf{H} \tag{83}$$

$\mathbf{\Gamma}$ $(n \times 1)$ vector of joint torques,
$\mathbf{q}$ $(n \times 1)$ vector of joint positions,
$\mathbf{A}$ is the $(6+n) \times (6+n)$ inertia matrix of the robot, it can be partitioned as follows:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{12}^{T} & \mathbf{A}_{22} \end{bmatrix} \tag{84}$$

$\mathbf{A}_{11}$ is the (6×6) inertia matrix of the composed link 0, which is composed of the inertia of all the links referred to frame R_0 (the base).

$\mathbf{A}_{22}$ is the $(n \times n)$ inertia matrix of the other links when the head is fixed,

$\mathbf{A}_{12}$ is the $(6 \times n)$ coupled inertia matrix of the joints and the base. It reflects the effect of the joint accelerations on the base motion, and the dual effect of base accelerations on the joint motions.

$\mathbf{H}$ is the $(n+6)\times 1$ vector representing the Coriolis, centrifugal, gravity and external forces effect on the robot. Its elements are functions of the base and joint velocities and the external forces. This vector can be partitioned as follows:

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_1 \\ \mathbf{H}_2 \end{bmatrix} \quad (85)$$

where:

$\mathbf{H}_1$ the Coriolis, centrifugal, gravity and external forces on the base.

$\mathbf{H}_2$ the Coriolis, centrifugal, gravity and external forces on the links 1,…,n.

The inverse dynamic model gives the joint torques and the base acceleration in terms of the desired trajectory (position, velocity and acceleration) of the articulated system (links 1 to n) and the base position and velocity. Using equation (83) and (84), the inverse dynamic model is solved by using the first row of equation (83) to obtain the base acceleration:

$${}^{0}\dot{\mathbb{V}}_0 = -\left(\mathbf{A}_{11}\right)^{-1}\left(\mathbf{H}_1 + \mathbf{A}_{12}\ddot{\mathbf{q}}\right) \quad (86)$$

Then the second row of (83), can be used to find the joint torques:

$$\Gamma = \mathbf{A}_{12}^{T}\,\dot{\mathbb{V}}_0 + \mathbf{A}_{22}\,\ddot{\mathbf{q}} + \mathbf{H}_2 \quad (87)$$

The direct dynamic model gives the joint accelerations and the base acceleration in terms of the position and velocity of the base and the articulated system and the joint input torques. Thus using (83), the direct dynamic model is solved as follows:

$$\begin{bmatrix} \dot{\mathbb{V}}_0 \\ \ddot{\mathbf{q}} \end{bmatrix} = \mathbf{A}^{-1}\begin{bmatrix} -\mathbf{H}_1 \\ \Gamma - \mathbf{H}_2 \end{bmatrix} \quad (88)$$

The calculation of **A** and **H** can be done by Lagrange method. They can also be calculated using the inverse dynamic model of tree structure of section (3.2) and using the procedure of section (3.3.1). The base can be taken into account by either of the following methods:

- The velocity and acceleration of the base will be the initial conditions $\dot{\mathbb{V}}_0$ and $\boldsymbol{\omega}_0$ for the forward recursive calculation. The backward recursive calculation must continue to j=0, where this new iteration will obtain the 6 equations of Newton-Euler equations of the base.

- We can assign link 1 to be the base, and suppose that link 0 is a virtual link whose inertial parameters are equal to zero but has the velocity and acceleration of the base. This can be done by putting σ_2=2. The six equations of the base will be those of $\mathbb{f}_1 = 0$;

Solving the inverse and direct dynamic problems using **A** and **H** may be very time consuming for systems with big number of degrees of freedom (as the eel like robot). Therefore, we propose here to use a recursive method, which is easy to programme, and its computational complexity is linear wrt the number of degrees of freedom.

The recursive Newton-Euler algorithm is based on the kinematic equations presented in section 3.

7.2 Recursive NE Calculation of the Inverse Dynamic Model of Robots with Mobile Base

The inverse dynamic algorithm in this case consists of three recursive equations (a forward, then a backward, then a forward).

i) Forward recursive calculation:

In this step we calculate the screw transformation matrices, link velocities, and the elements of the accelerations and external wrenches on the links, which are independent of the acceleration of the robot base ($\dot{\mathbf{V}}_0, \dot{\boldsymbol{\omega}}_0$). Thus we calculate for j=1,…,n: ${}^{j}\mathbb{T}_i$, ${}^{j}\mathbb{V}_j$ and ${}^{j}\boldsymbol{\gamma}_j$ using equations (13)-(17). We calculate also ${}^{j}\boldsymbol{\beta}_j$ representing the elements of the Newton-Euler equations, which are independent of the base acceleration in equations (14) and (15) such that:

$${}^{j}\zeta_j = {}^{j}\boldsymbol{\gamma}_j + \ddot{q}_j\,{}^{j}\mathbf{a}_j \quad (89)$$

$${}^{j}\boldsymbol{\beta}_j = -{}^{j}\mathbb{f}_{ej} - \begin{bmatrix} {}^{j}\boldsymbol{\omega}_j \times \left({}^{j}\boldsymbol{\omega}_j \times {}^{j}\mathbf{MS}_j\right) \\ {}^{j}\boldsymbol{\omega}_j \times \left({}^{j}\mathbf{J}_j\,{}^{j}\boldsymbol{\omega}_j\right) \end{bmatrix} \quad (90)$$

ii) Backward recursive equations:

In this step we obtain the base acceleration using the inertial parameters of the composite link 0, where the composite link j consists of the links j, j+1, …, n.

We note that (32), giving the equilibrium equation of link j, can be rewritten using (90) as:

$$ {}^{j}\mathbb{f}_{j} = {}^{j}\mathbb{J}_{j}\,{}^{j}\dot{\mathbb{V}}_{j} - {}^{j}\boldsymbol{\beta}_{j} + \sum_{\mathbf{k}} {}^{k}\mathbb{T}_{j}^{T}\,{}^{k}\mathbb{f}_{k} \qquad (91) $$

Applying the Newton-Euler equations on the composite link j, we obtain:

$$ {}^{j}\mathbb{f}_{j} = {}^{j}\mathbb{J}_{j}\,{}^{j}\dot{\mathbb{V}}_{j} - {}^{j}\boldsymbol{\beta}_{j} + \sum_{s(j)} {}^{s(j)}\mathbb{T}_{j}^{T}\left({}^{s(j)}\mathbb{J}_{s(k)}\,{}^{s(j)}\dot{\mathbb{V}}_{s(j)} - {}^{s(j)}\boldsymbol{\beta}_{s(j)} \right) \qquad (92) $$

Where s(k) means all the links succeeding joint j, that is to say joining j to any terminal link.

Substituting for ${}^{s(j)}\dot{\mathbb{V}}_{s(j)}$ in terms of ${}^{j}\dot{\mathbb{V}}_{j}$ using (14), we obtain:

$$ {}^{s(j)}\dot{\mathbb{V}}_{s(j)} = {}^{s(j)}\mathbb{T}_{j}\,{}^{j}\dot{\mathbb{V}}_{j} + \sum_{r} {}^{s(j)}\mathbb{T}_{r}\,{}^{r}\boldsymbol{\zeta}_{r} \qquad (93) $$

Where r denotes all links between j and s(j).

From (92), we obtain:

$$ {}^{j}\mathbb{f}_{j} = {}^{j}\mathbb{J}_{j}^{c}\,{}^{j}\dot{\mathbb{V}}_{j} - {}^{j}\boldsymbol{\beta}_{j}^{c} \qquad (94) $$

with:

$$ {}^{j}\mathbb{J}_{j}^{c} = {}^{j}\mathbb{J}_{j}^{c} + \sum_{k} {}^{k}\mathbb{T}_{j}^{T}\,{}^{k}\mathbb{J}_{k}^{c}\,{}^{k}\mathbb{T}_{j} \qquad (95) $$

$$ {}^{j}\boldsymbol{\beta}_{j}^{c} = {}^{j}\boldsymbol{\beta}_{j}^{c} - \sum_{k} {}^{k}\mathbb{T}_{j}^{T}\,{}^{k}\boldsymbol{\beta}_{k}^{c} + {}^{k}\mathbb{T}_{j}^{T}\,{}^{k}\mathbb{J}_{k}^{c}\,{}^{k}\boldsymbol{\zeta}_{k} \qquad (96) $$

${}^{j}\mathbb{J}_{j}^{c}$ is the inertial matrix of the composite link j.

For j = 0, and supposing ${}^{0}\mathbb{f}_{0}$ is equal to zero, we obtain using (94):

$$ {}^{0}\dot{\mathbb{V}}_{0} = \left({}^{0}\mathbb{J}_{0}^{c} \right)^{-1}\,{}^{0}\boldsymbol{\beta}_{0}^{c} \qquad (97) $$

To conclude, the recursive equations of this step consist of initialising ${}^{n}\mathbb{J}_{n}^{c} = {}^{n}\mathbb{J}_{n}$, ${}^{n}\boldsymbol{\beta}_{n}^{c} = {}^{n}\boldsymbol{\beta}_{n}$ and then calculating (95)-(96) for j = n,…, 0. At the end ${}^{0}\dot{\mathbb{V}}_{0}$ is calculated by (97).

Comparing (97) with (68) we can deduce that $\mathbf{A}_{11}$ is equal to ${}^{0}\mathbb{J}_{0}^{c}$, whereas ${}^{0}\boldsymbol{\beta}_{0}^{c}$ is equal to ($\mathbf{H}_{1} + \mathbf{A}_{12}\ddot{\mathbf{q}}$).

iii) Forward recursive equations:

After calculating ${}^{0}\dot{\mathbb{V}}_{0}$, the wrench ${}^{j}\mathbb{f}_{j}$ and the joint torques are obtained using equations (6) and (22) for j= 1,…, n as:

$$ {}^{j}\dot{\mathbb{V}}_{j} = {}^{j}\mathbb{T}_{i}\,{}^{i}\dot{\mathbb{V}}_{i} + {}^{j}\boldsymbol{\zeta}_{j} \qquad (98) $$

$$ {}^{j}\mathbb{f}_{j} = \begin{bmatrix} {}^{j}\mathbf{f}_{j} \\ {}^{j}\mathbf{m}_{j} \end{bmatrix} = {}^{j}\mathbb{J}_{j}^{c}\,{}^{j}\dot{\mathbb{V}}_{j} - {}^{j}\boldsymbol{\beta}_{j}^{c} \qquad (99) $$

The joint torque is calculated by projecting ${}^{j}\mathbb{f}_{j}$ on the joint axis, and by taking into account the friction and the actuators inertia:

$$ \Gamma_{j} = {}^{j}\mathbb{f}_{j}^{T}\,{}^{j}\mathbb{a}_{j} + F_{sj}\,\mathrm{sign}\left(\dot{q}_{j}\right) + F_{vj}\,\dot{q}_{j} + I_{aj}\,\ddot{q}_{j} \qquad (100) $$

It is to be noted that the inverse dynamic model algorithm can be used in the dynamic simulation of the mobile robot when the objective is to study the effect of the joint motions on the base. In this case the joint positions, velocities and accelerations trajectories are given. At each sampling time the acceleration of the base will be integrated to provide the angular and linear velocities for the next sampling time.

7.3 Recursive Direct Dynamic Model

The direct dynamic model consists of three recursive calculations in the same order as those of the inverse dynamic model (forward, backward and forward):

i) Forward recursive equations:

We calculate the link Cartesian velocities using (13) and the terms of Cartesian accelerations and equilibrium equations of the links that are independent of the accelerations of the base and of the joints. We calculate the following recursive equations for j = 1, ..., n:

$$ {}^{j}\boldsymbol{\gamma}_{j} = \begin{bmatrix} {}^{j}\mathbf{R}_{i}\left[{}^{i}\boldsymbol{\omega}_{i} \times \left({}^{i}\boldsymbol{\omega}_{i} \times {}^{i}\mathbf{P}_{j} \right) \right] + 2\sigma_{j}({}^{j}\boldsymbol{\omega}_{i} \times \dot{q}_{j}\,{}^{j}\mathbf{a}_{j}) \\ \bar{\sigma}_{j}\,{}^{j}\boldsymbol{\omega}_{i} \times \dot{q}_{j}\,{}^{j}\mathbf{a}_{j} \end{bmatrix} \qquad (101) $$

$$ {}^{j}\boldsymbol{\beta}_{j} = -{}^{j}\mathbb{f}_{ej} - \begin{bmatrix} {}^{j}\boldsymbol{\omega}_{j} \times \left({}^{j}\boldsymbol{\omega}_{j} \times {}^{j}\mathbf{MS}_{j} \right) \\ {}^{j}\boldsymbol{\omega}_{j} \times \left({}^{j}\mathbf{J}_{j}\,{}^{j}\boldsymbol{\omega}_{j} \right) \end{bmatrix} \qquad (102) $$

ii) Backward recursive equations:

In this second step, we first initialise ${}^{n}\mathbb{J}_{n}^{*} = {}^{n}\mathbb{J}_{n}$, ${}^{n}\boldsymbol{\beta}_{n}^{*} = {}^{n}\boldsymbol{\beta}_{n}$ and then we calculate for j = n,…,1 the following elements, which permit to calculate ${}^{j}\mathbb{f}_{j}$ and $\ddot{q}_{j}$ in terms of ${}^{i}\dot{\mathbb{V}}_{i}$ and will be used in the third recursive equations (these matrices can be obtained using a similar procedure as for the direct dynamic model of rigid links):

$$ H_{j} = {}^{j}\mathbb{a}_{j}^{T}\,{}^{j}\mathbb{J}_{j}^{*}\,{}^{j}\mathbb{a}_{j} + Ia_{j} \qquad (103) $$

$$ {}^{j}\mathbb{K}_j = {}^{j}\mathbb{J}_j^* - {}^{j}\mathbb{J}_j^*\, {}^{j}\mathbb{a}_j H_j^{-1}\, {}^{j}\mathbb{a}_j^T\, {}^{j}\mathbb{J}_j^* \tag{104} $$

$$ {}^{i}\mathbb{J}_i^* = {}^{i}\mathbb{J}_i + {}^{j}\mathbb{T}_i^T\, {}^{j}\mathbb{K}_j\, {}^{j}\mathbb{T}_i \tag{105} $$

$$ \tau_j = \Gamma_j - F_{sj}\,\mathrm{sign}(\dot{q}_j) - F_{vj}\,\dot{q}_j \tag{106} $$

$$ {}^{j}\boldsymbol{\alpha}_j = {}^{j}\mathbb{K}_j\, {}^{j}\boldsymbol{\gamma}_j + {}^{j}\mathbb{J}_j^*\, {}^{j}\mathbb{a}_j H_j^{-1}\left(\tau_j + {}^{j}\mathbb{a}_j^T\, {}^{j}\boldsymbol{\beta}_j^*\right) - {}^{j}\boldsymbol{\beta}_j^* \tag{107} $$

$$ {}^{i}\boldsymbol{\beta}_i^* = {}^{i}\boldsymbol{\beta}_i - {}^{j}\mathbb{T}_i^T\, {}^{j}\boldsymbol{\alpha}_j \tag{108} $$

iii) Forward recursive equations:

At first, the base acceleration is calculated by the following relation:

$$ {}^{0}\dot{\mathbb{V}}_0 = \left({}^{0}\mathbb{J}_0^*\right)^{-1}\, {}^{0}\boldsymbol{\beta}_0^* \tag{109} $$

We note that ${}^{0}\boldsymbol{\beta}_0^*$ is a function of $\boldsymbol{\tau}$, whereas ${}^{0}\boldsymbol{\beta}_0^c$ (used in the inverse model) is a function of $\ddot{\mathbf{q}}$.

$\ddot{q}_j$ and ${}^{j}\mathbb{f}_j$ (if desired) are calculated for j=1,…,n using the following equations:

$$ \ddot{q}_j = H_j^{-1}\left[-{}^{j}\mathbb{a}_j^T\, {}^{j}\mathbb{J}_j^*\left({}^{j}\dot{\mathbb{V}}_{j-1} + {}^{j}\boldsymbol{\gamma}_j\right) + \tau_j + {}^{j}\mathbb{a}_j^T\, {}^{j}\boldsymbol{\beta}_j^*\right] \tag{110} $$

$$ {}^{j}\mathbb{f}_j = {}^{j}\mathbb{K}_j\, {}^{j}\mathbb{T}_i\, {}^{i}\dot{\mathbb{V}}_i + {}^{j}\alpha_j \tag{111} $$

where:

$$ {}^{j}\dot{\mathbb{V}}_j = {}^{j}\dot{\mathbb{V}}_{j-1} + {}^{j}\mathbb{a}_j \ddot{q}_j + {}^{j}\boldsymbol{\gamma}_j \tag{112} $$

8 CONCLUSIONS

This paper presents the inverse and direct dynamic modeling of different robotics systems. The dynamic models are developed using the recursive Newton-Euler formalism. The inverse model provides the torque of the joint and the acceleration of the free degrees of freedom such as the elastic joints, or the acceleration of the base in case of mobile base.

The direct model provides the joint acceleration of the joints including those of the free degrees of freedom.

These algorithms constitute the generalization of the algorithms of articulated manipulators to the other cases.

The proposed methods have been applied on more complicated systems such as:

- flexible link robots (Boyer and Khalil, 1998),
- Micro continuous system (Boyer, Porez and Khalil, 2006),
- hybrid structure, where the robot is composed of parallel modules, which are connected in serie, (Ibrahim, Khalil 2010).

REFERENCES

Angeles J. 2002. *Fundamentals of Robotic Mechanical Systems*. Second edition, Springer-Verlag, New York.

Armstrong W.W., 1979. Recursive solution to the equation of motion of an N-links manipulator. In *Proc. 5th World Congress on Theory of Machines and Mechanisms*, p. 1343-1346.

Boyer, F., Khalil,W, 1998. An efficient calculation of flexible manipulator inverse dynamic. In, *Int. Journal of Robotics Research,* vol. 17, No.3, pp.282-293

Boyer, F., Porez M., Khalil, W. 2006. Macro-continuous torque algorithm for a three-dimensional eel-like robot. In *IEEE Robotics transaction*, vol.22, No.4, 2006, pp.763-775.

Brandl H., Johanni R., Otter M., .1986. A very efficient algorithm for the simulation of robots and multibody systems without inversion of the mass matrix. In *Proc. IFAC Symp. on Theory of Robots*, Vienne, p. 365-370.

Craig J.J., 1986. *Introduction to robotics: mechanics and control*. Addison Wesley Publishing Company, Reading.

Featherstone R., 1983. The calculation of robot dynamics using articulated-body inertias. In the *Int. J. of Robotics Research*, Vol. 2(3), p. 87-101.

Gautier Gautier,M., Khalil W. 1990. Direct calculation of minimum set of inertial parameters of serial robots. In *IEEE Trans. on Robotics and Automation*, Vol. RA-6(3), p. 368-373.

Ibrahim, O., Khalil, W, 2010. Inverse and direct dynamic models of Hybride robots. In *Mechanism and machine theory,* Volume 45, Issue 4, p. 627-640.

Luh J.Y.S., Walker M.W., Paul R.C.P.,1980. On-line computational scheme for mechanical manipulators. In *Trans. of ASME, J. of Dynamic Systems, Measurement, and Control*, Vol. 102(2) p. 69-76.

Khalil W., Kleinfinger J.-F., 1986. A new geometric notation for open and closed-loop robots. In *Proc. IEEE Int. Conf. on Robotics and Automation*, San Francisco, p. 1174-1180.

Khalil W., Kleinfinger J.-F., 1987. Minimum operations and minimum parameters of the dynamic model of tree structure robots. In *IEEE J. of Robotics and Automation*, Vol. RA-3(6), p. 517-526.

Khalil W., Bennis F., 1994. Comments on Direct Calculation of Minimum Set of Inertial Parameters of Serial Robots. In *IEEE Trans. on Rob.& Automation*, Vol. RA-10(1), p. 78-79.

Khalil W., Creusot D., 1997. SYMORO+: a system for the symbolic modelling of robots. In *Robotica*, Vol. 15, p. 153-161.

Khalil W., Gautier M., 2000. Modeling of mechanical systems with lumped elasticity", In *Proc. IEEE Int. Conf. on Robotics and Automation*, San Francisco, p. 3965-3970.

Khalil, W., Dombre, E. 2002. *Modeling identification and control of robots*. Hermes, Penton-Sciences, London.

Khalil W. and Guegan S.,2004. Inverse and Direct Dynamic Modeling of Gough-Stewart Robots. In

IEEE Transactions on Robotics and Automation, 20(4), p. 754-762.

Khalil W., G. Gallot G., Boyer F., 2007. Dynamic Modeling and Simulation of a 3-D Serial Eel-Like Robot. In *IEEE Transactions on Systems, Man and Cybernetics, Part C: Application and reviews, Vol. 37, N° 6.*

Khalil W., Ibrahim O., 2007**.** General solution for the Dynamic modeling of parallel robots. In *Journal of Intelligent and Robotic Systems*, Vol.49, pp.19-37.

Khosla P.K., 1986. Real-time control and identification of direct drive manipulators. Ph. D. Thesis, Carnegie Mellon.

Walker M.W., Orin D.E., 1982. Efficient dynamic computer simulation of robotics mechanism. In *Trans. of ASME, J. of Dynamic Systems, Measurement, and Control*, Vol. 104, p. 205-211.

BRIEF BIOGRAPHY

Wisama Khalil received the Ph.D. and the “Doctorat d’Etat” degrees in robotics and control engineering from the University of Montpellier, France, in 1976 and 1978, respectively. Since 1983, he has been a Professor at the Automatic Control and Robotics Department, Ecole Centrale de Nantes, France. He is the coordinator of Erasmus Mundus master course EMARO “European Master in Advanced Robotics”. He is carrying out his research within the Robotics team, Institut de Recherche en Communications et Cybernétique de Nantes (IRCCyN). His current research interests include modeling, control, and identification of robots. He has more than 100 publictions in journals and international conferences.

EMOTIVE DRIVER ADVISORY SYSTEM

Oleg Gusikhin
Ford Motor Company, Research and Innovation Center
2101 Village Road, Dearborn, MI 48121, U.S.A.

EXTENDED ABSTRACT

In 2007, Ford, in cooperation with Microsoft, introduced an in-car communication and entertainment system, SYNC. This system enables Bluetooth and USB connectivity for consumer phones and MP3 players and allows hands-free voice-activated control of brought-in devices. Since its initial introduction, there has been rapid growth of SYNC-enabled services, such as remote monitoring of vehicle health, personalized traffic reports, weather, news, and turn-by-turn directions utilizing data-over-voice technology. Furthermore, SYNC takes advantage of existing networking capabilities of smart phones/PDAs by providing a SYNC API to mobile application developers.

The Emotive Driver Advisory System (EDAS) is a Ford Research project that fills the technology pipeline for future SYNC versions, exploiting advances in information technology and consumer electronics to enhance the driver's experience. EDAS was inspired by recent developments in affective computing, open mic grammar-based speech recognition, embodied conversational agents, and humanoid robotics focusing on personalization and context-aware adaptive and intelligent behavior. The EDAS concept was revealed at the 2009 Consumer Electronics Show and the 2009 North American International Auto Show as EVA, Emotive Voice Activation.

The core elements of EDAS include an emotive and natural spoken dialogue system and an AVATAR-based visual interface integrated with adaptive vehicle controls and cloud-based infotainment. The system connects the vehicle, the driver, and the environment, while providing the dialogue strategy best suited for the given driving context and emotive status of the driver.

Voice interaction is the prevalent method for the driver to interface with vehicle systems for hands-free, eyes-free communication. The effectiveness of such communication depends on the quality and sophistication of both speech recognition and speech generation. The EDAS spoken dialogue system allows recognition of the driver's commands in an open mic, natural, non-hierarchical manner. In turn, the system response depends on the driving environment, as well as the driver's status. The responses can be more extensive and engaging in open road conditions, while concise in high traffic situations. The ability to recognize the driver's emotions and generate emotions in response can further improve such communication. The spoken interface is augmented by the AVATAR as a universal intelligent gauge. The AVATAR supplements the emotive intent in delivering system messages, as well as providing non-verbal cues into the status of the active task.

The system leverages cloud-based infotainment, allowing for personalized, context-aware and interactive delivery of infotainment services. The ability to maintain connectivity between vehicle systems and the internet not only gives access to a vast amount of up-to-date information, but also allows outsourcing of computationally intensive tasks to a remote server, tapping into the power of cloud computing. Specifically, we demonstrate how EDAS enhances four most common in-vehicle infotainment activities: points of interest, news radio, music and refueling notification and advice.

BRIEF BIOGRAPHY

Dr. Oleg Gusikhin is a Technical Leader at Ford Manufacturing, Vehicle Design and Safety Research Laboratory. He received his Ph.D. from the St. Petersburg Institute of Informatics and Automation of Russian Academy of Sciences and an MBA from the Ross Business School at the University of Michigan. For over 15 years, he has been working at Ford Motor Company in different functional areas including Information Technology, Advanced Electronics Manufacturing, and Research & Advanced Engineering. During his tenure at Ford, Dr. Gusikhin has been involved in the design and implementation of advanced information technology and intelligent controls for manufacturing and vehicle systems. Dr. Gusikhin is a recipient of 2004 Henry Ford Technology Award and two Ford

Research and Advanced Engineering Technical Achievement Awards. He holds 2 patens and is a co-author of 8 patent applications on advanced vehicle infotainment technology.

FINGERTIP FORCE MEASUREMENT BY IMAGING THE FINGERNAIL

John Hollerbach
University of Utah, U.S.A.

Abstract: Shear and normal forces from fingertip contact with a surface are measured by external camera images of the fingernail. Due to mechanical interaction between the surface, fingertip bone, and fingernail, regions of tension or compression are set up that result in reddening or whitening due to blood flow. The effect is quantitative enough to serve as a transducer of fingertip force. Due to individual differences, calibration is required for the highest accuracy. Automated calibration is achieved by use of a magnetically levitated haptic interface probe.

BRIEF BIOGRAPHY

John M. Hollerbach is Professor of Computing, and Research Professor of Mechanical Engineering, at the University of Utah. He also directs the Robotics Track, a joint graduate program between the School of Computing and Department of Mechanical Engineering. From 1989-1994 he was the Natural Sciences and Engineering/Canadian Institute for Advanced Research Professor of Robotics at McGill University, jointly in the Departments of Mechanical Engineering and Biomedical Engineering. From 1982-1989 he was on the faculty of the Department of Brain and Cognitive Sciences and a member of the Artificial Intelligence Laboratory at MIT; from 1978-1982 he was a Research Scientist. He received his BS in chemistry ('68) and MS in mathematics ('69) from the University of Michigan, and SM ('75) and PhD ('78) from MIT in Computer Science. He is presently the Vice President for Technical Activities of the IEEE Robotics and Automation Society, and Editor of the International Journal of Robotics Research.

ROBOTICS AND AUTOMATION

FULL PAPERS

POSITION ESTIMATION OF MOBILE ROBOTS CONSIDERING CHARACTERISTIC TERRAIN PROPERTIES

Michael Brunner, Dirk Schulz
Department of Unmanned Systems, Fraunhofer-Institute FKIE, Wachtberg, Germany
{michael.brunner, dirk.schulz}@fkie.fraunhofer.de

Armin B. Cremers
Department of Computer Science, University of Bonn, Bonn, Germany
abc@cs.uni-bonn.de

Keywords: Mobile robots, Position estimation, Terrain classification, Machine learning.

Abstract: Due to the varying terrain conditions in outdoor scenarios the kinematics of mobile robots is much more complex compared to indoor environments. In this paper we present an approach to predict future positions of mobile robots which considers the current terrain. Our approach uses Gaussian process regression (GPR) models to estimate future robot positions. An unscented Kalman filter (UKF) is used to project the uncertainties of the GPR estimates into the position space. The approach utilizes optimized terrain models for estimation. To decide which model to apply, a terrain classification is implemented using Gaussian process classification (GPC) models. The transitions between terrains are modeled by a 2-step Bayesian filter (BF). This allows us to assign different probabilities to distinct terrain sequences, while taking the properties of the classifier into account and coping with false classifications. Experiments showed the approach to produce better estimates than approaches considering only a single terrain model and to be competitive to other dynamic approaches.

1 INTRODUCTION

The kinematics of mobile robots in outdoor scenarios is much more complex than in indoor environments due to the varying terrain conditions. Therefore, truly reliable velocity controls for robots which are able to drive up to *4 m/s* or even faster (e.g. Figure 1) are hard to design. The drivability is primarily determined by the terrain conditions. Thus, we developed a machine learning system which predicts the future positions of mobile robots using optimized terrain models.

In this paper we present a Gaussian processes (GPs) based method for estimating the positions of a mobile robot. Our approach considers the different terrain conditions to improve prediction quality. Gaussian process regression (GPR) models are utilized to estimate the translational and rotational velocities of the robot. These estimated velocities are transferred into the position space using an unscented Kalman filter (UKF). By projecting the uncertainty values of the GPR estimates onto the positions, the UKF enables us to also capitalize the GPR uncertainties. To distinguish the different terrains the robot is traversing, we classify the spectrums of vertical accelerations using Gaussian process classification (GPC). The transitions between terrains are modeled by a 2-step Bayesian filter (BF). This allows us to assign different probabilities to distinct terrain sequences, as we incorporate the properties of the classifier.

The remainder of this paper is organized as follows: Related work is presented in Section 2. In Section 3 we explain GPs. In Section 4 we describe our dynamic approach. Experiments are shown in Section 5. We conclude in Section 6.

2 RELATED WORK

Several works study Gaussian processes (GPs) and their application to machine learning problems. A detailed view is provided by Rasmussen and Williams (Rasmussen and Williams, 2005). Other works are Williams (Williams, 2002), and MacKay (MacKay, 1998).

There is a lot of research on prediction of positions or trajectories of mobile robots. Many different systems are proposed, like a stereo-vision approach (Agrawal and Konolige, 2006), probability

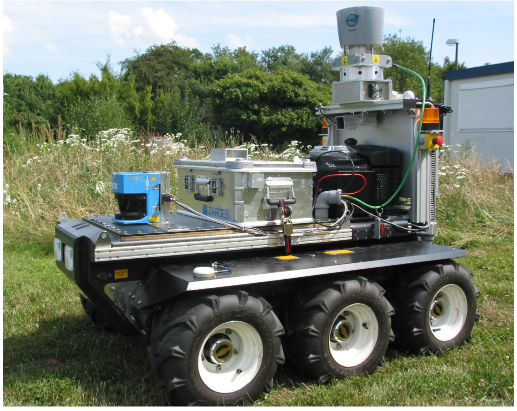


Figure 1: The experimental Longcross platform *Suworow*.

networks (Burgard et al., 1996) or an approach using a particle filter combined with a Monte-Carlo method (Thrun et al., 2000). In (Seyr et al., 2005) artificial neural networks (ANN) are employed to build a model predictive controller (MPC). The current prediction error is considered to improve the quality of the velocity estimations. Another least-square support vector machine (LS-SVM) based controller adjusts it's data model iteratively by removing the least important data point from the model when adding a new point (Li-Juan et al., 2007).

Similar to the work presented here, GPs are used by Girard et al. (Girard et al., 2003) to make predictions several time steps ahead. The uncertainty of the previous step is integrated in the regression to track the increasing uncertainty of the estimation. Hence, the uncertainty value of the GP is the accumulated uncertainty of the previous time series. In contrast, we are able to relate the Gaussian uncertainties by using an UKF. A similar approach has been suggested by (Ko et al., 2007b), and (Ko et al., 2007a). They estimated the trajectory of an autonomous blimp by combining GPR with an UKF or ordinary differential equations (ODE), respectively. Localization of wireless devices, like mobile telecommunication devices or mobile robots is solved by modeling the probability distribution of the signal strength with GPs integrated in a BF (Ferris et al., 2006).

Several works employ information about the current terrain conditions to improve the navigation systems of mobile robots. One intuitive approach is to distinguish between traversable and non-traversable terrain (Dahlkamp et al., 2006), (Rasmussen, 2002). In contrast to the binary separation, Weiss et al. (Weiss et al., 2006) are using a SVM to classify the vertical accelerations and hence the terrain type. Using spectral density analysis (SDA) and principal component analysis (PCA) Brooks et al. (Brooks et al., 2005) concentrate on the vibrations a mobile robot experiences while driving. The preprocessed data records are categorized through a voting scheme of binary classifiers. Another way to analyse the driving conditions is to measure the slippage (Iagnemma and Ward, 2009), (Ward and Iagnemma, 2007).

Kapoor et al. (Kapoor et al., 2007) are using GPs with pyramid-match kernels to classify objects from visual data. Urtasun and Darrell (Urtasun and Darrell, 2007) chose a GP latent variable model to achieve a better classification by reducing the input dimension.

Although many works concentrate on position estimation as well as terrain classification, none combined GPR and GPC to solve both problems at once for a velocity controller of high speed outdoor robots.

3 GAUSSIAN PROCESSES

GPs are applicable to regression as well as classification tasks. In contrast to other methods GPs do not have any parameters that have to be determined manually. However, the kernel function affects the properties of a GP essentially and must be chosen by hand. The parameters of of GPs can be automatically optimized using the training data. Furthermore, GPs provide uncertainty values additionally to the estimates. These properties makes GPs attractive for regression and classification task like position estimation and terrain classification. However, a drawback of GPs is their running time which is quadratic in the number of training cases due to the inversion of the kernel matrix.

3.1 Regression

Computing the predictive distribution of GPs consists of three major steps. First, we determine the Gaussian distribution of the training data and the test data. To integrate the information of the training data into the later distribution, we compute the joint distribution. Finally, this joint distribution is transformed to the predictive equations of GPs by conditioning it completely on the training data.

Let $X = [x_1, \ldots, x_n]^\top$ be the matrix of the n training cases x_i. The measured process outputs are collected in $y = [y_1, \ldots, y_n]^\top$. The noise of the random process is modeled with zero mean and variance σ_n^2. The kernel function is used to computed the similarities between two cases. Our choice is the squared exponential kernel given by

$$k(x_i, x_j) = \exp(-\tfrac{1}{2}|x_i - x_j|^2), \tag{1}$$

where k is the kernel and x_i, x_j are two inputs. A further quantity we need to provide the prior distribution of the training data is the kernel matrix of the training data $K = K(X,X)$. It is given by $K_{ij} = k(x_i,x_j)$ using the kernel function. Now, we are able to specify the distribution of y:

$$y \sim \mathcal{N}(0, K + \sigma_n^2 I). \tag{2}$$

We are trying to learn the underlying function of the process. To get the distribution of the values of the underlying function $f = [f_1,\dots,f_n]^\top$ we simply need to neglect the noise term from the distribution of the training data.

$$f \sim \mathcal{N}(0,K). \tag{3}$$

Secondly, we determine the distribution of the test data. Let $X_* = [x_{*_1},\dots,x_{*_{n_*}}]^\top$ be the set of n_* test cases assembled in a matrix like the training data. $f_* = [f_{*_1},\dots,f_{*_{n_*}}]^\top$ are the function values of the test cases. The normal distribution of the test data is therefore given by

$$f_* \sim \mathcal{N}(0, K_{**}), \tag{4}$$

where $K_{**} = K(X_*,X_*)$ is the kernel matrix of the test cases, representing the similarities of this data points.

To combine training and test data distributions in a joint Gaussian distribution we further require the kernel matrix of both sets, denoted by $K_* = K(X,X_*)$. Consequently, the joint distribution of the training and test data is:

$$\begin{bmatrix} y \\ f_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K + \sigma_n^2 I & K_* \\ K_*^\top & K_{**} \end{bmatrix}\right). \tag{5}$$

This combination allows us to incorporate the knowledge contained in the training data into the distribution of the function values of the test cases f_*, the values of interest.

Since the process outputs y are known, we can compute the distribution of f_* by conditioning it on y, resulting in the defining predictive distribution of GP models.

$$f_*|X,y,X_* \sim \mathcal{N}(\mathbb{E}[f_*],\ \mathbb{V}[f_*]) \tag{6}$$

with the mean and the variance given as

$$\mathbb{E}[f_*] = K_*^\top [K + \sigma_n^2 I]^{-1} y \tag{7}$$

$$\mathbb{V}[f_*] = K_{**} - K_*^\top [K + \sigma_n^2 I]^{-1} K_*. \tag{8}$$

The final distribution is defined only in terms of the three different kernel matrices and the training targets y.

3.2 Classification

The basic principle of GPC for multi-class problems is similar to GPR. Yet GPC suffers from the problem that the class labels $p(y|f)$ are not Gaussian distributed. Hence, the distribution of the function values given the training data

$$p(f|X,y) = \frac{p(y|f)p(f|X)}{p(y|X)} \tag{9}$$

is also not Gaussian. X and f still denote the training data and its function values, whereas y now represents the class labels of the training data. Yet to use GPs we have to have Gaussian distributed variables and must approximate $p(f|X,y)$, by the Laplace method for instance. Once the approximation is done, the further procedure is similar to GPR models, resulting in the predictive distribution of the GPC models $f_*|X,y,x_*$. Class probabilities are computed by drawing samples from this final distribution and applying the softmax function to squeeze the values into $[0,1]$.

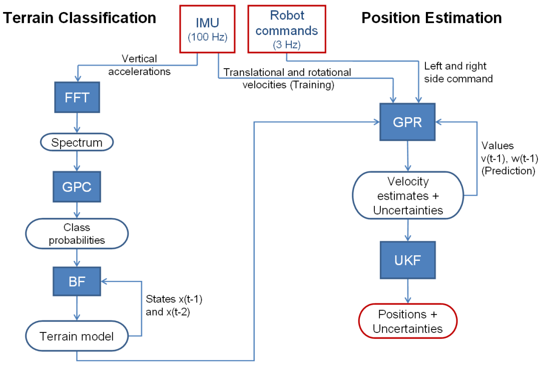


Figure 2: Graphical model of our dynamic approach showing the processing of the data and the dependencies between the terrain classification and position estimation.

GPC have the same properties as GPR. The class probabilities can be seen as a confidence value and are perfectly suited to be combined with probabilistic filters.

4 DYNAMIC APPROACH

Simple controller consider only one type of terrain or use one averaged terrain model. In contrast, our approach considers the different effects on the kinematics of mobile robots caused by varying terrain conditions. Our algorithm splits naturally in two parts, the estimation of the robot's position and the classification of the terrain. Figure 2 shows a graphical model of our dynamic approach.

4.1 Position Estimation

The position estimation consists of two methods. We use GPs to do regression on the robot's velocities and an UKF to transfer the results into the position space.

The values used were provided by an IMU installed on the robot. A series of preceding experiments were done to determine the composition of data types which in combination with the projection into the position space allowed the best position estimation. The data compositions compared in these experiments varied in two aspects, first the type of the data, and second the amount of past information.

The data combinations considered were: (1) the change in x-direction[1] and y-direction, (2) the change in x-direction, y-direction and in the orientation, (3) the x-velocity and y-velocity, (4) the x-velocity, y-velocity and the change in the orientation, (5) the translational and rotational velocity, and (6) the translational and rotational velocity and the orientation change. The number of past values was altered from 1 to 4 values to determine the minimal amount of previous information necessary for the system to be robust, while at the same time does not cloud the data records and does not complicate the learning task. Leading to a total number of 24 different data compositions. While differing in the motion model and complexity, all approaches can be easily transformed into positions. However, some approaches suffer from a poor transformation function, making them rather useless.

The results showed that a single past value of the translational and rotational velocity works best for position estimation. Thus, the data records of the GPR reads as follows:

$$[l_{t-1}, l_t, r_{t-1}, r_t, v_{t-1}, \omega_{t-1}] \tag{10}$$

Since our Longcross robot has a differential drive, l_i and r_i denote the values of the speed commands to the robot for the left and right side of it's drive. v_i and ω_i represent the translational and rotational velocity, respectively. To learn a controller we need both the commands given to the robot and the implementation of these commands. The values of interest are the current velocities, v_t and ω_t.

We use GPR to solve this regression task which not only provides the estimates of translational and rotational velocities but also gives us uncertainty values. We transform the estimates into positions using an UKF. Additionally, the uncertainties are projected onto these positions and are propagated over time (Figure 3). Starting with a quite certain position the sizes of the error ellipses are increasing, due to the uncertainties of the velocity estimates and the increasing uncertainties of the previous positions they rely on.

Given only the first translational and rotational velocity, the regression does not rely on any further IMU values. It takes previous estimates as inputs for the next predictions. Assuming the first velocities to be zero (i.e. the robot is standing) makes our approach completely independent of the IMU device. However, IMU data is required for training.

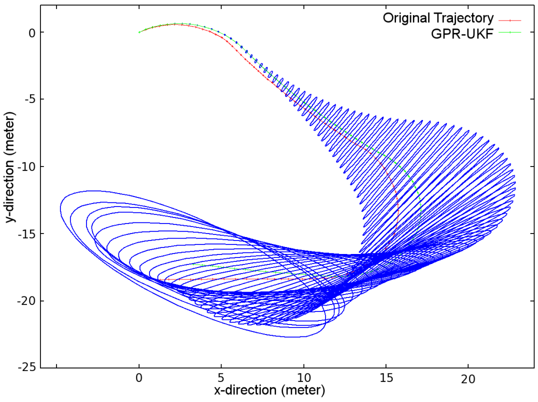


Figure 3: Original (red) and predicted (green) trajectory of a 30 second drive. The UKF error ellipses are displayed in blue. Starting out with a small uncertainty the error ellipses (blue) increase over time, due to the uncertainty of previous positions and the velocity estimates.

For each type of terrain we trained two[2] GPR models with data recorded only on that terrain. This has several advantages: Firstly, the effects of the robot commands on a specific terrain can be learned more accurately, since they are not contaminated by effects on other terrain types. Secondly, in combination more effects can be learned. And thirdly, due to the allocation of the training data to several GPR models the size and complexity of each model is smaller compared to what a single model covering all terrain types would require. The classification described in the next paragraph enables us to select the appropriate terrain models for regression.

4.2 Terrain Classification

The terrain classification is implemented with a GPC model, and a 2-step BF is used to model the transitions between terrains and to smooth the classification results.

As for the position estimation task preceding experiments were conducted to determine the best suited data records for the classification task. In contrast, here are mainly two possible data types, the vertical velocities and accelerations.

By taking a look at the raw IMU data it is evident that the different terrains are hardly distinguishable and that it may need to be preprocessed. To be sure, we classified vertical velocity and acceleration records of varying sizes. Since the single values carry

[1] All directions are relative to the robot. The x-directions points always in the front direction of the robot, and the y-direction orthogonal laterally.

[2] GPs are not able to handle multi-dimensional outputs. Hence, we need one GPR model for each velocity, translational and rotational.

no information about the terrain, we took vectors of 25, 30, 35, and 40 values. The results showed the raw data to be improper for the classification task at hand.

The information about the terrain types is enclosed in the vibrations a mobile robot experiences while driving. We used the fast Fourier transform (FFT) to extract the frequency information. Given that the FFT works best with window sizes being a power of 2, we chose 16, 32, 64, and 128 as input lengths. Due to the symmetric structure of the FFT output, it is sufficient to use only the first half of the result. This reduces the complexity of the problem and allows us to consider larger window sizes as for the raw data. Using Fourier transformed vertical velocities and accelerations improved the classification results, as was expected. We found the vector of 128 vertical acceleration values to work best for our task. Hence, the input to the GPC model is given by

$$[fftaz_0, \ldots, fftaz_{63}], \tag{11}$$

where $fftaz_i$ denotes the ith value of the FFT result.

If the robot's speed is too slow the terrain characteristics are not longer present in the vibrations, and therefore a good classification is impossible. Thus, we omit data for which the robot's translational velocity is below some threshold τ.

Due to the native charaters of the terrains, the driving conditions vary within a terrain type. This makes it hard to achieve a perfect classification. To deal with false predictions and to account for the properties of the classifier, we used a 2-step BF to model the transitions between terrains. The quality values GPC models provide, come in terms of class probabilities which are well suited to be included into a probabilistic filter like the BF. The defining equation of our 2-step BF is slightly modified to accommodate the classification and the dependency on two previous beliefs.

$$\begin{aligned} b(x_{t|t-1}) \propto & \sum_{x_{t-1|t-1}} \sum_{x_{t-2|t-2}} p(x_{t|t-1} | x_{t-1|t-1}, x_{t-2|t-2}) \cdot \\ & b(x_{t-1|t-1}) \cdot b(x_{t-2|t-2}) \end{aligned} \tag{12}$$

The notation $x_{t|t-1}$ denotes the value of the state x at time t given all observations up to time $t-1$. $b(\cdot)$ is the belief, and $p(x_{t|t-1}|x_{t-1|t-1}, x_{t-2|t-2})$ represents the transition probabilities, i.e. the dependency on the previous states. At this point the belief of the current state x_t depends only on the observations up to z_{t-1}. We include the current observation z_t via the result of the classifier c_t.

$$b(x_{t|t}) \propto \frac{p(x_{t|t-1}|c_t, z_t) p(c_t|z_t)}{p(x_{t|t-1}|z_t)} \cdot b(x_{t|t-1}), \tag{13}$$

where $p(c_t|z_t)$ is the classification result. The output of our BF is the belief of state x_t given all observations including z_t. It considers the classification and two previous states.

As already mentioned, the terrain classification is used to determine on which terrain the robot is currently driving and hence which terrain model should be used to estimate the next position. Since the robot commands required for the position estimation are given at a rate of approximately $3\,Hz$, whereas the terrain classification relies only on IMU data provided at $100\,Hz$, we implemented both parts to work independently of each other. However, it takes about a second to acquire enough data for classification, thus we reuse the last terrain model until a new classification is available. Below we will refer to the terrain models used by the GPR as *applied models* to distinguish them from the fewer classifications which were actually performed. Algorithm 1 summarizes our approach.

Algorithm 1. Dynamic Approach.

```
Input: initialized and trained GPR and GPC models
while sensor data $q_*$ available do
    get $p_*$ and $q_*$ from sensors
    if $p_*$ available and $speed > \tau$ then
        $s_*$ = FFT($p_*$)
        $t_p$ = GPC.classify($s_*$)
        $b_p$ = BF.process($t_p$)
        $model$ = selectModel($b_p$)
    end if
    $[v, \omega]$ = GPR.predict($model$,$q_*$)
    $pos$ = UKF.process($v, \omega$)
    publish($pos$)
end while
```

The algorithm requires trained GP models. If sensor data q_* is available, the algorithm predicts the velocity values using the current terrain model. If sufficient vibration data p_* is available for classification (omitting low speed data), a new classification of the terrain will be performed and the current model will be updated.

5 EXPERIMENTAL RESULTS

The Longcross (Figure 1) is an experimental platform weighing about *340 kg* with a payload capacity of at least *150 kg*. The compartment consists of carbon-fibre and is environmentally shielded. Our version is equipped with a SICK LMS 200 2D laser scanner, a Velodyne Lidar HDL-64E S2 3D laser scanner, and an Oxford Technical Solutions Ltd RT3000 combined GPS receiver and inertial unit. The software runs on a dedicated notebook with a Intel Core 2 Extreme Quad processor and *8 GB* memory.

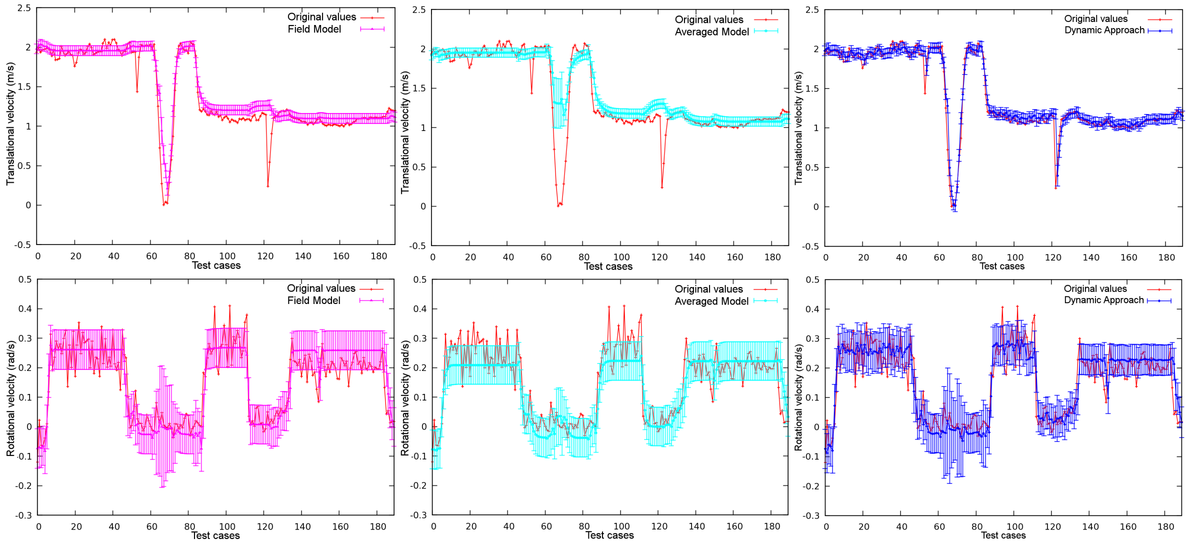


Figure 4: Estimation of the translational velocities (upper row), and of the rotational velocities (lower row) with error bars. Even if the differences between the models are not significant, they result in rather distinct trajectories.

5.1 Simple Approaches

We evaluated our approach on a test drive covering three different terrains, starting on field, continuing with grass, and ending on asphalt. The test sequence consists of 190 positions (about a 1 minute) almost evenly partitioned on the three terrains. We compared our dynamic approach to a *field model*, an *asphalt model*, a *grass model* and an *averaged model*. The first three models were trained only on the specific terrain. The *averaged model* was trained on all three terrains equally. All models used a total of 1500 training cases. The dynamic approach utilizes the three terrain specific models.

We used the mean squared error (MSE) and the standardized MSE to evaluate the quality of the regression. The SMSE is the MSE divided by the variance of the test points, thus it does not depend on the overall scaling of the values. The MSE is not applicable to measure the quality of a trajectory because errors at the beginning are propagated through the entire trajectory influencing all proceeding positions. So we introduced the mean segment distance error (MSDE). First, we split the original trajectory and the prediction in segments of fixed size. Each pair of segments is normalized to the same starting position and orientation. The distance of the resulting end positions is computed subsequently. The mean of all these distances constitutes the MSDE. We found a segment size of 10 to be a reasonable size.

The estimation results of the translational (upper row) and the rotational velocities (lower row) of the *field model*, the *averaged model* and of our dynamic approach are shown in Figure 4. The *field model* is the best of the simple models. The *averaged model* would be the model of choice if one wants to consider different terrain conditions but does not want to employ a dynamic approach.

The estimation of the translational velocities by the *averaged model* shows little or no reaction to any of the outliers. Averaging the effects of commands over several terrains involves diminishing effects of certain command-terrain combinations, hence causing inaccurate estimates. The *field model* performs best of the simple approaches, mainly because it recognizes the slowdown around the 70th test case. However, the other two outliers are not identified. The estimates of the remaining test cases are similar to the *averaged model*, yet the *field model* is slightly superior. Our dynamic approach outperforms both

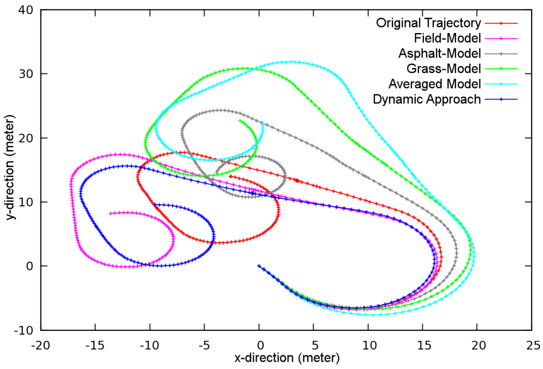


Figure 5: Predicted trajectories. Starting the models' predictions are exact but differ increasingly along the time line, leading to unlike trajectories.

Table 1: Prediction quality. The translational velocity unit is m/s, the rotational velocity unit is rad/s, and the MSDE unit is m.

Model		MSE	SMSE	MSDE$_{10}$
Field	v	0.02522	0.10588	0.42923
	ω	0.00331	0.23605	
Asphalt	v	0.07032	0.29517	0.56363
	ω	0.00533	0.38034	
Grass	v	0.05161	0.21662	0.51907
	ω	0.00365	0.26041	
Averaged	v	0.06203	0.26038	0.45478
	ω	0.00377	0.26929	
Dynamic	v	0.01181	0.04956	0.31407
	ω	0.00279	0.19917	

previous models. In contrast, it recognizes all outliers, as the right terrain model is applied most of the times. Additionally, the estimates of each test point are pretty accurate.

The estimates of the rotational velocities give a similar picture. The *averaged model* is somewhat off at the beginning, underestimating the true values. The *field model* is much more accurate. The error bars seem to be larger than in the upper row but it is due to the scale of the data. However, our approach again provides the best estimation, reacting even to minor changes in the velocity values.

The velocities are translated into positions using an UKF. The resulting trajectories of all models tested are shown in Figure 5. The previous tendencies are reflected in the quality of the trajectories. The *averaged model's* trajectory is rather inaccurate, followed by the trajectory of the *field model* which is outperformed by our dynamic approach. At the beginning the predicted tracks are close, the differences start during the first turn and increase by the time. Even though the distinction between the velocity estimates are relatively small, the impact on the trajectories are formidable. Especially the rotation values are crucial to the trajectory quality.

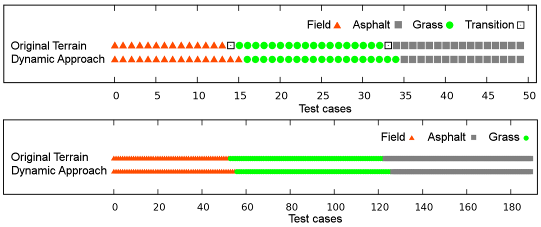


Figure 6: Classification results (upper image) and applied models (lower image). The classification matches the sequence of field (red triangle), grass (green circles), and asphalt (grey squares) with high accuracy. Transitions are not classifiable because the records contain data of two terrains.

Table 2: Classification results and applied models. F = field, A = asphalt, G = grass.

Dynamic Approach	Classification	Applied Model
Correct Terrain	96.06%	96.45%
$p(y=\mathbf{F}\|x\in\mathbf{F})$	100.00%	100.00%
$p(y=\mathbf{A}\|x\in\mathbf{A})$	93.75%	93.84%
$p(y=\mathbf{G}\|x\in\mathbf{G})$	94.44%	95.52%

We analyzed the quality of the GPR estimations of the velocities with the MSE and the SMSE. Due to the reasons mentioned above, the trajectories are evaluated by the MSDE. The results are presented in Table 1.

We also examined the classification performance. Figure 6 displays the results of the classification after applying the BF, and the terrain models used for the regression. The figures show that the classifier tends to categorize field as grass vibrations and grass records as asphalt. One problem is that the field and grass terrains are alike. At the low speed of approximately *1 m/s* the vibrations experienced on grass are close to the ones recorded on asphalt. Nonetheless, almost all false classifications are compensated by the BF.

Table 2 lists the classification quality in terms of correct classifications and applied models. Also, the classification rates are broken down into the classes giving more insight.

5.2 Dynamic Approaches

In the previous section we compared our dynamic approach to single terrain models, which by definition cannot be optimal when the robot drives on several different terrains. Therefore, we constructed another dynamic approach similar to the one presented. By doing so, we combined a support vector machine (SVM) to perform the regression of the velocities, and a k-nearest neighbor (k-NN) method for classification. We chose to combine a SVM and k-NN because of their low runtimes. The conducted experiments showed the overall runtime of the GP approach to be a serious issue.

We omitted the UKF from the system because the SVM does not provide any uncertainty values. These values are essential to the UKF in order to work properly. However, we kept the 2-step BF, even though the k-NN method usually returns simple class labels rather than class probabilities. We counted the votes for each class and divided them by the total number of neighbors k to provide class probabilities anyway. We used the *LibSVM* library (Chang and Lin, 2001)

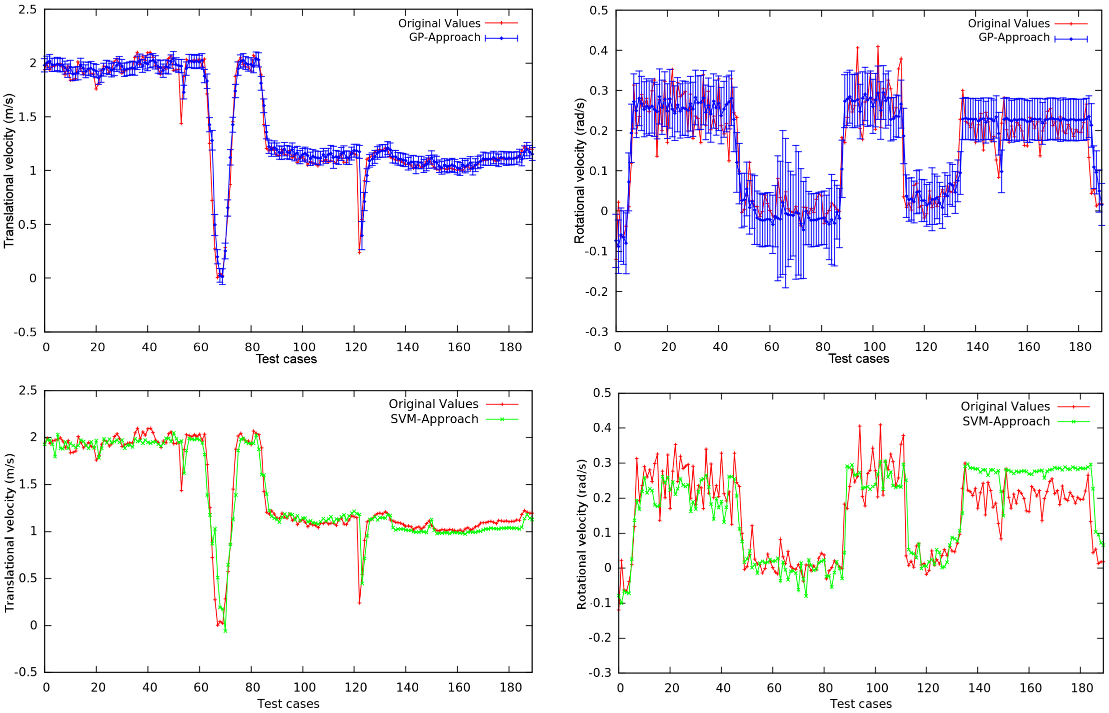


Figure 7: Comparison of the two dynamic approaches. Estimation of the translational velocities (left column), and of the rotational velocities (right column) with error bars.

with a radial basis function (RBF) as SVM kernel and found the *OpenCV* k-NN implementation with $k = 8$ neighbors to work best for vibrations.

To train the terrain models of the SVM we used the same training data as we did for the GP models. The training set of the classifiers are also unchanged.

Since this approach is able utilize several terrain models for prediction, it should be more competitive than the single terrain model approaches. To distinguish the two dynamic approaches, we will refer to the GP based system as GP-Approach and to the SVM and k-NN system as SVM-Approach.

The top row of Figure 7 shows the estimations of the translational and rotational velocities by the GP-Approach from the previous section. The prediction results of the SVM-Approach are illustrated below. Both dynamic approaches fit the original translational velocities fairly accurately and recognize all outliers. The estimation differences between the two systems are somewhat more apparent when we concentrate on the rotational velocities. The SVM-Approach underestimates the first part and overestimates the velocities towards the end. Its predictions are more agitated in contrast to the averaging characteristic of the GP estimations.

The observations are reflected in the error values of Table 3. Even though the errors of the dynamic approaches are very small - they are the smallest of all tested models - the values of the SVM-Approach are more than $\frac{1}{3}$ higher. As we will see later, this is partly due to the lower k-NN classification quality.

The SVM's underestimation of the rotational ve-

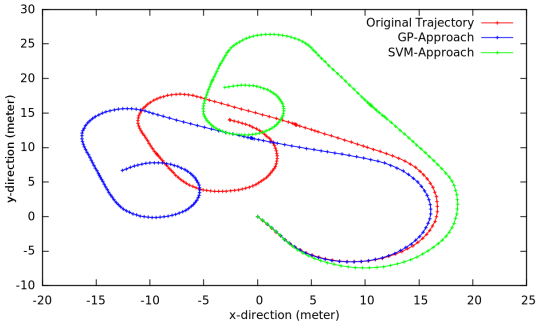


Figure 8: Predicted trajectories of the two dynamic approaches.

Table 3: Prediction quality of the two dynamic approaches. The translational velocity unit is m/s, the rotational velocity unit is rad/s, and the MSDE unit is m.

Model		**MSE**	**SMSE**	**MSDE$_{10}$**
GP Approach	v	0.01181	0.04956	0.31407
	ω	0.00279	0.19917	
SVM Approach	v	0.01923	0.08073	0.30336
	ω	0.00464	0.33089	

Table 4: Classification results of the two dynamic approaches. F = field, A = asphalt, G = grass.

Classification	SVM-Approach	GP-Approach
Correct Terrain	70.00%	96.06%
$p(y = \mathbf{F} \mid x \in \mathbf{F})$	71.43%	100.00%
$p(y = \mathbf{A} \mid x \in \mathbf{A})$	93.75%	93.75%
$p(y = \mathbf{G} \mid x \in \mathbf{G})$	55.55%	94.44%

Table 5: Applied models of the two dynamic approaches. F = field, A = asphalt, G = grass.

Applied Model	SVM-Approach	GP-Approach
Correct Terrain	75.26%	96.45%
$p(y = \mathbf{F} \mid x \in \mathbf{F})$	71.70%	100.00%
$p(y = \mathbf{A} \mid x \in \mathbf{A})$	91.04%	93.84%
$p(y = \mathbf{G} \mid x \in \mathbf{G})$	62.86%	95.52%

locity corresponds to the assumption of a less sharper turn. The green trajectory in Figure 8 shows exactly this behavior. Furthermore, the higher estimated rotational velocities at the end of the trajectory result in tighter turns. Regardless of the rather different predictive trajectories, the trajectory qualities in Table 3 are almost the same.

With respect to the classification task the GP classifier performs much better than the k-NN classifier. The sequence of classifications and applied models are compared in Figure 9. While the GP-Approach keeps the current terrain until enough evidence is present that the terrain may have changed, the SVM-Approach tends to change the terrain class more quickly. This makes the system to apply the wrong terrain model more often.

Table 4 and 5 point out that the k-NN classifier struggles to distinguish grass and field records. Leading to a performance more than 25% worse than the GP classification. Evaluating the quality of the applied models, we can see a slight improvement by about 5%, still staying way behind our approach.

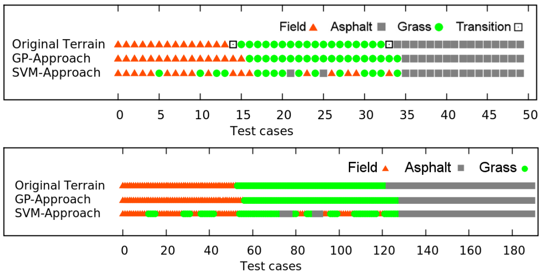


Figure 9: Comparison of the two dynamic approaches. Classification results (upper image) and applied models (lower image). Transitions are not classifiable because the records contain data of two terrains.

As stated earlier the classification quality is crucial to the performance of the overall system, since false classifications lead to the application of wrong terrain models, resulting in worse velocity estimations. However, the SVM-Approach, incorporating SVM regression and k-NN classification, is significantly faster than our GP-Approach.

6 CONCLUSIONS

In this paper we introduced a dynamic approach to estimate positions of a mobile robot. Since the terrain conditions are crucial to the robot's implementation of velocity commands, we consider the terrain for the prediction of future positions. We used GPs for the regression of translational and rotational velocities. An UKF transfers the velocity estimates into positions and propagates the uncertainties of the positions over time. The vibration affecting the robot are classified by a GPC model in order to separate different terrains. A 2-step BF is used to compensate for classification errors and to model the terrain transitions.

The prediction problem considered in this paper has several difficult properties: first, with respect to the classification task, the visual ground truth is not always accurate. Due to natural terrain types, which in themselves can be vastly different, vibrations induced by separate terrains may sometimes look alike or even show characteristics of another terrain. In a consequence, making it tricky to evaluate the the classifier's performance. Second, finding a reasonable diversity of terrains is quite challenging, which aggravates the task of determining the transition probabilities from real data.

Nevertheless, our approach proved to be more effective than simple approaches utilizing a single model for all terrains. Furthermore, our GP-Approach is competitive with other dynamic approaches using different machine learning techniques. The dynamic systems, our GP-Approach and the slightly structurally altered SVM-Approach, are both better than all simple models, proving the basic idea of our approach to be well suited to solve such prediction problems.

Future work will be to reduce the run time. Due to the use of GPs the entire systems suffers of a long run time. Several sparse algorithms for GPs exist which would help to improve this issue. Moreover, expanding the current system by additional terrains as sand or gravel would be desirable. Extending the classification records by the translational velocity could improve classification results, since it allows to learn the effects of different speeds on the vibrations.

REFERENCES

Agrawal, M. and Konolige, K. (2006). Real-time localization in outdoor environments using stereo vision and inexpensive GPS. In *International Conference on Pattern Recognition (ICPR)*, volume 3, pages 1063–1068.

Brooks, C. A., Iagnemma, K., and Dubowsky, S. (2005). Vibration-based terrain analysis for mobile robots. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3415–3420.

Burgard, W., Fox, D., Hennig, D., and Schmidt, T. (1996). Estimating the absolute position of a mobile robot using position probability grids. In *AAAI National Conference on Artificial Intelligence*, volume 2.

Chang, C.-C. and Lin, C.-J. (2001). *LIBSVM: a library for support vector machines*. Software available at *http://www.csie.ntu.edu.tw/∼cjlin/libsvm*.

Dahlkamp, H., Kaehler, A., Stavens, D., Thrun, S., and Bradski, G. R. (2006). Self-supervised monocular road detection in desert terrain. In *Robotics: Science and Systems*.

Ferris, B., Hähnel, D., and Fox, D. (2006). Gaussian processes for signal strength-based location estimation. In *Robotics: Science and Systems*.

Girard, A., Rasmussen, C. E., Quinonero-Candela, J., and Murray-Smith, R. (2003). Gaussian process priors with uncertain inputs - Application to multiple-step ahead time series forecasting. In *Advances in Neural Information Processing Systems (NIPS)*, pages 529–536.

Iagnemma, K. and Ward, C. C. (2009). Classification-based wheel slip detection and detector fusion for mobile robots on outdoor terrain. *Autonomous Robots*, 26(1):33–46.

Kapoor, A., Grauman, K., Urtasun, R., and Darrell, T. (2007). Active learning with gaussian processes for object categorization. In *IEEE International Conference on Computer Vision (ICCV)*, volume 11, pages 1–8.

Ko, J., Klein, D. J., Fox, D., and Hähnel, D. (2007a). Gaussian processes and reinforcement learning for identification and control of an autonomous blimp. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 742–747.

Ko, J., Klein, D. J., Fox, D., and Hähnel, D. (2007b). GP-UKF: Unscented Kalman filters with gaussian process prediction and observationmodels. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1901–1907.

Li-Juan, L., Hong-Ye, S., and Jian, C. (2007). Generalized predictive control with online least squares support vector machines. In *Acta Automatica Sinica (AAS)*, volume 33, pages 1182–1188.

MacKay, D. J. C. (1998). Introduction to gaussian processes. In Bishop, C. M., editor, *Neural Networks and Machine Learning*, NATO ASI Series, pages 133–166. Springer-Verlag.

Rasmussen, C. E. (2002). Combining laser range, color, and texture cues for autonomous road following. In *IEEE International Conference on Robotics and Automation (ICRA)*.

Rasmussen, C. E. and Williams, C. K. I. (2005). *Gaussian Processes for Machine Learning*. The MIT Press.

Seyr, M., Jakubek, S., and Novak, G. (2005). Neural network predictive trajectory tracking of an autonomous two-wheeled mobile robot. In *International Federation of Automatic Control (IFAC) World Congress*.

Thrun, S., Fox, D., Burgard, W., and Dellaert, F. (2000). Robust monte carlo localization for mobile robot. *Artificial Intelligence*, 128:99–141.

Urtasun, R. and Darrell, T. (2007). Discriminative gaussian process latent variable models for classification. In *International Conference on Machine Learning (ICML)*.

Ward, C. C. and Iagnemma, K. (2007). Model-based wheel slip detection for outdoor mobile robots. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2724–2729.

Weiss, C., Fröhlich, H., and Zell, A. (2006). Vibration-based terrain classification using support vector machines. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4429–4434.

Williams, C. K. I. (2002). Gaussian processes. In *The Handbook of Brain Theory and Neural Networks*. The MIT Press, 2 edition.

SELECTION OF DIFFERENT PATHS FOR DOUBLY NONHOLONOMIC MOBILE MANIPULATORS

Alicja Mazur and Elżbieta Roszkowska
Institute of Computer Engineering, Control and Robotics,
Wrocław University of Technology, ul. Janiszewskiego 11/17, 50-372 Wrocław, Poland
{*alicja.mazur, elzbieta.roszkowska*}*@pwr.wroc.pl*

Keywords: Mobile manipulator, Nonholonomic constraints, Path following, Frenet parametrization.

Abstract: This paper describes a problem of designing control laws preserving a motion along desired path for doubly nonholonomic mobile manipulators. The doubly nonholonomic mobile manipulator is composed of a mobile platform moving without slipping effect between wheels and the surface (i.e. nonholonomic platform) and an onboard manipulator equipped with special nonholonomic gears, designed by Nakamura, Chung and Sørdalen. A task for any nonholonomic subsystem (i.e. nonholonomic platform or nonholonomic manipulator) is to follow a desired path – some geometric curve parameterized by curvilinear distance from selected point. A description of the nonholonomic subsystem relative to the desired path using so-called Frenet parametrization is a basis for formulating the path following problem and designing a kinematic control algorithm.

1 INTRODUCTION

A mobile manipulator is defined as a robotic system which consists of a mobile platform, equipped with non–deformable wheels, and a manipulator mounted on the platform.

The problem with a path definition for the end-effector of the mobile manipulator is that behavior of the subsystems is unpredictable, because the same path defined in global coordinates can be executed by separate subsystems or by both of them. Sometimes, it is important to move the platform and simultaneously unload a payload: such a task is defined relative to the base of the manipulator mounted on the platform; definition relative to the end-effector is ill-conditioned. In such situation the decomposition of the task into tasks defined separately for both subsystems is more natural and convenient (Mazur and Szakiel, 2009).

In the paper we assume that a desired task can be decomposed into two separate subtasks defined for each subsystem independently: the end-effector has to follow a desired geometric path described relative to the base of manipulator (i.e. relative to the platform) and the task of the platform is to follow a desired curve lying on a plane. Such a formulation of the task makes possible successive unloading of payload transported by the mobile manipulator during the control process.

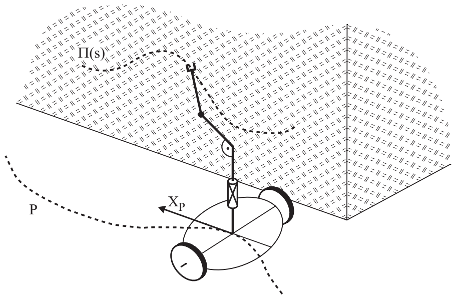


Figure 1: Decomposition of a task for doubly nonholonomic mobile manipulator: $P(s)$ – desired path for a nonholonomic platform, $\Pi(s)$ – desired path for a nonholonomic manipulator.

Taking into account the type of components mobility of mobile manipulators, there are 4 possible configurations: type (h,h) – both the platform and the manipulator holonomic, type (h,nh) – a holonomic platform with a nonholonomic manipulator, type (nh,h) – a nonholonomic platform with a holonomic manipulator, and finally type (nh,nh) – both the platform and the manipulator nonholonomic. The notion *doubly nonholonomic* manipulator was introduced in (Tchoń et al., 2004) for the type (nh,nh).

2 MATHEMATICAL MODELS OF DOUBLY NONHOLONOMIC MOBILE MANIPULATOR

In this paper we restrict our considerations to mobile manipulators of (nh, nh) type, i.e. to doubly nonholonomic objects. We will only discuss the constraints occurring in the motion of both subsystems. Nonholonomic constraints appearing in the motion of mechanical systems come from different sources. Very often they come from an assumption that a motion of the system can be treated as pure rolling of components, without slippage effect. We have taken such assumption in description of constrained motion of the considered mobile manipulator.

2.1 Nonholonomic Constraints for Wheeled Mobile Platform

Motion of the mobile platform can be described by generalized coordinates $q_m \in R^n$ and generalized velocities $\dot{q}_m \in R^n$. The wheeled mobile platform should move without slippage of its wheels. It is equivalent to an assumption that the momentary velocity at the contact point between each wheel and the motion plane is equal to zero. This assumption implies the existence of l $(l < n)$ independent nonholonomic constraints expressed in Pfaffian form

$$A(q_m)\dot{q}_m = 0, \tag{1}$$

where $A(q_m)$ is a full rank matrix of $(l \times n)$ size. Since due to (1) the platform velocity is in a null space of $A(q_m)$, it is always possible to find a vector of special auxiliary velocities $\eta \in R^m$, $m = n - l$, such that

$$\dot{q}_m = G(q_m)\eta, \tag{2}$$

where G is an $n \times m$ full rank matrix satisfying the relationship $AG = 0$. We will call the equation (2) the kinematics of the nonholonomic mobile platform.

2.2 Nonholonomic Constraints for Manipulator

A rigid manipulator can be a holonomic or a nonholonomic system – it depends on construction of its drives. In (Nakamura et al., 2001; Chung, 2004) the authors have presented a new nonholonomic mechanical gear, which could transmit velocities from the inputs to many passive joints, see Figures 2-3. The prototype of 4-link manipulator with such gears has been developed in last 1990th years. Similar nonholonomic 3-link manipulator is under construction at Poznań University of Technology, Poland (Michałek and Kozłowski, 2004). The nonholonomic constraints in the gear appear by assumption on rolling contact without slippage between balls of gear and wheels in the robot joints.

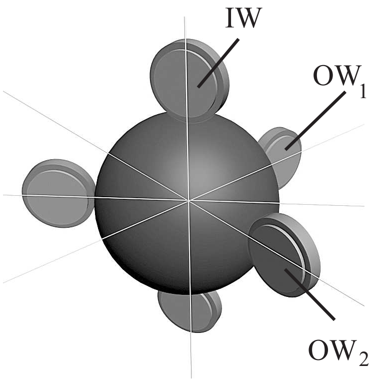


Figure 2: Schematic of the nonholonomic gear.

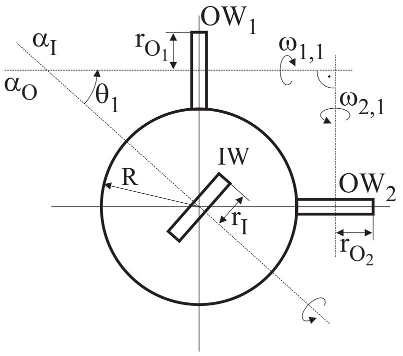


Figure 3: Nonholonomic gear seen from above.

The basic components of the gear presented in Figure 2 are a ball and three wheels – an input wheel IW and two output wheels OW_1 and OW_2. The velocity constraints of the ball are only due to point contact with the wheels. The input wheel IW is mounted in the first joint, the output wheels are mounted in the next joint. The wheel IW rotates around the fixed axis α_I with an angular velocity u_2, which plays the role of a control input. The rotating input wheel IW makes the ball rotate. The wheel OW_1 rotates around an axis α_O, which makes with the axis of the input wheel a joint angle θ_1. The angular velocity $\dot{\theta}_1 = u_1$ is the second control input for the manipulator with nonholonomic gears.

Nonholonomic constraints (the kinematics) of n-pendulum have the form

$$\dot{\theta}_1 = u_1, \tag{3}$$

$$\dot{\theta}_i = a_i \sin\theta_{i-1} \prod_{j=1}^{i-2} \cos\theta_j u_2, \quad i \in \{2, \ldots, n\}, \tag{4}$$

with positive coefficients a_i depending on gear ratios. Hypothetical manipulator with 3 links and nonholonomic gears has been presented in Figure 4.

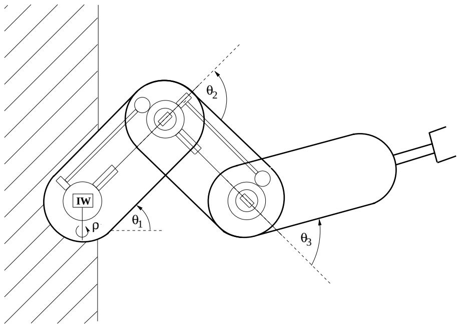


Figure 4: Schematic of 3-link nonholonomic manipulator.

It is worth to emphasize that only two inputs u_1 and u_2 are able to control many joints of manipulator equipped with gears designed by Nakamura, Chung and Sørdalen.

3 DESCRIPTION OF NONHOLONOMIC SYSTEM RELATIVE TO A GIVEN PATH

For nonholonomic systems whose workspace is planar, it is possible to describe state variables relative to global inertial frame as well as to a given path (Mazur, 2004), see Figure 5.

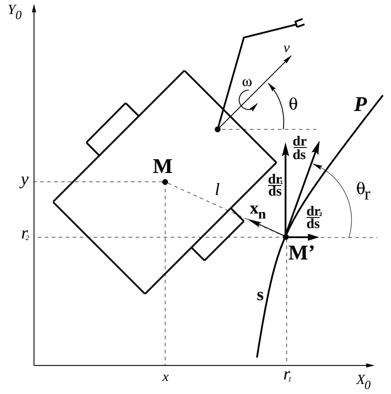


Figure 5: Illustration of path following problem for the nonholonomic platform.

The path P is characterized by a curvature $c(s)$, which is the inversion of the radius of the circle tangent to the path at a point characterized by the parameter s. Consider a moving point M and the associated Frenet frame defined on the curve P by the normal and tangent unit vectors x_n and $\frac{dr}{ds}$. The point M is the mass center of a mobile platform and M’ is the orthogonal projection of the point M on the path P.

The point M’ exists and is uniquely defined if the following conditions are satisfied (Fradkov et al., 1999):

- The curvature $c(s)$ is not bigger than $1/r_{min} > 0$.
- If the distance between the path P and the point M is smaller than r_{min}, there is a unique point on P denoted by M’.

The coordinates of the point M relative to the Frenet frame are $(0,l)$ and relative to the basic frame X_0Y_0 are equal to (x,y), where l is the distance between M and M’. A curvilinear abscissa of M’ is equal to s, where s is a distance along the path from some arbitrarily chosen point.

If we want to express the position of the point M not in coordinates (x,y) relative to inertial frame, but relative to the given path P, we should use some geometric relationships, (Mazur, 2004),

$$\dot{l} = (-\sin\theta_r \quad \cos\theta_r)\begin{pmatrix}\dot{x}\\ \dot{y}\end{pmatrix}, \tag{5}$$

$$\dot{s} = \frac{(\cos\theta_r \quad \sin\theta_r)}{1-c(s)l}\begin{pmatrix}\dot{x}\\ \dot{y}\end{pmatrix}, \tag{6}$$

where $\dot{x}$ and $\dot{y}$ are defined by nonholonomic constraints for the system (wheeled mobile platform or nonholonomic manipulator) and θ_r is a desired orientation at the point M’ on the path.

3.1 Path Following with Desired Orientation

3.1.1 Nonholonomic Mobile Platform

Posture of the mobile platform is defined not only by the position of the mass center, but by the orientation, too. For this reason, it is necessary to define the orientation tracking error equal to $\tilde{\theta} = \theta - \theta_r$. Moreover, at the point M’ the desired orientation of the platform fulfills a condition, (Samson, 1995),

$$\dot{\theta}_r = c(s)\dot{s}. \tag{7}$$

Then the coordinates

$$\xi = (l,\tilde{\theta},s)^T \tag{8}$$

i.e. the Frenet coordinates (l,s) and orientation tracking error $\tilde{\theta}$, constitute path following errors for nonholonomic mobile platform. It is worth to mention that Frenet parametrization is valid only locally, near the desired path.

As we mentioned earlier, out of Frenet coordinates it is enough to consider only l and $\tilde{\theta}$. Due to expressions (5) and (7), Frenet variables for mobile platform of $(2,0)$ class described by nonholonomic constraints

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{pmatrix} v \\ \omega \end{pmatrix} = G(q_m)\eta \tag{9}$$

can be defined as follows

$$\begin{cases} \dot{l} = v\sin\tilde{\theta}, \\ \dot{\tilde{\theta}} = \dot{\theta} - \dot{\theta}_r = \omega - v\cos\tilde{\theta}\frac{c(s)}{1-c(s)l} = w, \end{cases} \tag{10}$$

where w is a new control input for the second equation.

For the system (10) we can use many control laws, e.g. algorithm introduced in (Samson and Ait-Abderrahim, 1991),

$$\begin{cases} v = const \\ w = -k_2 l v \frac{\sin\tilde{\theta}}{\tilde{\theta}} - k_3\tilde{\theta}, \quad k_2, k_3 > 0, \end{cases} \tag{11}$$

which is asymptotically stable. It can be shown using the following Lyapunov-like function

$$V(l,\tilde{\theta}) = k_2\frac{l^2}{2} + \frac{\tilde{\theta}^2}{2} \tag{12}$$

and Barbalat lemma.

Path following with a desired orientation is very important for mobile systems, especially for mobile manipulators. It comes from the fact that it would be impossible to unload a payload if the platform had wrong orientation, i.e. it would be in a right place but back-oriented.

3.1.2 Nonholonomic Manipulator

For a nonholonomic manipulator it is possible to follow along the desired path with prescribed orientation. However, this issue has a drawback. Namely, nonholonomic manipulator has only two control inputs, therefore it is impossible to have the mentioned three parameters $(l,s,\tilde{\theta})$ under control. In such a case many authors decide to regulate only two tracking errors $(l,\tilde{\theta})$ to zero and they omit the differential equation for $\dot{s}$, because it does not matter at which point s of the desired path the mobile platform enters the desired curve $P(s)$, see (Fradkov et al., 1999) for details. Such a case of the path following problem we will call the asymptotic path following.

The Frenet parametrization can be evoked once again in the problem of path following for the planar manipulator with nonholonomic gears moving on the XZ surface.The role of the point M in Figure 5 plays a point at the end of a gripper. The orientation of the end-effector θ_m is an rotation angle of the frame associated with the gripper around –Y_b axis, which is located in the base of the manipulator. It means that the orientation of the end-effector in the planar nonholonomic n-pendulum is then equal to

$$\theta_m = \sum_{i=1}^{n}\theta_i.$$

In the considered planar nonholonomic manipulator lying in XZ-plane, relationships between velocity of the working point M expressed in Cartesian and curvilinear coordinates have the form

$$\dot{l}_m = \begin{pmatrix} -\sin\theta_{rm} & \cos\theta_{rm} \end{pmatrix} \begin{pmatrix} \dot{x} \\ \dot{z} \end{pmatrix}, \tag{13}$$

$$\dot{s} = \frac{\begin{pmatrix} \cos\theta_{rm} & \sin\theta_{rm} \end{pmatrix}}{1 - c(s)l_m} \begin{pmatrix} \dot{x} \\ \dot{z} \end{pmatrix}, \tag{14}$$

where l_m denotes distance between the point M and the path $\Pi(s)$, and θ_{rm} is the orientation of the Frenet frame in the point M'. Subscripts were introduced to distinguish Frenet variables for both subsystems of the (nh,nh) mobile manipulator.

Coordinates of the end-effector in the n-pendulum relative to its base are equal to

$$\begin{cases} x = \sum_{i=1}^{n} l_i\cos\left(\sum_{j=1}^{i}\theta_j\right), \\ z = \sum_{i=1}^{n} l_i\sin\left(\sum_{j=1}^{i}\theta_j\right). \end{cases} \tag{15}$$

Substituting time derivatives of variables (15), we obtain the following equations

$$\dot{l}_m = \sum_{i=1}^{n}\cos\left(\theta_{rm} - \sum_{j=1}^{i}\theta_j\right) l_i \sum_{k=1}^{i}\dot{\theta}_k, \tag{16}$$

$$\begin{aligned} \dot{\tilde{\theta}}_m &= \dot{\theta}_m - c(s)\dot{s} = \sum_{i=1}^{n}\dot{\theta}_i - \frac{c(s)}{1-c(s)l_m}\cdot \\ &\cdot\sum_{i=1}^{n}\sin\left(\theta_{rm} - \sum_{j=1}^{i}\theta_j\right) l_i \sum_{k=1}^{i}\dot{\theta}_k. \end{aligned} \tag{17}$$

Using the kinematics of the nonholonomic manipulator given by (3)-(4), the equations (16) and (17) can be expressed in the matrix form as follows

$$\begin{aligned} \dot{\xi}_m &= \begin{pmatrix} \dot{l}_m \\ \dot{\tilde{\theta}}_m \end{pmatrix} = H(q_r,\xi_m)\begin{pmatrix} \dot{\theta}_1 \\ \vdots \\ \dot{\theta}_n \end{pmatrix} = \\ &= H(q_r,\xi_m)G_2(q_r)u = K_l(q_r,\xi_m)u. \end{aligned} \tag{18}$$

Matrix $K_l(q_r,\xi_m)$ fulfills the regularity condition (i.e. it is invertible) if some configurations, which imply the matrix singularity, are excluded from a set of possibly achieved poses of the nonholonomic manipulator.

For nonholonomic 3-pendulum matrix $K_l(q_r,\xi_m)$ has the form

$$K_l(q_r,\xi_m)=\begin{bmatrix} K_{l11} & K_{l12} \\ K_{l21} & K_{l22}\end{bmatrix},$$

with elements defined below

$$\begin{aligned}
K_{l11} &= \sum_{i=1}^{3} l_i \cos(\theta_{rm}-\sum_{j=1}^{i}\theta_i),\\
K_{l12} &= a_2 s_1 \sum_{i=2}^{3} l_i \cos(\theta_{rm}-\sum_{j=1}^{i}\theta_i)+\\
&\quad +a_3 s_2 c_1 l_3 \cos(\theta_{rm}-\sum_{j=1}^{3}\theta_i),\\
K_{l21} &= 1-\frac{c(s)}{1-c(s)l_m}\sum_{i=1}^{3} l_i \sin(\theta_{rm}-\sum_{j=1}^{i}\theta_i),\\
K_{l22} &= a_2 s_1 [1-\frac{c(s)}{1-c(s)l_m}\sum_{i=2}^{3} l_i \sin(\theta_{rm}-\sum_{j=1}^{i}\theta_i)]\\
&\quad +a_3 s_2 c_1 [1-\frac{c(s)}{1-c(s)l_m} l_3 \sin(\theta_{rm}-\sum_{j=1}^{3}\theta_i)].
\end{aligned}$$

Note that Frenet transformation is valid only locally, i.e. $l_m(0)<r_{min}$, where r_{min} is an inversion of maximal curvature c_{max} of the manipulator path $\Pi(s)$, therefore nominators of all fractions are well defined. In turn, the nonholonomic planar 3-pendulum cannot achieve angles equal to $\theta_1,\theta_2=0,\pm\pi$. Moreover, singularities in K_l matrix appear for $\sin(\theta_{rm}-\theta_1)=\sin(\theta_{rm}-\theta_1-\theta_2)=\sin(\theta_{rm}-\theta_1-\theta_2-\theta_3)=0$.

For the regular matrix K_l, the following control signals guaranteeing a convergence of tracking errors to zero for pure kinematic constraints can be proposed

$$u_r=-K_l^{-1}(q_r,\xi_m)\Lambda\xi_m, \qquad \Lambda=\Lambda^T>0. \quad (19)$$

It is easy to observe that the system (18) with closed-loop of the feedback signal (19) has a form

$$\dot{\xi}_m+\Lambda\xi_m=0,$$

i.e. it is asymptotically stable.

3.2 Path Following without Desired Orientation

The manipulator with gears designed by Nakamura, Chung and Sørdalen has two control inputs. It means that only two parameters can be regulated during the path following process. If we mean that the orientation of the end-effector of such manipulator is not very important, it is possible to control other Frenet parameters, e.g. l_m – distance error from the desired path and curvilinear length s of the path.

In such a case the following differential equations

$$\begin{aligned}
\dot{l}_m &= \sum_{i=1}^{n}\cos\left(\theta_{rm}-\sum_{j=1}^{i}\theta_j\right) l_i \sum_{k=1}^{i}\dot{\theta}_k,\\
\dot{s} &= \frac{1}{1-c(s)l_m}\cdot\sum_{i=1}^{n}\sin\left(\theta_{rm}-\sum_{j=1}^{i}\theta_j\right) l_i \sum_{k=1}^{i}\dot{\theta}_k
\end{aligned}$$

have to be considered. Similarly to (16) and (17), using the kinematics of the nonholonomic manipulator (3)–(4), these equations can be expressed in the matrix form as follows

$$\begin{aligned}
\begin{pmatrix}\dot{l}_m\\ \dot{s}\end{pmatrix} &= H(q_r,\xi_m)\begin{pmatrix}\dot{\theta}_1\\ \vdots\\ \dot{\theta}_n\end{pmatrix}=H(q_r,\xi_m)G_2(q_r)u\\
&= K_s(q_r,\xi_m)u. \qquad (20)
\end{aligned}$$

Matrix $K_s(q_r,\xi_m)$ fulfills the regularity condition (i.e. it is invertible) if some configurations, which imply the matrix singularity, are excluded from a set of possibly achieved poses of the nonholonomic manipulator.

For nonholonomic 3-pendulum matrix $K_s(q_r,\xi_m)$ has the form

$$K_s(q_r,\xi_m)=\begin{bmatrix} K_{s11} & K_{s12} \\ K_{s21} & K_{s22}\end{bmatrix},$$

with elements defined below

$$\begin{aligned}
K_{s11} &= \sum_{i=1}^{3} l_i \cos(\theta_{rm}-\sum_{j=1}^{i}\theta_i),\\
K_{s12} &= a_2 s_1 \sum_{i=2}^{3} l_i \cos(\theta_{rm}-\sum_{j=1}^{i}\theta_i)+\\
&\quad +a_3 s_2 c_1 l_3 \cos(\theta_{rm}-\sum_{j=1}^{3}\theta_i),\\
K_{s21} &= \frac{c(s)}{1-c(s)l_m}\sum_{i=1}^{3} l_i \sin(\theta_{rm}-\sum_{j=1}^{i}\theta_i),\\
K_{s22} &= \frac{a_2 s_1 c(s)}{1-c(s)l_m}\sum_{i=2}^{3} l_i \sin(\theta_{rm}-\sum_{j=1}^{i}\theta_i)\\
&\quad +\frac{a_3 s_2 c_1 c(s)}{1-c(s)l_m} l_3 \sin(\theta_{rm}-\sum_{j=1}^{3}\theta_i).
\end{aligned}$$

Singular configurations of nonholonomic 3-pendulum for the matrix K_s are equal to configurations

$$\sin(\theta_1)=\sin(\theta_1-\theta_2)=\sin(\theta_1-\theta_2-\theta_3)=0.$$

If the following control law is applied

$$u_r=-K_s^{-1}(q_r,\xi_m)v, \qquad (21)$$

where $v \in R^2$ is a new input to the system (20), then we obtain the decoupled and linearized control system of the form

$$\begin{pmatrix} \dot{l}_m \\ \dot{s} \end{pmatrix} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}. \qquad (22)$$

Now it is possible to control each variable separately. For instance, if we want to move along the desired path, not only to converge to this path, it seems to be a good idea to preserve $\dot{s} \neq 0$. Possible choice of the control algorithm for the decoupled system (22) is

$$v_1 = -\Lambda l_m, \qquad v_2 = const, \qquad \Lambda > 0.$$

Such control algorithm guarantees the motion along the geometrical curve with constant velocity and, simultaneously, the convergence of the distance tracking error l_m to 0.

4 SIMULATIONS

As an object of a simulation study we have chosen a planar vertical 3-pendulum with nonholonomic gears mounted on a unicycle.
The desired path for the manipulator (a circle) was selected as

$$\begin{aligned} \Pi_1(s) &= 0.25\cos 4s + 1\ [m], \\ \Pi_2(s) &= -0.25\sin 4s + 0.6\ [m], \end{aligned}$$

and the desired path for the mobile platform was a straight line

$$P(s): \qquad x(s) = \frac{\sqrt{2}}{2}s\ [m], \qquad y(s) = \frac{\sqrt{2}}{2}\ [m].$$

The initial configuration of the manipulator was equal to $(\theta_1, \theta_2, \theta_3)(0)$= $(0, 0.6732, -\pi/3)$ and initial posture of the platform was selected as $(x, y, \theta)(0) = (0, 2, 3\pi/4)$.

Tracking of the desired path for the mobile platform has been presented in Figures 6–8. Parameters of the Samson & Ait-Abderrahim algorithm were selected as $v = 1$, $k_2 = 0.1$ and $k_3 = 1$.

Tracking of the desired path with prescribed orientation for the nonholonomic manipulator 3R has been presented in Figures 9–11.
Tracking of the desired path without preserved orientation for the same manipulator has been presented in Figures 12–13.

5 CONCLUSIONS

In the paper the problem of defining the path for doubly nonholonomic mobile manipulators has been considered. We have proposed a new approach to the path

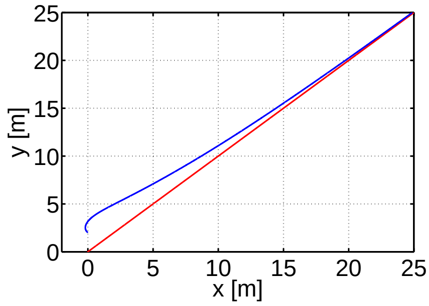


Figure 6: Path tracking for the mobile platform – XY plot.

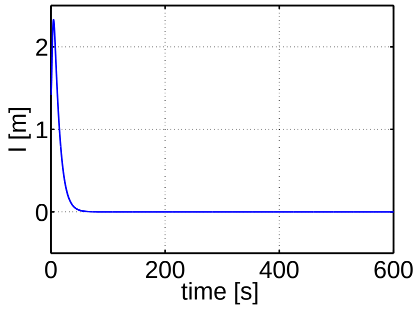


Figure 7: Path tracking for the mobile platform – distance error l.

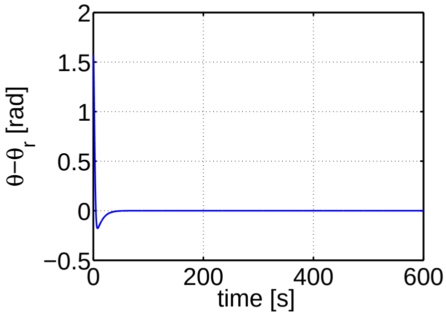


Figure 8: Path tracking for the mobile platform – orientation error $\tilde{\theta}$.

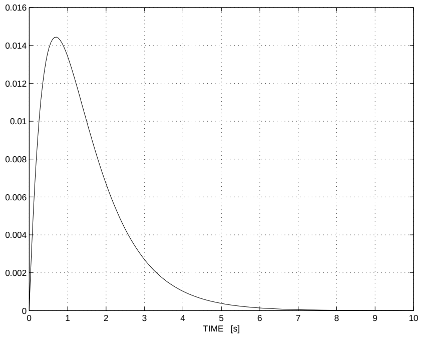


Figure 9: Path tracking for the 3-pendulum: an error of x coordinate.

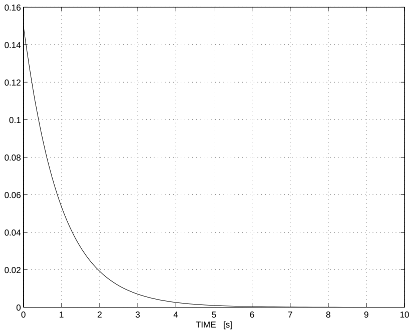


Figure 10: Path tracking for the 3-pendulum: an error of z coordinate.

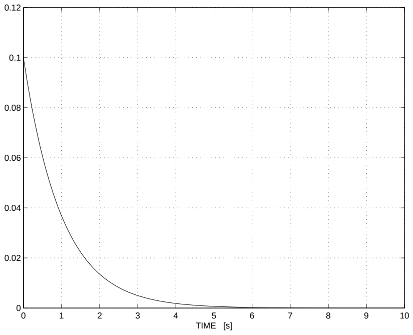


Figure 11: Path tracking for the 3-pendulum: an error of orientation $\tilde{\theta}$.

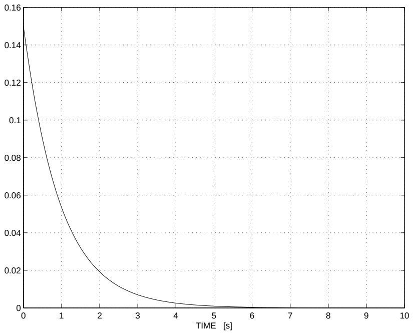


Figure 12: Path tracking for the 3-pendulum – XZ plot.

as a geometric curve defined with the orientation or not. Path following problem with prescibed orientation is very important for mobile systems, especially for mobile manipulators – it results from the fact that it is impossible to realize a task, namely unload a payload if the platform has wrong orientation during the regulation process.

In turn, for nonholonomic manipulator the desired path need not be defined with orientation. In such a case a new approach to the path following problem has been presented in the paper. A new control algorithm, guaranteeing not only asymptotic convergence to the desired path but simultaneously the motion along the path with nonzero velocity, has been introduced. It is possible to define another kinematic control algorithms with specific properties using Frenet parametrization.

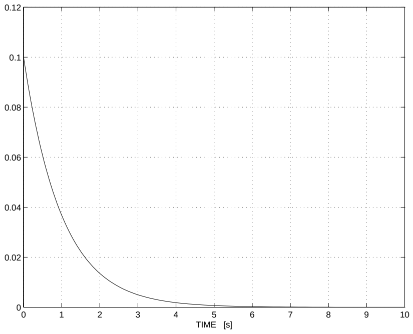


Figure 13: Path tracking for the 3-pendulum: a distance error l_m.

REFERENCES

Chung, W. (2004). Design of the nonholonomic manipulator. *Springer Tracts in Advanced Robotics*, 13:17–29.

Fradkov, A., Miroshnik, I., and Nikiforov, V. (1999). *Nonlinear and Adaptive Control of Complex Systems*. Kluwer Academic Publishers, Dordrecht.

Mazur, A. (2004). Hybrid adaptive control laws solving a path following problem for nonholonomic mobile manipulators. *International Journal of Control*, 77(15):1297–1306.

Mazur, A. and Szakiel, D. (2009). On path following control of nonholonomic mobile manipulators. *International Journal of Applied Mathematics and Computer Science*, 19(4):561–574.

Michałek, M. and Kozłowski, K. (2004). Tracking controller with vector field orientation for 3-d nonholonomic manipulator. In *Proceedings of 4th int. Workshop on Robot Motion and Control*, pages 181–191, Puszczykowo, Poland.

Nakamura, Y., Chung, W., and Sørdalen, O. J. (2001). Design and control of the nonholonomic manipulator. *IEEE Transactions on Robotics and Automation*, 17(1):48–59.

Samson, C. (1995). Control of chained systems - application to path following and time-varying point-stabilization of mobile robots. *IEEE Transactions on Automatic Control*, 40(1):147–158.

Samson, C. and Ait-Abderrahim, K. (1991). Feedback control of a nonholonomic wheeled cart in cartesian space. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 1136–1141, Sacramento.

Tchoń, K., Jakubiak, J., and Zadarnowska, K. (2004). Doubly nonholonomic mobile manipulators. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 4590–4595, New Orleans.

ON REAL-TIME WHOLE-BODY HUMAN TO HUMANOID MOTION TRANSFER

Francisco-Javier Montecillo-Puente, Manish N. Sreenivasa and Jean-Paul Laumond
CNRS, LAAS, 7 avenue du colonel Roche, F-31077 Toulouse, France
Université de Toulouse, UPS, INSA, INP, ISAE, LAAS, F-31077 Toulouse, France
{fmonteci, manu, jpl}@laas.fr

Keywords: Humanoids, HRP2, Human motion, Inverse kinematics, Neuroscience.

Abstract: We present a framework for online imitation of human motion by the humanoid robot HRP-2. We introduce a representation of human motion, the Humanoid-Normalized model, and a Center of Mass (CoM) anticipation model to prepare the robot in case the human lifts his/her foot. Our proposed motion representation codifies operational space and geometric information. Whole body robot motion is computed using a task-based prioritized inverse kinematics solver. By setting the human motion model as the target, and giving the maintenance of robot CoM a high priority, we can achieve a large range of motion imitation. We present two scenarios of motion imitation, first where the humanoid mimics a dancing motion of the human, and second where it balances on one foot. Our results show that we can effectively transfer a large range of motion from the human to the humanoid. We also evaluate the tracking errors between the original and imitated motion, and consider the restrictions on the range of transferable human motions using this approach.

1 INTRODUCTION

Since the inception of humanoid robots several approaches have been developed, aiming at enabling autonomous and intelligent behavior of the robot. The general goal behind this idea is to develop humanoids that, in the natural environment of the humans and being among humans, can exhibit complex behavior. The humanoid and the human share a common anthropomorphic structure, and recent studies have exploited this to directly transfer motion from humans to humanoids. However, several obstacles imped the direct transfer of motion like technological incompatibility of the humanoid robot, fragility, slow (relative to human) motion speed, as well as a limited range of motion.

There are several ways to approach the motion transfer problem. In computer graphics, studies have looked at “motion retargeting”, where human motion is transferred from one virtual actor to another (Multon et al., 2008)(Chois and Ko, 2000). Machine learning approaches have also been developed to generate humanoid motion by observing and learning from a human teacher (Schaal et al., 2003) (Shon et al., 2005) (Takano et al., 2007). Another method is to optimize the recorded human motion while considering humanoid kinematics and dynamics (Suleiman et al., 2008) (Ruchanurucks et al., 2006). For our purpose we broadly classify motion imitation studies into offline and online methods. Offline methods have the advantage of having ample processing time and hence the human motion can be modified to fit the humanoid’s limitations. While this allows for gen-

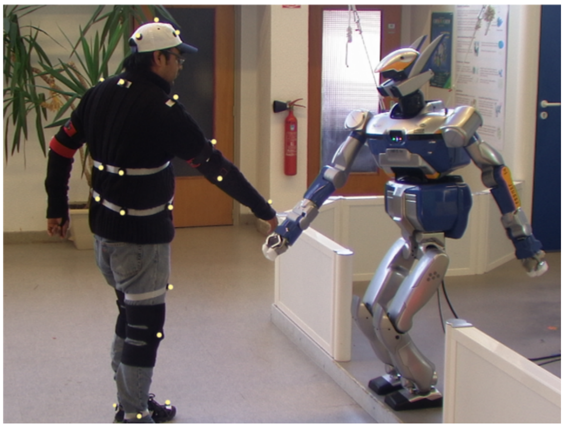


Figure 1: Picture of a human performer extending a handshake and the humanoid robot HRP-2 imitating the gesture. HRP-2 is a 30 Dof, 58 kg, 1.54 m tall humanoid robot manufactured by Kawada Industries, Japan. The human motion was tracked using reflective motion markers and transferred in real-time to the humanoid.

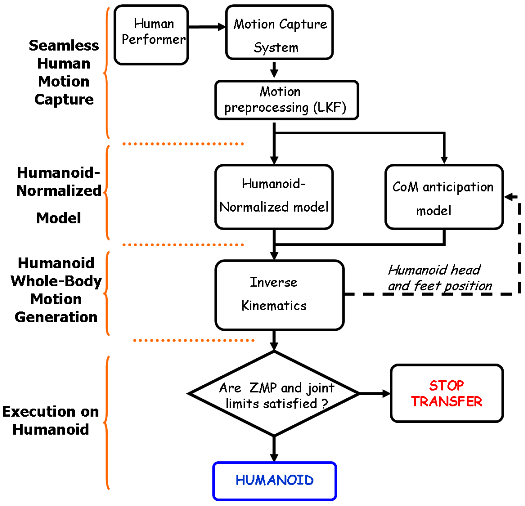

Seamless Human Motion Capture
Human Performer
Motion Capture System
Motion preprocessing (LKF)
Humanoid-Normalized Model
Humanoid-Normalized model
CoM anticipation model
Humanoid Whole-Body Motion Generation
Inverse Kinematics
Humanoid head and feet position
Execution on Humanoid
Are ZMP and joint limits satisfied ?
STOP TRANSFER
HUMANOID

Figure 2: Organization of the algorithm to enable real-time motion transfer from the human performer to the humanoid robot.

eration of smooth and optimized motion of the humanoid, these solutions can rarely be applied to online motion transfer, often due to computational delays during the optimization stage. Thus, online transfer of human motion to a humanoid robot provides a very different set of challenges compared to its offline counterpart, and has been the focus of recent research (Dariush et al., 2008a) (Dariush et al., 2008b) (Yamane and Hodgins, 2009).

We explore this aspect of real-time human-humanoid motion transfer by developing a framework that allows us to transfer a large range of human motions to the humanoid, including balancing on one foot. The challenges in this task include reliably recording human motion, morphing the human data such that it can be applied on a humanoid of a different size and joint structure, and, generating the resulting humanoid motion in real-time while maintaining balance.

1.1 Literature Review

1.1.1 Offline Motion Transfer

First we examine the work by Nakaoka et. al in 2005 (Nakaoka et al., 2005a), that enabled the humanoid robot HRP2 to execute the famous and visually striking, traditional Aizu-Bandaisan Japanese dance. Upper body motion for the HRP1s robot was generated using motion capture markers on human dancers and inverse kinematics. The leg motion of the dancers was analyzed and extracted as motion primitives (Nakaoka et al., 2005b). The upper and lower body motions were then coupled and modified such that the robot motion satisfied the dynamic stability criteria, the Zero Moment Point (ZMP) (Vukobratovic and Stepanenko, 1972). This method was implemented offline and required several runs of the analytical process to reach a viable solution. However, this method is not be easily applicable to a range of motions due to the extensive tuning and re-calculation required. As a more general framework, Ude et. al proposed to solve a large scale optimization problem to generate joint trajectories for a DB robot (Ude et al., 2004). Joint angle trajectories of human motion were computed by embedding a scalable kinematic structure to the human body motion. This was also solved as an offline process because of the optimization computational load. Robot balance was not taken into consideration.

In the study by Ruchanurucks et al. (Ruchanurucks et al., 2006), a non-linear optimization process was solved subject to joint limits, autocollision, velocity limits and force limits constraints. To increase the convergence speed they parametrized the motion by B-splines. In Suleiman et. al (Suleiman et al., 2008), first the joint motion data was scaled into the humanoid robot's joints. Then, an optimization problem was solved to fit this motion to the robot structure and its physical limits, keeping in mind the analytical gradient of the dynamic model. Among other approaches, studies have also used machine learning algorithms to imitate human motion (Shon et al., 2005). The idea here was to generate a low dimensional latent space to map a model variable from the robot motion to human motion, and vice-versa. In order to generate the final stable motion, extensive training was required using pairs of human and robot motion.

In most of these studies the humanoid is constrained to maintain both its feet on the ground. Even in the approaches that allowed feet motion, for example (Suleiman et al., 2008) and (Ruchanurucks et al., 2006), the timing of the foot lift-off has to be predefined which results in a rigid range of motion.

1.1.2 Online Methods

In the context of online transfer of human motion to humanoid robots or even virtual avatars in animation, the means used to capture human motion plays a very important role. One way is the use of motion capture technology. These systems represent human motion by directly tracking the position of infra-red markers attached to the human body or attaching a skeleton to these markers. However, some recent studies have also implemented markerless tracking where human motion is reconstructed by using video cameras.

First, we consider the study by Dariush et al. (Dariush et al., 2008b), where the authors developed a methodology to retarget human motion data to the humanoid robot ASIMO. Human motion was captured using a markerless pose tracking system. Here upper body motion was considered by mapping the cartesian positions of the waist, wrists, elbows, shoulder and neck. The corresponding joint motion on the humanoid was calculated using inverse kinematics, while respecting the individual joint limits. In this case, a seperate balance controller was used to move the legs in order to compensate for the retargeted motion of the upper body. In a later study, the authors used a learning approach to pre-generate knowledge about a number of human postures (Dariush et al., 2008a). During the actual motion retargeting, head and torso motion was monitored and the template closest from the ones learned was assigned. The arms were analyzed as 3D blobs and their position estimated. From this data the 3D features for head, shoulder, waist, elbows and wrists were localized. Using inverse kinematics and the balance controller, the motion was then played on the humanoid robot.

Using a different approach, Yamane et al. (Yamane and Hodgins, 2009), simultaneously tracked motion capture data using a balance controller and a tracking controller. Tha balance controller was a linear quadratic regulator designed for an inverted pendulum model. The tracking controller computed joint torques to minimize the difference from the desired human capture data while considering full-body dynamics. The resulting motion was retargeted on the humanoid in simulation.

1.2 Our Contribution

We present an alternative way to transfer human motion data to our humanoid robot HRP2.

- Taking inspiration from computer animation studies we extended the idea of the normalized skeleton (Multon et al., 2008) and developed a "Humanoid-Normalized model" on which filtered motion capture data can be retargeted online. To do this we first devise a method to reliably record human motion by using Kalman filters to fill gaps (due to occlusions) in capture data. Rather than directly using cartesian positions like in (Dariush et al., 2008a) (Dariush et al., 2008b), or joint angles (Yamane and Hodgins, 2009), we also encode the orientation of the postures by attaching virtual planes to sets of human motion marker points. These virtual planes define the orientation of the important joints in the human, like the head, chest, arms and waist. We propose a Humanoid-Normalized model that consists of a combination of the positions of the extremities and the normals to the virtual planes. The motion of this model is used to drive the humanoid robot via a task based inverse kinematics solver.

- In order to achieve single foot support phases we also introduce the original idea of an anticipation model, motivated by results in human neuroscience. This model serves the purpose of preparing the humanoid to stand on one foot by taking into account previous motion of the human head, and the humanoid's Center of Mass (CoM). Figure 2 shows the organization of the overall algorithm.

We evaluate the results from our experiments and compare them to those reported in literature. In the following sections we describe in detail 1) The capture of human motion capture data, 2) Application to the Humanoid-Normalized model, 3) Generating humanoid motion using inverse kinematics 4) The CoM Anticipation model 5) Experiments on our humanoid robot, HRP2 and finally 6) Discussion of our results as well as future perspectives.

2 SEAMLESS HUMAN MOTION CAPTURE

Human motion was recorded in a tracking hall equipped with 10 infra-red tracking cameras (MotionAnalysis, CA, USA). The system is capable of tracking the position of markers within a 5x5 m space within an accuracy of 1mm, at a data rate of 100 Hz. The human to be tracked wore 41 reflective markers firmly attached to their body using velcro straps, or tape (see Figure 1 and 3). In some cases there was the possibility of loss of marker data due to self-occlusions. This is very detrimental to the imitation algorithm since the humanoid model depends on continous human motion data. To solve this, we implemented a linear Kalman filter that estimates the position of markers that are lost for short durations of time (less than 0.5 seconds). If a marker is lost for periods longer than 0.5 seconds, we assume the model to be unrecoverable and initiate an emergency stop. The latency between capture of marker data and the kalman filtered data is about 30 ms. This filtered data is then applied to the Humanoid-Normalized model.

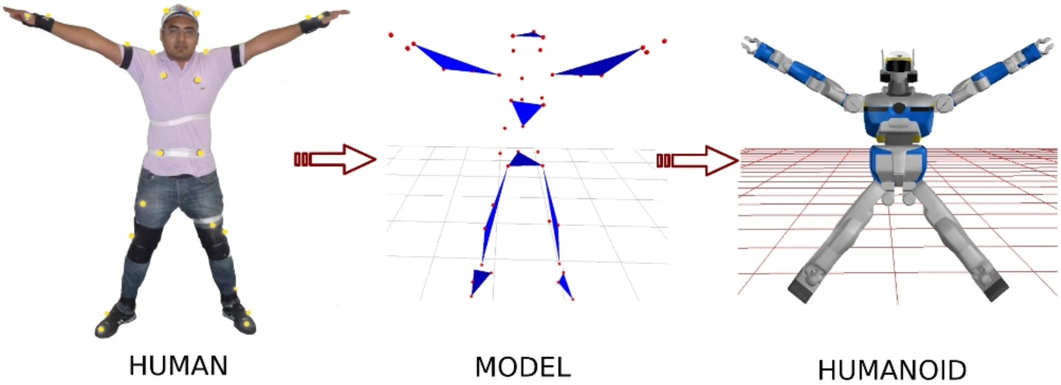


Figure 3: Human→Humanoid Normalized Model→Humanoid. Motion capture position data from the human is transferred to the normalized model and associated with the planes and humanoid joints. The motion of these planes and joints drives the humanoid motion.

3 HUMANOID-NORMALIZED MODEL

3.1 Why a Normalized Model?

Multon et al. (Multon et al., 2008) proposed the use of a normalized skeleton, which acts as an intermediate entity to map motion between two different digital actors. The generation and adaptation of motion was done by assuming that there exists planes formed by the arms and legs. The normalized skeleton was represented by normalized segments, limbs of variable lengths and the spine. Humans come in different sizes and shapes, and most adults are bigger than our humanoid HRP2. HRP2 also has two additional DoF's in its chest (pitch and yaw) and this requires special treatment unlike that for the humanoid ASIMO, used in (Dariush et al., 2008a), where the torso is one single joint. There is thus a need to extract the relevant characteristics of human motion, and define it in a form that is suitable for application on humanoids. We extended the idea presented by Multon and colleagues by defining planes for not only the arms, but also the head, chest and waist. The planes are attached to the human body in such a way that they can be directly used to define geometric tasks in the prioritized inverse kinematics solver. A combination of the orientation of these planes, and the position of the extremities (head and wrists) form the Humanoid-Normalized model (henceforth HN Model).

3.2 Components of Humanoid-normalized Model

The physical model of the upper body was built from the cartesian positions of 22 markers on the human Figure 3. Before the model is computed we scale the marker positions according to a modified version of a standard scaling algorithm (Boulic et al., 2009). First, we start with the head. Three markers on the head were used to form the head virtual plane. The center of these markers were considered in the HN model. The normal to the plane defined by these markers was computed. In the HN model we represent the orientation of the head as the orthogonal to this normal vector. For the head, the plane normal is computed as,

$$N_{head} = V_0 \times V_1 \tag{1}$$

where,

$$V_0 = \frac{p_0 - p_1}{||p_0 - p_1||}$$
$$V_1 = \frac{p_0 - p_2}{||p_0 - p_2||}$$

p_0, p_1, p_2 are the markers associated to the head virtual plane.

Similarly, chest and waist virtual planes were constructed using the relevant markers such that their normals were approximately in the sagittal direction (see Figure 3 for illustration of markers used). Virtual planes for the arms were constructed using markers on the shoulder, elbow and wrist. Instead of using the positions of all these markers, the arm posture is represented by the normal to this plane and the wrist position. We chose this representation because of the difficulty to directly map the cartesian position of all these markers to the robot structure. In addition to upper body characteristics, we also included the position of the feet in the HN model.

Thus, overall the Humanoid-Normalized model is expressed by the following set of geometric properties:

$$[P_h, V_h, N_c, V_w, P_{lh}, N_{la}, P_{rh}, N_{ra}, P_{lf}, P_{rf}] \tag{2}$$

where P_h is a point representing the position of the head, V_h is a vector representing the orientation of the head, N_c is a vector representing the normal of the chest plane, V_w is a vector representing the orientation of the waist, P_{lh} is a point representing the position of the left hand, N_{la} is a vector representing the normal to the left arm plane, P_{rh} is a point representing the position of the right hand, N_{ra} is a vector representing the normal to the right arm plane, P_{lf} is a point representing the position of the left foot, P_{rf} is a point representing the position of the right foot.

4 HUMANOID INVERSE KINEMATICS

The HN model explained in the previous section allows us to retarget human motion into a form suitable for transfer onto the humanoid. To generate motion of the humanoid we used a task-based inverse kinematics solver (Nakamura, 1991).

4.1 Prioritized Inverse Kinematics

Forward kinematics expresses the relationship between the variation of the joint parameters δq and the corresponding displacement δx in the operational space (Nakamura, 1991). This is given by,

$$\delta x = J\delta q \tag{3}$$

where J is the $m \times n$ jacobian matrix, m being the dimension of the task, and, n is the number of degrees of freedom. The inverse kinematics model determines the joint variation that produces an expected displacement, which is obtained by solving the system in Eq. 3. However, for the humanoid robot we have $m < n$, thus this linear system is under-constrained. In this case, all the solutions of the system can be written as:

$$\delta q = J^{\#}\delta q + (I - J^{\#}J)z \tag{4}$$

where $J^{\#} = J^T(JJ^T)^{-1}$ is the pseudo inverse of J, I is the $n \times n$ identity matrix, $(I - J^{\#}J)$ is the null-space projector of J, and z is an n-dimensional arbitrary vector. The first term on the right side of equation 4 is a generic solution and z in the second term can be used to satisfy additional constraints without modifying δx.

The generic expressions for solving n tasks with descending order of priorities are (Siciliano and Slotine, 1991):

$$\begin{aligned} N_0 &= I \\ \hat{J}_i &= J_i N_{i-1} \\ N_i &= I - \hat{J}_i^{\#}\hat{J}_i \\ \delta q_{i+1} &= \hat{J}_{i+1}^{\#}(\delta x_{i+1} - J_{i+1}\delta q_i) \end{aligned}$$

where N_i is the projector associated with the i^{th} task, J_i is the jacobian of the i^{th} task, I is the $n \times n$ identity matrix. The update δq_i is iteratively computed for all tasks.

4.2 Whole-body Motion Generation

The position for each joint is generated from a prioritized stack of tasks (Yoshida et al., 2006), which is solved using the inverse kinematics formulation expressed above. Each property of the HN model representation is used as the target input for the tasks. Our task stack was defined as (in decreasing priority):

1. Homogenous transformation task for each feet, i.e. both position and orientation are fixed,
2. Position task for Center of Mass (CoM) projection (X and Y positions),
3. Position task for the head,
4. Homogenous transformation task for the left wrist,
5. Homogenous transformation task for the right wrist,
6. Orientation vector task for the chest,
7. Orientation vector task for the waist,
8. Orientation vector task for the head.

We use four kinds of tasks: position task, orientation vector task, homogenous transformation task and a CoM task. As examples, we define in more detail the task construction for the head and the arms. The CoM task is detailed in the next section. The position task of the head $f_h(\theta)$ is defined as,

$$f_h(\theta) = P_h^t - P_h(\theta)$$

where P_h^t is the target of the task given by the position of the head in the HN model representation. $P_h(\theta)$ is the position of the humanoid head expressed as a function of the robot dof's θ.

For the orientation vector task of the head $f_h(\theta)$ we have

$$f_h(\theta) = V_h^t \times V_h(\theta) \tag{5}$$

where V_h^t is the target head direction, which is given by the head orientation vector in the HN model. And $V_h(\theta)$ is the corresponding vector of the humanoid head, as a function of the robot dof's θ. The orientation vector tasks for the chest and waist are defined in a similar way.

For each arm a homogeneous transformation task is constructed for the wrist joint. The target transformation is constructed from two properties, wrists

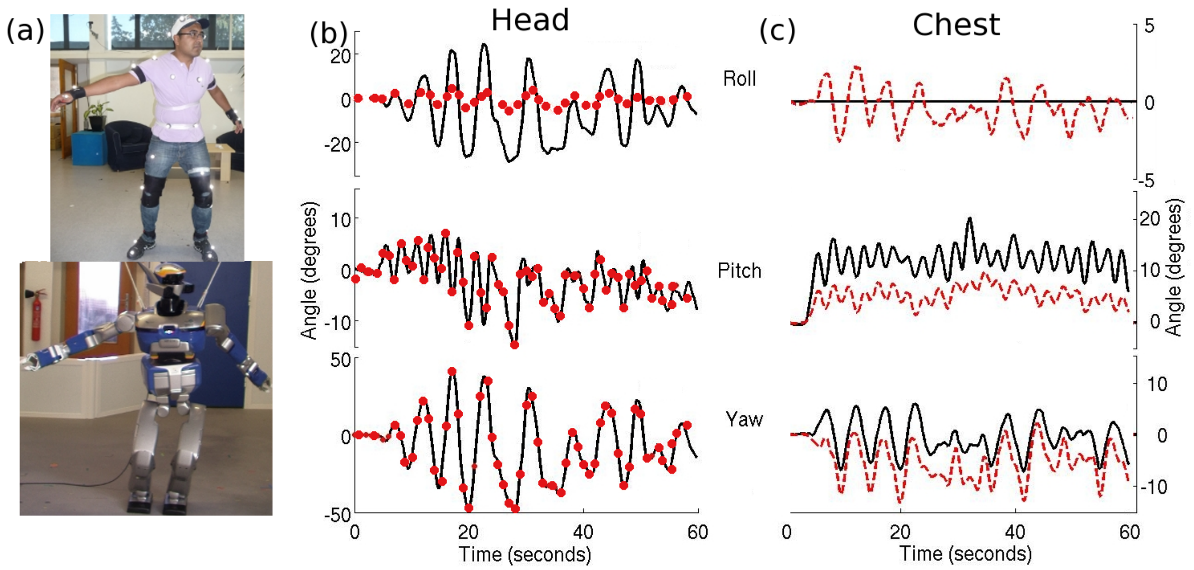


Figure 4: Scenario 1. (a) Snapshot of human dancing and its imitation by HRP2 (b) Roll, pitch and yaw angles of the head joint during the dancing motion. Solid black line indicates the angles of the HN model, this was the target the humanoid had to follow. Red circles indicate the corresponding angular value on HRP2 (c) Chest angles of the HN model (solid black line) and the corresponding angles on the humanoid (dashed red line).

position and the normal to the HN model's arm plane. The target rotation matrix in the homogenous transformation is computed as,

$$R^t = [N_{arm} \times V \;\; N_{arm} \;\; V] \tag{6}$$

where N_{arm} is the normal of the left or right arm plane, and V is a unit vector connecting the elbow and wrist markers. It should be noted that the N_{arm} is parallel to the axes of the humanoid elbow joint.

5 COM ANTICIPATION MODEL

The CoM of a humanoid robot is a vital indicator to its stability. In order to remain statically stable, the projection of CoM on the floor should remain within the support polygon defined by the two feet of the humanoid. If the human performer were to lift his/her foot, the CoM of the humanoid robot would have to be shifted in advance towards the other foot in order to maintain balance. In order to know when this shift is required, we take inspiration from results in human neuroscience research. Studies have reported strategies by which motion of the CoM in humans can be related to foot placement and hip orientation (Patla et al., 1999), (Vallis and McFadyen, 2005).

To manipulate the projection of the humanoid CoM on the floor we constrain it to track a target. The target position is computed depending on the current stance of the HN model, i.e. Double Support (DS) or Single Support (SS). The transition of stance from single to double support is detected using the position and velocity of the feet. When either of these measures exceed a pre-determined threshold a change of stance is said to have occured. For the motion of the CoM the target is computed as:

$$CoM_i = \begin{cases} CoM_{i-1} + \alpha(V_{head} \cdot V_{feet})V_{feet} & \text{if DS} \\ p_{foot} + \beta V_{head} & \text{if SS} \end{cases} \tag{7}$$

where,

CoM_i = CoM X and Y positions at time step i,
V_{head} = HN Model head 2D velocity vector,
V_{feet} = Unity vector across robot's feet,
p_{foot} = Humanoid support foot X and Y positions,
α, β are constants.

6 IMPLEMENTATION OF ONLINE HUMAN-HUMANOID MOTION TRANSFER

Our framework was implemented using the software achitecture Genom, (Fleury et al., 1997). Mainly, we have four modules to establish communication from the motion capture system to the HRP2 robot interface. First, we have a motion capture server whose function is to send motion data to the network via UDP protocol. These data are filtered as described

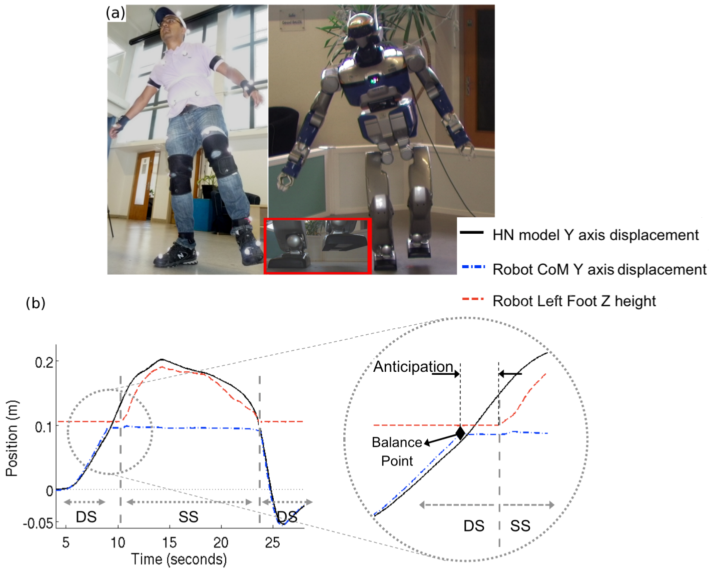


Figure 5: Scenario 2. (a) Picture of humanoid imitating the human lifting his foot. (b) Sideways displacement of the HN model head (solid black line), and the vertical height of the humanoid's lifting foot (dashed red line). Also shown is the Y displacement of the humanoid's CoM (dash-dot blue line). Zoom inset shows a magnified view of the anticipation phase. The anticipation occurs between the 'Balance point' and the time when the humanoid lifts its foot.

in section 2. A second Genom module reads the seamless motion data and computes the robot motion. This module implements the HN-normalized model, the CoM anticipation model and the prioritized inverse kinematics solver. Finally, via a plugin we send robot motion data from the motion generation Genom-module to the HRP2 interface control panel.

We present two scenarios that illustrate the capabilities of our algorithm. In the first scenario we assume the robot's feet to be fixed and imitate the motion of a human performer executing a slow dance with the upper body, including bending of the knees and ankles. In the second scenario, the robot is allowed to lift-off with one of its feet and balance on the other foot. This was chosen to illustrate the anticipation model which prepares the humanoid for balancing on one foot. The parameters used for the CoM anticipation model were, $\alpha = 0.12$, $\beta = 0.01$. All computations were run on an Intel Core 2 CPU 6400 @2.13GHz, with 2GB of RAM memory. At each solution step we required $\sim$30 ms to build the Humanoid-Normalized model and to solve the stack of tasks using a damped inverse kinematics solver. The video of the results can be accessed at http://homepages.laas.fr/manu/videos/motionImitation.mp4

6.1 Dancing

The human performer was asked to perform a simple dance without stepping or sliding his feet. Figure 4-a shows the posture of the human and the humanoid in the middle of the dance. The motion computed by the algorithm was smooth, without joint position or velocity limit violations, and was quasi-statically stable. Figure 4-b & c show the roll, pitch and yaw angles of the head and chest of the HN model and those of the humanoid robot. Despite the low priority given to the head orientation task, we see that the yaw and pitch angles were matched very closely, while roll angle of the humanoid was much lesser than the HN model. This was because the yaw and pitch axis are directly available on HRP2 (independent of the other joints), however, the humanoid does not have a roll axis for the head joint. The roll variation seen in Figure 4-b was due to the movement of the whole body. Chest roll of the HN model was not considered, but to account for the movement of the rest of the body we see an induced roll component on HRP2. The pitch and yaw angles of the humanoid's chest followed the HN model less closely due to the lower priority of this task. Since the arms and the head are connected to the chest, and their respective tasks have a higher priority, the chest joint has a reduced degree of mobility.

6.2 Foot Lift

In this scenario the human performer shifted his weight onto one leg and maintained his balance for a few seconds before slowly returning to rest on both feet. Figure 5-a shows the human and the humanoid balancing on one foot (SS stance). The motion of the

Table 1: Mean RMS error between HN model and humanoid. Values in brackets denote the mean RMS error in X,Y and Z positions for wrist positions, and roll (R), pitch (P) and yaw (Y) for head, chest and waist orientations.

Property	Mean RMS position (m)	Mean RMS orientation (deg)
CoM	~ 0	-
Head	~ 0	4.09 (R: 11.8 P: 0.22 Y: 0.26)
Left wrist	0.02 (X: 0.013 Y: 0.02 Z: 0.3)	-
Right wrist	0.05 (X:0.05 Y:0.017 Z: 0.08)	-
Chest	-	4.2 (R: 1.1 P: 6.7 Y: 4.8)
Waist	-	6.4 (R: 1.1 P: 4.62 Y: 0.52)

head in the HN model, and the vertical position of the foot of the humanoid is plotted in Figure 5-b. We observe that the head shifts towards the support foot (right foot) before the lifting the other foot (Figure 5-b). The sideways displacement of the head reaches the Y position of the support foot about 1s before foot lift. Before reaching this point, the CoM projection was derived according to Eq. 7 (DS stance). Once the head reaches the support foot, the CoM is maintained at this position (referred to as "Balance Point" in Figure 5-b). After this point, the behavior of the CoM is dictated by a different relation (SS stance in Eq. 7). It should be noted that for slow head motion, the projection of the CoM and the head position coincide (a small offset can be seen in the zoom inset in Figure 5-b)).

7 DISCUSSION

Among studies that have looked at human-humanoid motion retargeting it is generally difficult to define a single metric reflecting the quality of motion transfer. This can be because of several reasons. The physical structure of humanoids can vary quite significantly (size, number of DoFs, range of motion etc). Moreover, defining a mathematical term to the rather abstract idea of "good" motion imitation is not clear. Dariush and colleagues, formulated a measure for "tracking error" to express the effectiveness of their algorithm. We employ a similar strategy and additionally investigate the limitations of our approach vis-à-vis dynamic stability of the humanoid.

7.1 Quality of Motion Imitation

Quality of motion imitation was quantified by measuring the root mean square error between the target (HN model) and the humanoid robot. Table 1 lists the relevant parameters and the errors. The position of the CoM and head were tracked almost prefectly. This was because both these tasks had a very high priority. Comparitively, head orientation which had a lower priority had a mean error of 4 deg. But it should be noted that most of this error was because of the roll angle (HRP2 does not have a head roll axis). The right wrist position error was slightly larger that the left wrist. This can be attributed to the fact that left wrist task came before the right wrist task in the priority list. Thus, once the left wrist position and orientation was fixed, it became more difficult for the right wrist to reach exactly its target transformation. Comparing across studies, Dariush et al. 2008b, reported an error of about 0.02 m in tracking the wrist position while assigning them to a "medium priority group". In our case the head was the highest priority, and hence a low error, while the hands were low priority, hence the larger error. Chest and waist orientation were lower in the priority list and hence show larger errors in orientation than the other joints. Overall, these results show that we were able to retarget a large part of the motion of the human onto the humanoid.

7.2 Limitations

We analysed the limitations of our imitation system, and those of our humanoid, by setting up the following test. We attached the marker set on the performer and he was asked to move his right hand in an up-and-down motion at different speeds. After transferring the motion to the humanoid, we observed the shift of the Zero Moment Point (ZMP) of the humanoid for the different human hand speeds. For the up and down motion, we detect the maximum and minimum values of the ZMP components, and check if it is inside the supporting polygon. We found that the humanoid became unstable when the hand speed was higher than 1 m/s (Figure 6). This example illustrates the limitations of using inverse kinematics without considering, simultaneously, the dynamic stability of the humanoid. To truly imitate both the kinematics and dynamics of the human motion, it would be important to take both of these into account during the modeling stage itself. For example, using a dynamical model (exact or simplified) at the motion planning stage could be a useful in this regard.

Kanehiro et al. (Kanehiro et al., 2008b), devised a

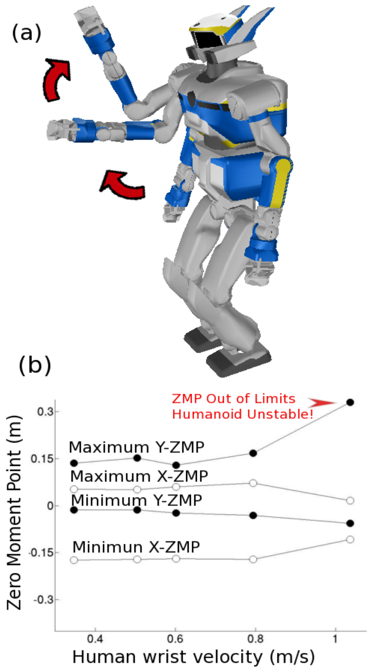


Figure 6: (a) Illustration of the hand up and down motion on the simulated HRP2, (b)Plot of ZMP of the humanoid robot vs. human hand velocities. Illustrated is the point at which the humanoid becomes unstable because of excessive hand speeds.

way to optimize quasi-static humanoid motion, such that it can be performed faster, but also within the dynamic limits of the humanoid. We recorded the online dancing motion of the humanoid, and optimized it offline to see how much faster the same motion could be played. We found that our original 60 seconds of dancing motion could be optimized to 31 seconds while respecting the kinematic and dynamic limits of the humanoid HRP2. Thus, offline optimization could serve as a benchmark in judging how effective and robust online motion retargeting algorithms really are.

Finally, one limitation of our formulation of the HN model is that it cannot guarantee self-collison avoidance on the humanoid. This could be included at the inverse kinematics stage by defining self-collison avoidance as a constraint (Dariush et al., 2008a) (Kanoun, 2009) (Kanehiro et al., 2008a).

8 SUMMARY AND FUTURE PERSPECTIVES

In this study we have presented an online method by which a humanoid robot can imitate human motion. The evaluation of the results show that the motion generated by the humanoid closely resembles the original human motion. The proposed CoM anticipation model allows the humanoid to balance itself on one foot taking the cue from the human. This model inspired from neuroscience research opens up new windows towards incorporating biological principles in humanoid motion control (Berthoz, 2000), (Sreenivasa et al., 2009). The use of the Humanoid-Normalized model allows standardization across anthropomorphic figures irrespective of proportions (for example marionettes). To further improve imitation it could be interesting to consider the exact dynamics of the humanoid, as well as self-collison avoidance, in the motion planning algorithm. At the present state of this work, we can say that our humanoid robot HRP-2, is capable of mimicking Tai-chi like movements, but not yet quite at the level of Karate.

ACKNOWLEDGEMENTS

F.J. Montecillo-Puente benefits of a Mexican CONACyT, SEP grant. The authors would like to thank Anthony Mallet, Tan Viet Anh Truong and Oussama Kanoun for taking part in helpful discussions and assisting with the experiments on HRP2 and motion capture. Part of this work is supported by the French ANR Project Locanthrope.

REFERENCES

Berthoz, A. (2000). *The brain's sense of movement*. Harvard University Press, Cambridge, MA.

Boulic, R., Maupu, D., and Thalmann, D. (2009). On scaling strategies for the full body interaction with virtual mannequins. *Journal Interacting with Computers, Special Issue on Enactive Interfaces*, 21(1-2):11–25.

Chois, K. J. and Ko, H. S. (December 2000). Online motion retargetting. *The Journal of Visualization and Computer Animation*, 11(5):223–235.

Dariush, B., Gienger, M., Arumbakkam, A., Goerick, C., Zhu, Y., and Fujimura, K. (2008a). Online and markerless motion retargeting with kinematic constraints. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 191–198.

Dariush, B., Gienger, M., Jian, B., Goerick, C., and Fujimura, K. (2008b). Whole body humanoid control from human motion descriptors. In *IEEE International Conference on Robotics and Automation*, pages 2677–2684.

Fleury, S., Herrb, M., and Chatila, R. (1997). Genom: A tool for the specification and the implementation of operating modules in a distributed robot architecture. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 842–848.

Kanehiro, F., Lamiraux, F., Kanoun, O., Yoshida, E., and Laumond, J.-P. (2008a). A local collision avoidance method for non-strictly convex polyhedra. *Robotics: Science and Systems*, IV.

Kanehiro, F., Suleiman, W., Lamiraux, F., Yoshida, E., and Laumond, J.-P. (2008b). Integrating dynamics into motion planning for humanoid robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 660–667.

Kanoun, O. (2009). *Task-driven motion control for humanoid robots*. PhD thesis, LAAS-CNRS; Université de Toulouse.

Multon, F., Kulpa, R., and Bideau, B. (2008). Mkm: A global framework for animating humans in virtual reality applications. *Presence: Teleoper. Virtual Environ.*, 17(1):17–28.

Nakamura, Y. (1991). *Advanced Robotics: Redundancy and Optimization*. Addison-Wesley Longman Publishing, Boston.

Nakaoka, S., Nakazawa, A., Kanehiro, F., Kaneko, K., Morisawa, M., and Ikeuchi, K. (2005a). Task model of lower body motion for a biped humanoid robot to imitate human dances. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3157–3162.

Nakaoka, S., Nakazawa, A., Kanehiro, F., Kaneko, K., Morisawa, M., and Ikeuchi, K. (2005b). Task model of lower body motion for a biped humanoid robot to imitate human dances. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3157–3162.

Patla, A., Adkin, A., and Ballard, T. (1999). Online steering: coordination and control of body center of mass, head and body reorientation. *Experimental Brain Research*, 129(4):629–634.

Ruchanurucks, M., Nakaoka, S., Kudoh, S., and Ikeuchi, K. (2006). Humanoid robot motion generation with sequential physical constraints. In *IEEE International Conference on Robotics and Automation,*, pages 2649–2654.

Schaal, S., Ijspeert, A., and Billard, A. (2003). Computational Approaches to Motor Learning by Imitation. *philosophical transactions: biological sciences*, 358(1431):537–547. philosophical transactions: biological sciences (The Royal Society).

Shon, A., Grochow, K., and Rao, R. (2005). Robotic imitation from human motion capture using gaussian processes. In *5th IEEE-RAS International Conference on Humanoid Robots*.

Siciliano, B. and Slotine, J. (1991). A general framework for managing multiple tasks in highly redundant robotic systems. In *IEEE Internatioal Conference on Advanced Robotics*, pages 1211–1216.

Sreenivasa, M.-N., Soueres, P., Laumond, J.-P., and Berthoz, A. (2009). Steering a humanoid robot by its head. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Suleiman, W., Yoshida, E., Kanehiro, E., Laumond, J.-P., and Monin, A. (2008). On human motion imitation by humanoid robot. In *IEEE International Conference on Robotics and Automation*.

Takano, W., Yamane, K., and Nakamura, Y. (2007). Capture database through symbolization, recognition and generation of motion patterns. In *IEEE International Conference on Robotics and Automation*, pages 3092–3097.

Ude, A., Atkeson, C., and M., R. (2004). Programming full-body movements for humanoid robots by observation. In *Robotics and Autonomous Systems*, volume 47, pages 93–108.

Vallis, L. and McFadyen, B. (2005). Children use different anticipatory control strategies than adults to circumvent an obstacle in the travel path. *Experimental Brain Research*, 167(1):119–127.

Vukobratovic, M. and Stepanenko, J. (1972). On the stability of anthropomorphic systems. *Mathematical Biosciences*, 15:1–37.

Yamane, K. and Hodgins, J. (2009). Simultaneous tracking and balancing of humanoid robots for imitating human motion capture data. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2510–2517.

Yoshida, E., Kanoun, O., Esteves, C., and Laumond, J.-P. (2006). Task-driven support polygon humanoids. In *IEEE-RAS International Conference on Humanoid Robots*.

FREQUENCY EXTRACTION BASED ON ADAPTIVE FOURIER SERIES
Application to Robotic Yoyo

Tadej Petrič, Andrej Gams and Leon Žlajpah
Department of Automation, Biocybernetics and Robotics, Jožef Stefan Institute, Jamova cesta 39, 1000 Ljubljana, Slovenia
{tadej.petric, andrej.gams, leon.zlajpah}@ijs.si

Keywords: Fourier series, Nonlinear dynamical systems, Adaptive frequency oscillators, Imitation, Yo-yo.

Abstract: We present a novel method to obtain the basic frequency of an unknown periodic signal with an arbitrary waveform. The method originates from nonlinear dynamical systems for frequency extraction, which are based on adaptive frequency oscillators in a feedback loop. While using several adaptive frequency oscillators in a loop results in extraction of separate frequency components, our method extracts the basic frequency of the input signal without any additional logical operations. The proposed method uses a whole Fourier series representation in the feedback loop. In this way it can extract the frequency and the phase of an unknown periodic signal, in real-time, and without any additional signal processing or preprocessing. The method also determines the Fourier series coefficients and can be used for dynamic Fourier series implementation. It can be used for the control of rhythmic robotic tasks, where only the extraction of the fundamental frequency is crucial. This is demonstrated on a highly nonlinear and dynamic task of playing the robotic yo-yo.

1 INTRODUCTION

Controlling rhythmic robotic tasks that require synchronization with the actuated device or interaction with the external environment is a difficult task, and requires complex sensory systems and advanced knowledge (Petrič et al., 2009). For example, such rhythmic tasks include handshaking (Kasuga and Hashimoto, 2005), locomotion (Ijspeert, 2008), drumming (Degallier et al., 2008), or playing with different toys, like the yo-yo(Žlajpah, 2006) or the gyroscopic device called Powerball (Gams et al., 2007). Controlling these tasks with robots requires both accurate trajectory generation and frequency tuning.

Determining the fundamental frequency of a task is a complex problem and can be achieved in different ways, e.g. with signal processing methods, such as FFT, or with the use of nonlinear oscillators (Matsuoka et al., 2005). Furthermore, trajectory generation and modulation are still difficult tasks in robotics. One possible approach to trajectory generation and the modulation is the imitation (Schaal, 1999), which can be preformed in several different ways, using encoding methods like splines (Ude et al., 2000) or dynamic movement primitives (DMP) (Schaal et al., 2007).

Not many approaches that combine both frequency extraction and waveform learning exist. One of them is the use of a two-layered imitation system based on nonlinear dynamical systems (Gams et al., 2009). In their work, the authors explained that the imitation system can be used for extracting the frequency of the input signal, learning its waveform, and imitating the waveform at the extracted or any other frequency. Similar, but with less properties for trajectory generation and modulation can be achieved by using only the first layer of this system for both frequency extraction and waveform learning (Righetti et al., 2006). The described systems are based on adaptive frequency oscillators in a feedback loop. Such an approach can determine several frequency components of an input signal. Despite favorable properties of these systems, there is a considerable drawback in determining the basic or fundamental frequency of the input signal.

For complex periodic signals with several frequency components, the first layer of the imitation system, referred to as the canonical dynamical system, has to include a high number of oscillators in the feedback loop. Using this system for movement imitation requires determining the basic frequency. This is accomplished by a logical algorithm that follows the feedback loop. With a high number of oscillators, and when several of the oscillators tune to the same frequency, this can become extremely complex. A high number of oscillators is practically necessary and cannot be avoided.

The contribution of this paper is the novel design of the canonical dynamical system for the two-layered imitation system. The proposed approach does not require a logic algorithm to determine the fundamental frequency of the input signal, as in the original approach (Gams et al., 2009). We use a single adaptive phase oscillator in a feedback loop. The oscillator is followed by a complete Fourier series approximation, with a built-in algorithm to determine the Fourier coefficients. The combination of an adaptive phase oscillator and the adaptive Fourier series allows us to extract the fundamental frequency of the input signal and use it to control rhythmic robotic tasks.

With this approach we essentially implemented a real-time, adaptive Fourier series analysis. Our system is able to calculate the Fourier coefficients of an unknown periodic signal in real-time and is computationally inexpensive. The usefulness of this system is presented on the case of playing the yo-yo. Controlling the yo-yo has already been a subject of several studies (Žlajpah, 2006; Jin et al., 2009), which mostly rely on complex, specially designed controllers based on the models of the device. The task of playing yo-yo is highly non-linear and requires on-line frequency adaptation. The proposed approach simplifies the synchronization between the upward jerk of the robot and the movement of the yo-yo by determining the frequency of the up-down motion from a measurable periodic quantity.

The paper is organised as follows. In section 2, we give a brief description of the original two-layered imitation system with an emphasis on the first layer - the canonical dynamical system. In section 3 we describe the novel approach using the Fourier series in the feedback loop. In section 4 we evaluate the proposed approach in simulation, and on a real-world experiment of playing the yo-yo. Conclusions and summary are in section 5.

2 TWO-LAYERED IMITATION SYSTEM

The two-layered imitation system was presented in detail in (Gams et al., 2009). In their work the authors explained that the system can be used for extracting the frequency spectrum of the input signal, learning the waveform of one period, and imitating the desired waveform at an arbitrary frequency. The system structure is presented in Figure 1. The first layer, i.e. the canonical dynamical system, is used for frequency extraction. It is based on a set of adaptive frequency oscillators in a feedback loop. The second layer is called the output dynamical system and is used for learning and repeating the desired waveform. The latter is based on dynamic movement primitives - DMPs, e.g. (Schaal et al., 2007).

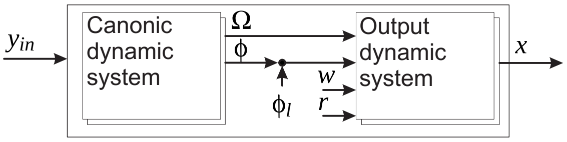


Figure 1: Two-layered structure of the imitation system. The input y_{in} is a measured quantity and the output is the desired trajectory x of the robot. The input ϕ_l is the additional phase lag and r is the amplitude of the trajectory. The system can work in parallel for an arbitrary number of dimensions.

The first layer of the system has two major tasks. It has to extract the fundamental frequency Ω of the input signal and it has to exhibit stable limit cycle behavior in order to provide the phase signal Φ. The basis of the canonical dynamical system is a set of adaptive phase oscillator with applied learning rule as introduced in (Buchli and Ijspeert, 2004). In order to accurately determine the frequency, it is combined with a feedback structure (Buchli et al., 2008) (see Figure 2). The feedback structure of M adaptive frequency oscillators is governed by

$$\dot{\phi}_i = \omega_i - K \cdot e \cdot \sin\phi_i, \qquad (1)$$

$$\dot{\omega}_i = -K \cdot e \cdot \sin\phi_i, \qquad (2)$$

$$e = y_{in} - \hat{y}, \qquad (3)$$

$$\hat{y} = \sum_{i=1}^{M} \alpha_i \cos\phi_i, \qquad (4)$$

$$\dot{\alpha}_i = \eta \cdot e \cdot \cos\phi_i, \qquad (5)$$

where K is the coupling strength, ϕ_i, $i = 1...M$ is the phase of separate oscillators, y_{in} is the input signal, M is the number of oscillators, α_i is the amplitude associated with the i-th oscillator, and η is the learning constant.

As shown in Figure 2, each of the oscillators in the feedback structure receives the same input, i.e. the difference between the input signal and the weighted sum of separate frequency components. Such a feedback structure preforms a kind of Fourier analysis. The number of extracted frequencies depends on how many oscillators are used. As only the fundamental frequency is of interest, the feedback structure is followed by a logic algorithm. Determining the correct frequency and the phase is crucial, because they are

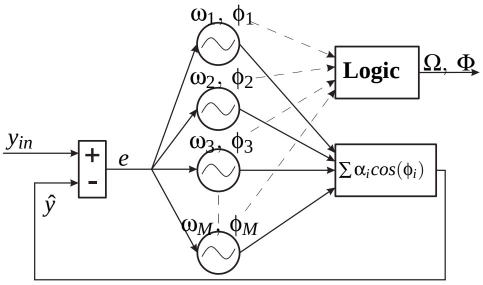


Figure 2: Feedback structure of M nonlinear adaptive frequency oscillators. Note the logic algorithm that follows the feedback loop.

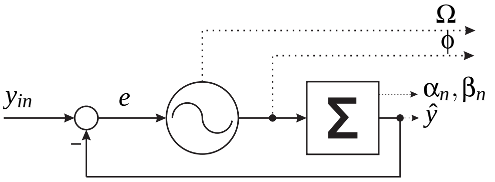


Figure 3: Feedback structure of nonlinear adaptive frequency oscillator combined with dynamic Fourier series. Note that no logic algorithm needed.

the basis for the output dynamic system and the desired behavior of the actuated device.

One possible approach is to choose the first nonzero frequency as was presented in (Gams et al., 2009). However, it has a drawback that when more than one oscillators converge to, or oscillate, around the same frequency, the logic algorithm switches between the oscillators, and consequently the phase will not be smooth, leading to oscillations in the output trajectory.

3 CANONICAL DYNAMICAL SYSTEM BASED ON FOURIER SERIES

In this section a novel architecture for canonical dynamical system is presented. As the basis of the canonical dynamical system we use a single nonlinear phase oscillator with applied learning rule (Buchli and Ijspeert, 2004). This is combined with a feedback structure based on an adaptive Fourier series in order to accurately determine the frequency. A feedback structure with an adaptive frequency oscillator combined with an adaptive Fourier series is shown in Figure 3. The feedback structure of an adaptive frequency phase oscillator is governed by

$$\dot{\phi} = \Omega - K \cdot e \cdot \sin\phi, \qquad (6)$$

$$\dot{\Omega} = -K \cdot e \cdot \sin\phi, \qquad (7)$$

$$e = y_{in} - \hat{y}, \qquad (8)$$

where K is the coupling strength, ϕ is the phase of the oscillator, e is the input into the oscillator and y_{in} is the input signal. If we compare Eqs. (1, 2) and Eqs. (6, 7), we can see that the frequency Ω and the phase ϕ are now clearly defined. The feedback loop $\hat{y}$ is now represented by the Fourier series

$$\hat{y} = \alpha_0 + \sum_{i=1}^{M} (\alpha_i \cos(i\phi) + \beta_i \sin(i\phi)), \qquad (9)$$

and not by the sum of separate frequency components as in Eq. 4. M is the size of the Fourier series and α_0 is the amplitude associated with the first segment of the series, it is governed by

$$\dot{\alpha}_0 = \eta \cdot e, \qquad (10)$$

here η is a learning constant. The amplitudes associated with the other terms of the Fourier series are determined by

$$\dot{\alpha}_i = \eta \cos(i\phi) \cdot e, \qquad (11)$$

$$\dot{\beta}_i = \eta \sin(i\phi) \cdot e, \qquad (12)$$

where $i = 1...M$. As shown in Figure 3, the oscillator of the feedback structure receives the difference between the input signal and the Fourier series. Since a negative feedback loop is used, the difference approaches zero when the Fourier series representation approaches the input signal. Such a feedback structure preforms an adaptive Fourier analysis, where the phase difference between the harmonics can only be 0, $\pi/2$, π or $3\pi/2$. This is not the case in the original approach (Righetti and Ijspeert, 2006), where the phase difference can be arbitrary.

The proposed approach has the ability to adapt to the basic frequency of the input signal. The number of harmonic frequency components it can accurately extract depends on how many terms of the Fourier series are used. Since in this structure only one oscillator is used and the harmonics are encoded in the Fourier series, the basic frequency and phase are clearly defined. This is an important improvement, especially for the usefulness of the imitation system when performing rhythmic tasks.

The new architecture of the canonical dynamic system can be used as an imitation system by itself, as it is able to learn arbitrary periodic signals. After convergence, e reaches zero (with an accuracy that depends on the number of elements of the Fourier series). Once e is zero, the periodic signal stays encoded in the Fourier series. The learning process is embedded and is done in real-time. There is no need for any external optimization process or learning algorithm.

Adding the output dynamical system enables us to synchronise the motion of the robot to a measurable periodic quantity of the task we would like to preform. The measured signal is now encoded into the Fourier series and the desired robot trajectory is encoded in the output dynamic system. Sice adaptation of the frequency and the learning of the desired trajectory can be done simultaneously, all of the system time-delays can be automatically included. Furthermore, when a predefined motion pattern for the trajectory is used, the system time-delays can be adjusted with a phase lag parameter ϕ_l. This enables us to either predefine the desired motion or to teach the robot how to preform the desired rhythmic task online.

The output dynamical system also ensures greater robustness against perturbations and smooth modulation. Specially greater robustness to perturbation is crucial when performing fast, dynamic tasks.

4 EVALUATION

In the following section we evaluate of the proposed imitation system with the new canonical dynamical layer. In the Section 4.1 the numerical results from the original and the novel architectures are presented. In Section 4.2 a real-world experiment of playing the yo-yo with the use of the proposed imitation system is shown.

4.1 Simulation

In this numerical experiment the proposed architecture for the canonical dynamical system learns an arbitrary signal. The populating of the frequency spectrum is done without any signal processing, as the whole process of frequency extraction and adaptation of the waveform is completely embedded in the dynamics of the adaptive frequency oscillator combined with the adaptive Fourier series. Unless stated otherwise, we use the following parameters: $\mu = 2$, $K = 20$, $M = 10$.

Frequency adaptation results from time- and shape-varying signals are illustrated in Figure 4. The input signal itself is of three parts: a periodic pulse signal, a sinusoid, and a sawtooth wave signal. Transition between the signal parts is instant for both frequency and waveform. We can see that after the change of the input signal, the output frequency stabilises very quickly.

A single adaptive frequency oscillator in a feedback loop is enough, because the harmonics of the input signal are encoded with the Fourier series in the feedback loop. As can be seen from the bottom plots in Figure 4, the input signal and the feedback signal are very well matched. The approximation error depends only on M. A comparison with the original approach as proposed in (Buchli et al., 2008) is given in Figure 5. In their approach, if there are not enough oscillators to encode the input signal, the system will only learn the frequency components with more power. Thus, the output signal will only be an approximation.

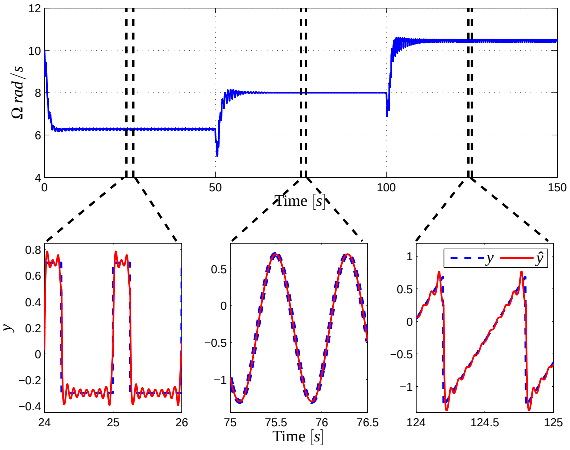


Figure 4: Typical convergence of an adaptive frequency oscillator combined with an adaptive Fourier series, driven by a periodic signal with different waveforms and frequencies. Frequency adaptation is presented in the top plot and the comparison between the input signal y and the approximation $\hat{y}$ in the bottom plot.

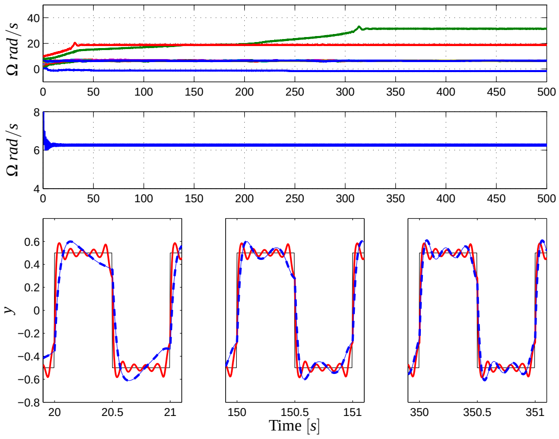


Figure 5: Comparison between the pool of the adaptive oscillators and our proposed approach. First plot shows evolution of frequency distribution using a pool of 10 oscillators. Middle plot shows the extracted frequency using an adaptive frequency oscillator combined with Fourier seres. The comparison of approximated signals is presented in the bottom plot. The thin solid line presents the input signal, the solid line presents our new proposed approach and the dotted line presents the pool of adaptive oscillators.

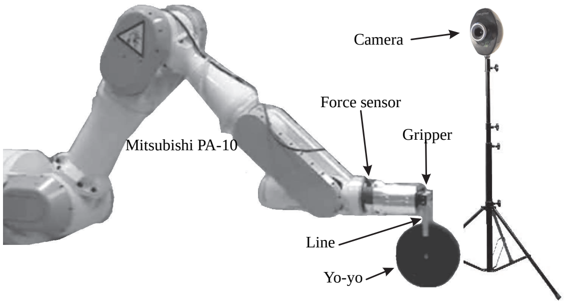


Figure 6: Experimental setup.

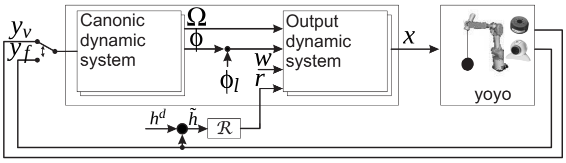


Figure 7: Proposed two-layered structure of the control system for controlling the peak height of the yo-yo. The input is either the force y_f or the visual feedback y_v.

However, if there are more oscillators than the frequency components to learn, either some of them will not converge to any frequency or the same frequency components will be coded by several oscillators, as shown in the top plot in Figure 5, where a pool of ten oscillators was used. In this particular experiment, five of the oscillators converge to the basic frequency of the signal.

Choosing the right oscillator from that pool is a very difficult task and requires a complex logic algorithm. On the other hand, using our new approach, where the feedback is encoded with a Fourier series, the oscillator converges to the basic frequency of the input signal. Therefore, the basic frequency and the phase are clearly defined. Furthermore, the approximation and the convergence of the feedback signal is quicker, as it is shown in the bottom plots in Figure 5. Even after 350 s, the original architecture from (Righetti and Ijspeert, 2006) did not produce as good an approximation as it was after 20 s when using our new proposed canonical dynamical system.

4.2 Application to Robotic Yo-yo

To illustrate the proposed approach we implemented it on a real robot playing yo-yo.

Playing yo-yo with a robot can be achieved in different ways, depending on what one can measure. It can be the length of the unwound string, which can be effectively measured by a vison system. As described in (Žlajpah, 2006), using vison is also one of the ways humans do it, even though approaches using only the measured force were described (Jin et al., 2009). With our proposed system, playing yo-yo can be accomplished either with force feedback or with visual feedback. Furthermore, the proposed system is able to synchronise even if the input signal is changed from one measurable quantity to another during the experiment.

We preformed the experiment on a Mitsubishi PA-10 robot as presented in Figure 6. A force sensor (JR3), was attached to the end effector to measure the impact force of the yoyo, and a USB camera was used to measure the length of the unwound string.

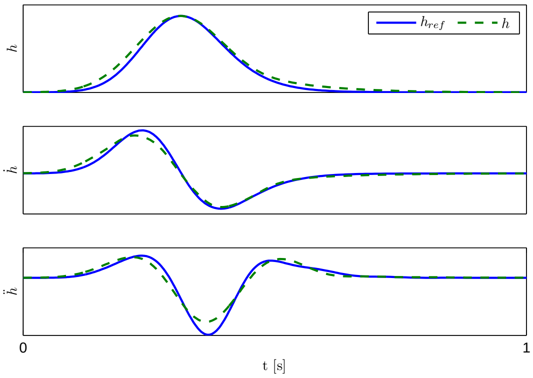


Figure 8: Pre-defined hand motion pattern for playing yo-yo.

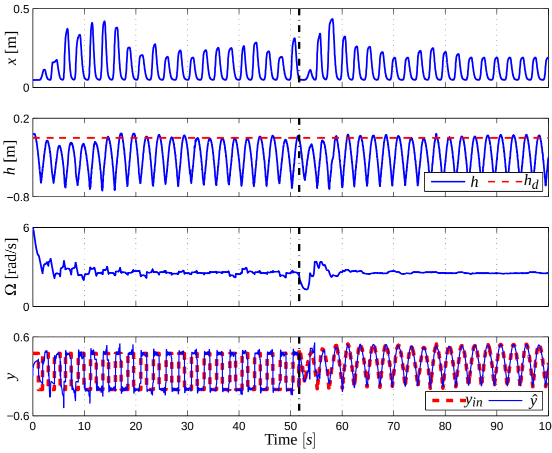


Figure 9: Robot trajectory x in the top plot, height h of the yo-yo in the second plot, extracted frequency in the third plot and signal adaptation $\hat{y}$ in the bottom plot. At 52 s the input signal is switched form force feedback to visual feedback. Yo-yo parameters in this case are: axle radius $r_a = 0.01$ m and mass $m = 0.2564$ kg.

The two layered imitation system with the novel canonical dynamical system was implemented in Matlab/Simulik. The control scheme is presented in Figure 7. As we can see, the imitation system, based on a nonlinear oscillator combined with dy-

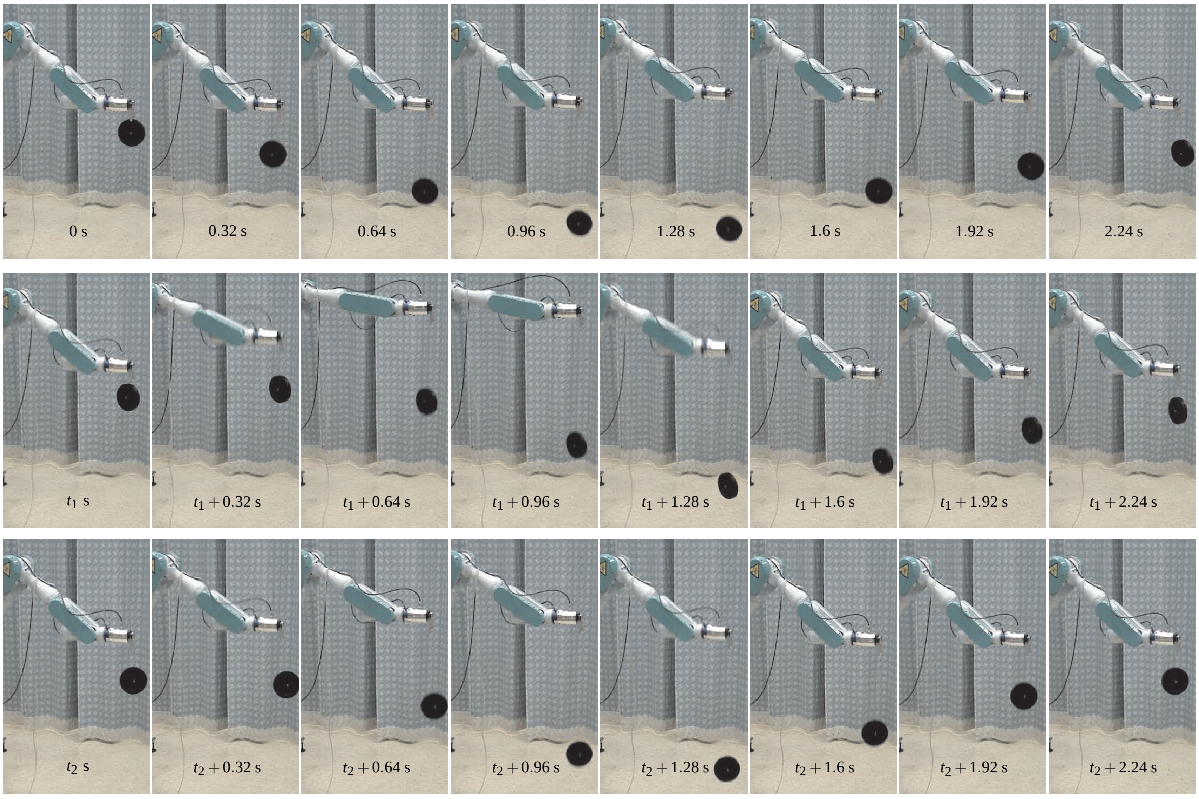


Figure 10: Image sequence of a robotic yoyo. Top sequence present the decent of the yoyo from the gripper. In the middle sequence a behavior of the system after switching from force feedback to vision feedback is shown ($t_1 = 54$ s) and in bottom sequence the behavior in steady state is presented ($t_2 = 81$ s).

namic Fourier series, provides the desired trajectory for the robot with a yo-yo attached at the top. The motion of the robot was constrained to up-down motion using inverse kinematics. The length of the string or the force from the top of the robot can be used as the input into the system. Since a measurable force difference appears only as a spike, when the yo-yo hits the end of the string, we modify the signal in a way that it carries more energy. In our particular case, we use the measured spike to create a short pulse.

To preform the task, we first determine the waveform of the required motion pattern. We chose the motion pattern described in (Žlajpah, 2006), which satisfies the required criteria for playing the yo-yo. The hand motion pattern encoded into the output dynamic system (dashed line), and the desired hand motion pattern (solid line) are presented in Figure8.

The frequency of the task depends on the parameters of the yo-yo itself, and on how high the yo-yo rolls up along the string. The height can be influenced by the amplitude of the hand motion, which can be easily modified using the amplitude parameter r of the motion, see Figure 7. PI controller was used to control the peak height of the yo-yo. The controller is given by

$$u(t) = k_p e(t) + k_i \int e(t) dt, \tag{13}$$

where $k_p = 2$, and $k_i = 0.4$ were determined empirically. Figure 9 shows the results of frequency adaptation and yo-yo height during the experiment.

As we can see, the frequency of the imitated motion quickly adapted to the motion of the yo-yo and stable motion was achieved. At approximately 52 s the input into the imitation system was switched from force feedback to visual feedback. At that point some oscillation in the frequency and the approximation of the input signal can be observed because they have to adapt to the new waveform of the input signal. Furthermore, form the middle sequence in Figure 10 we can see that the amplitude of hand motion is higher after switching from the force feedback to the vision feedback. Despite the change, the imitation system still manages to extract the correct frequency and the robot motion returns to steady-state oscillations. Note that in the bottom sequence in Figure 10 the hand amplitude is smaller than immediately after the switch.

As far as we know, this is the first system which has the capability of playing the yo-yo by force feedback or by vision feedback, without changing the system parameters. Furthermore, switching from one to another measured quantity can even be done during the experiment. This shows that the proposed system is adaptable and robust.

5 CONCLUSIONS

We presented a new architecture for the canonical dynamical system which is a part of a two layered imitation system, but can be used as an imitation system by itself. The dynamical system which, is used to extract the frequency, is composed of a nonlinear phase oscillator combined with a Fourier series. This system essentially implements an adaptive Fourier series of the input signal. It can extract the frequency, phase and the Fourier series coefficients of an unknown periodic signal. This is done in real-time without any additional processing of the input signal. Integrating this system into the imitation system based on dynamic motion primitives enables simple and computationally inexpensive control of rhythmic tasks with at least one measurable periodic quantity.

Furthermore, we presented the use of the imitation system to preform a rhythmic task that requires synchronization with the controlled device. For playing the yo-yo, we have shown that the information on how high the yo-yo rolls up along the string, or the force feedback is enough to achieve stable performance. The proposed approach enables to play yo-yo by measuring either the force or the yo-yo position. Furthermore we also showed that the system has the capability of changing the measured quantity in a single experiment without loosing the synchronization between the robot and the yo-yo.

REFERENCES

Buchli, J. and Ijspeert, A. (2004). A simple, adaptive locomotion toy-system. In Schaal, S., Ijspeert, A., Billard, A., Vijayakumar, S., Hallam, J., and Meyer, J., editors, *From Animals to Animats 8. Proceedings of the Eighth International Conference on the Simulation of Adaptive Behavior (SAB'04)*, pages 153–162. MIT Press.

Buchli, J., Righetti, L., and Ijspeert, A. J. (2008). Frequency Analysis with coupled nonlinear Oscillators. *Physica D: Nonlinear Phenomena*, 237:1705–1718.

Degallier, S., Righetti, L., Natale, L., Nori, F., Metta, G., and Ijspeert, A. (2008). A modular bio-inspired architecture for movement generation for the infant-like robot iCub. In *Proceedings of the 2nd IEEE RAS / EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*.

Gams, A., Ijspeert, A. J., Schaal, S., and Lenarcic, J. (2009). On-line learning and modulation of periodic movements with nonlinear dynamical systems. *Auton. Robots*, 27(1):3–23.

Gams, A., Žlajpah, L., and Lenarčič, J. (2007). Imitating human acceleration of a gyroscopic device. *Robotica*, 25(4):501–509.

Ijspeert, A. J. (2008). Central pattern generators for locomotion control in animals and robots: A review. *Neural Networks*, 21(4):642–653.

Jin, H.-L., Ye, Q., and Zacksenhouse, M. (2009). Return maps, parameterization, and cycle-wise planning of yo-yo playing. *Trans. Rob.*, 25(2):438–445.

Kasuga, T. and Hashimoto, M. (2005). Human-robot handshaking using neural oscillators. In *ICRA*, pages 3802–3807.

Matsuoka, K., Ohyama, N., Watanabe, A., and Ooshima, M. (2005). Control of a giant swing robot using a neural oscillator. In *ICNC (2)*, pages 274–282.

Petrič, T., Gams, A., and Žlajpah, L. (2009). Modeling and control strategy for robotic powerball. In *RAAD*, page 7.

Righetti, L., Buchli, J., and Ijspeert, A. J. (2006). Dynamic hebbian learning in adaptive frequency oscillators. *Physica D*, 216(2):269–281.

Righetti, L. and Ijspeert, A. J. (2006). Programmable Central Pattern Generators: an application to biped locomotion control. In *Proceedings of the 2006 IEEE International Conference on Robotics and Automation*.

Schaal, S. (1999). Is imitation learning the route to humanoid robots? *Trends in cognitive sciences*, 6:233–242.

Schaal, S., Mohajerian, P., and Ijspeert, A. (2007). Dynamics systems vs. optimal control–a unifying view. *Progress in Brain Research*, 165:425–445.

Ude, A., Atkeson, C. G., and Riley, M. (2000). Planning of joint trajectories for humanoid robots using b-spline wavelets. In *ICRA*, pages 2223–2228.

Žlajpah, L. (2006). Robotic yo-yo: modelling and control strategies. *Robotica*, 24(2):211–220.

BILATERAL TELEOPERATION FOR FORCE SENSORLESS 1-DOF ROBOTS

Stefan Lichiardopol, Nathan van de Wouw and Henk Nijmeijer
Dept. of Mechanical Engineering, Eindhoven University of Technology, PO Box 513, 5600 MB Eindhoven, The Netherlands
{*s.lichiardopol, n.v.d.wouw, h.nijmeijer*}*@tue.nl*

Keywords: Bilateral teleoperation, Force sensor-less robotic setups.

Abstract: It is well known that for bilateral teleoperation, force feedback information is needed. In this paper, we propose a control approach for bilateral teleoperation with uncertainties in the model of the slave robot and which does not use force sensors for haptic feedback. The controller design is based on a cyclic switching algorithm. In the first phase of the cyclic algorithm, we estimate the environmental force and in the second phase a tracking controller ensures that the position of the slave robot is tracking the position of the master robot. A stability analysis of the overall closed-loop system is presented and the approach is illustrated by means of an example.

1 INTRODUCTION

In this paper, we consider the problem of bilateral teleoperation in force-sensor-less robotic setups. It is well-known that haptic robotic devices and teleoperation systems exploit information regarding the external forces (see (Lawrence, 1993) and (Hokayem and Spong, 2006), e.g. for haptic feedback). The slave robot interacts with the environment and its dynamics are dependent on external forces induced by this interaction. These forces can be contact forces (interaction forces between environmental objects and the robot) or exogenous forces induced by the environment.

In bilateral teleoperation, knowledge on the unknown environmental force applied on the slave robot is typically needed to achieve coordinated teleoperation. One option for obtaining such disturbance information is to equip the slave robot with force-sensors; for examples of such robotic devices, especially haptic devices, which use force sensors the reader is referred to (Lawrence, 1993), (Yokokohji and Yoshikawa, 1994). However, in many cases, the most important external forces for multi-link robots appear at the end-effector. Note that force sensing at the end effector of the robot is often not feasible since the external forces will typically interact with the load, which the slave robot is e.g. positioning, directly (and not with the robot end-effector). Besides, in some cases, the position at which the external forces are applied is a priori unknown and may be on a robot link as opposed to on the end-effector. Moreover, the usage of force-sensors can be expensive and increase the production costs of the robot which can be undesirable especially in domestic applications.

For these reasons, a disturbance estimation scheme for force-sensor-less robots can be intersting. Disturbance observers (DOB) have been widely used in different motion control applications ((White et al., 1998), (Fujiyama et al., 2000), (Iwasaki et al., 1999)) for determining the disturbance forces, such as friction forces. However, the performance enhancement of these DOB strategies may lead to smaller stability margins for the motion control ((Komada et al., 2000)); therefore a robust design with respect to the environmental disturbances and model uncertainties is needed. Previous results on robustly stable DOB ((Kempf and Kobayashi, 1999), (Eom et al., 2000), (Güvenc and Güvenc, 2001), (Ryoo et al., 2004)) are based on linear robust control techniques. Some non-linear DOB have been developed for the estimation of harmonic disturbance signals ((Chen et al., 2000), (Liu and Peng, 2000)).

Various strategies have also been considered for force-sensor-less control schemes estimating the external force. (Eom et al., 1998) proposes an adaptive disturbance observer scheme, and (Ohishi et al., 1991) and (Ohishi et al., 1992) propose an H^∞ estimation algorithm. In (Alcocer et al., 2003), a control strategy called ”force observer” is introduced. This design uses an observer-type algorithm for the estimation of the exogenous force. The drawback of this approach is that it assumes perfect knowledge of the model of the system.

In parallel with force estimation strategies, based on disturbance observers, another approach using sensor fusion has been developed to diminish the noise levels of the force sensors. In (Kröger et al., 2007), force and acceleration sensors are used, while in (Garcia et al., 2008), data from force sensors and position encoders are fused. Sensor fusion provides better qualitative results than obtained by employing more expensive force sensors.

Here, we present a control approach for bilateral teleoperation with an estimation strategy for external forces acting on the slave robot with a load with unknown mass. This method extends a result presented in (Lichiardopol et al., 2008), which considered human-robotic co-manipulation problem. The proposed algorithm is robust for large uncertainties in the mass of the load.

The paper is structured as follows. Section 2 presents the problem formulation and in Section 3 we describe the control strategy we propose. In Section 4, we apply the algorithm to a 1-DOF master-slave robotic setup. In the final section of the paper, the conclusions and some perspectives on future work are discussed.

2 PROBLEM STATEMENT

The problem that is tackled in this paper is that of bilateral teleoperation in force sensor-less robotic setups. We assume that the slave robot is generally carrying a load (e.g. tool or product) and that the exogenous forces act on the slave or on the load. For 1-DOF robotic setups, this assumption does not induce any loss of generality. We consider the case in which no force sensor is present to measure the exogenous force directly. Moreover, we consider the realistic case in which the mass of the load is not known exactly which further challenges the estimation of the exogenous force. In order to solve this problem, we propose the design of a force estimator which is robust to the uncertainties in the mass of the load. In order to achieve the teleoperation, the position of the slave robot must track the position of the master robot. For the sake of simplicity, we have considered identical master and slave robots. The extension towards different inertias for the master and slave robots is relatively straightforward by introducing some scaling factors for the forces applied on the master and slave robots. In Figure 1, the block diagram of the teleoperation setup is presented with the blocks *Master* and *Slave* representing the dynamics of the master and the slave robot respectively and the block C representing the control algorithm for bilateral teleoperation. The signals F_H and F_E represent the human and the environmental force respectively; x_M and x_S are the positions of the master and the slave robots, u is the control signal for the slave robot and $\hat{F}_E$ is the signal that makes transparent the environmental force acting on the slave robot F_E to the master cockpit. We adopt the assumption that the only measurements available are the position of the joint(s) and hence we aim to construct an output-feedback control strategy.

The objective of this paper is to design the controller C such that the following goals are met:

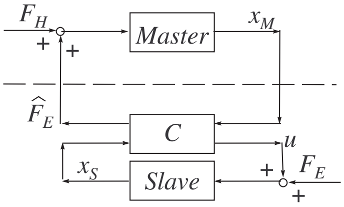


Figure 1: Problem Setup.

- the position of the slave robot is tracking the position of the master robot;
- an accurate estimate of the environmental force is transmitted to the master robot;
- the overall system is stable.

3 CONTROL DESIGN

Due to the uncertainties in the model of the slave robot we can not estimate the unknown environmental force and track the master robot position at the same time (unknown inertia and only position measurement available do not allow simultaneous force estimation and position tracking). Therefore, we are proposing a switching controller based on a cyclic algorithm. During one cycle of duration T, we will have two phases as in Figure 2:

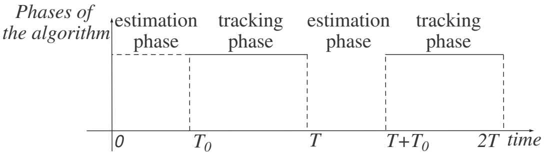


Figure 2: Temporal division of the control strategy.

1. Estimation of the environmental force;
2. Position tracking.

During the first phase, which last for a period of T_0 ($T_0 < T$), the controller will behave as a force estimator. Here we are using the force observer introduced

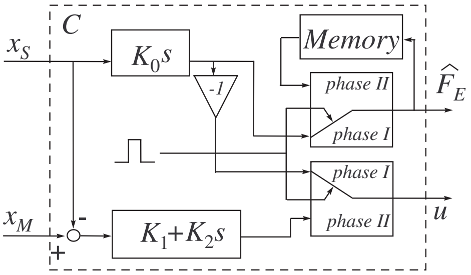


Figure 3: Controller Design.

in (Lichiardopol et al., 2008) to estimate the external force which will be used for the purpose of haptic feedback and during the second phase we are keeping the estimated force constant. In the second phase, we are using a PD controller for the slave robot to track the position of the master robot. In Figure 3, we present the block diagram representation of the controller where the controller blocks are represented by their transfer functions in the Laplace domain ($s \in \mathbb{C}$) and the block called *Memory* saves the last estimate of the environmental force at the end of the first phase and provides the same constant output during the entire second phase. The switches in Figure 3 are set on positions corresponding to the first phase of the algorithm.

In the following section, we study the stability for the closed-loop system (including force estimation error dynamics and tracking error dynamics).

3.1 Description

For the purpose of stability analysis, we first formulate the model of the error dynamics. In order to obtain the error dynamics, the dynamics of the master and slaves robots are needed in both phases. During the first phase ($kT \le t < kT + T_0$, $k \in \mathbb{N}$), the model dynamics are:

$$\begin{cases} m\ddot{x}_M = F_H(t) + K_0\dot{x}_S \\ m\ddot{x}_S = F_E(t) - K_0\dot{x}_S \end{cases}, \tag{1}$$

where x_M and x_S are the position of the master and the slave robots respectively, F_H and F_E are the human and the environmental force, respectively, m is the unknown inertia of the robot with the load (the mass is assumed to be bounded $m \in [M_{min}, M_{max}]$) and parameter K_0 is a scalar that defines the force estimation algorithm and is chosen such that the estimation of the force has converged in the interval $[kT, kT + T_0]$.

In the second phase of the algorithm ($kT + T_0 \le t < (k+1)T$, $k \in \mathbb{N}$), the system behavior is described by:

$$\begin{cases} m\ddot{x}_M = F_H(t) + \hat{F}_E(KT + T_0) \\ m\ddot{x}_S = F_E(t) + K_1(x_M - x_S) + K_2(\dot{x}_M - \dot{x}_S) \end{cases}, \tag{2}$$

where K_1 and K_2 define the PD controller that ensures the tracking of the master robot position by the slave robot (these parameters are chosen such that the polynomial $ms^2 + K_2 s + K_1$ is Hurwitz $\forall m \in [M_{min}, M_{max}]$) and $\hat{F}_E(KT + T_0)$ is the estimation of the environmental force at the end of the first phase.

In the sequel, we assume that the exogenous forces acting on the system (human force F_H and environmental force F_E) and their derivatives are bounded.

3.2 Stability Analysis

Let us define the vector $\varepsilon = [e_x, \dot{e}_x]^T = [x_M(t) - x_S(t), \dot{x}_M(t) - \dot{x}_S(t)]^T$, which contains the position and the velocity tracking errors, and the force estimation error $e_F = \hat{F}_E - F_E$. Then the force error dynamics are described by:

$$\dot{e}_F = -\frac{K_0}{m} e_F - \dot{F}_E, \tag{3}$$

during the first step of the algorithm ($kT \le t < kT + T_0$, $k \in \mathbb{N}$) and

$$\dot{e}_F = -\dot{F}_E, \tag{4}$$

during the second phase ($kT + T_0 \le t < (k+1)T$, $k \in \mathbb{N}$).

The position error dynamics is represented by:

$$\dot{\varepsilon} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \varepsilon + \begin{pmatrix} 0 & 0 \\ \frac{1}{m} & \frac{1}{m} \end{pmatrix} \begin{pmatrix} F_H \\ F_E \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{2}{m} \end{pmatrix} e_F, \tag{5}$$

for $t \in [kT, kT + T_0)$, with $k \in \mathbb{N}$ and

$$\dot{\varepsilon} = \begin{pmatrix} 0 & 1 \\ -\frac{K_1}{m} & -\frac{K_2}{m} \end{pmatrix} \varepsilon + \begin{pmatrix} 0 \\ \frac{1}{m} \end{pmatrix} F_H + \begin{pmatrix} 0 \\ \frac{1}{m} \end{pmatrix} e_F, \tag{6}$$

for $t \in [kT + T_0, (k+1)T)$, with $k \in \mathbb{N}$.

The goal of this section is to prove that the overall system presented in Figure 1 is input-to-state stable with respect to the inputs F_H and F_E. For this we are going to use a result introduced in (Jiang et al., 1996) that states that the series connection of two input-to-state stable systems is also an input-to-state stable system. In the sequel, this proof will be split into two parts:

- Prove that the force error dynamics are stable with respect to the input $\dot{F}_E$;
- Prove that the position error dynamics are stable with respect to the inputs F_H, F_E and e_F.

3.2.1 Input-to-state Stability of the Force Estimation Error Dynamics

The stability analysis of the force error dynamics is done by studying the discrete-time input-to-state stability (ISS) property of the system (3)-(4). For this we will now exploit an exact discretisation of the system at the sampling instances kT.

The solution of system (3) at time $t = kT + T_0$, with $k \in \mathbb{N}$, is:

$$\begin{aligned} e_F(kT+T_0) &= e^{-\frac{K_0}{m}T_0} e_F(kT) + \\ &+ \int_0^{T_0} e^{-\frac{K_0}{m}(T_0-\tau)} \dot{F}_E(kT+\tau)d\tau. \end{aligned} \tag{7}$$

The solution of system (4) at time $t = (k+1)T$, with $k \in \mathbb{N}$, is:

$$\begin{aligned} e_F((k+1)T) &= e_F(kT+T_0) \\ &- \int_0^{T-T_0} \dot{F}_E(kT+T_0+\tau)d\tau. \end{aligned} \tag{8}$$

Define the sampled force estimation error dynamics $e_k := e_F(kT)$, with $k \in \mathbb{N}$. Combining relations (7) and (8), one can obtain the discrete-time force estimation error dynamics:

$$e_{k+1} = e^{-\frac{K_0}{m}T_0} e_k + w_k, \tag{9}$$

with $w_k = \int_0^{T_0} e^{-\frac{K_0}{m}(T_0-\tau)} \dot{F}_E(kT+\tau)d\tau - \int_0^{T-T_0} \dot{F}_E(kT+T_0+\tau)d\tau$. The system (9) is input-to-state stable with respect to the input w_k because $\left| e^{-\frac{K_0}{m}T_0} \right| < 1$, since the parameters K_0, T_0 and the inertia m are positive. Note that w_k is bounded for any bounded $\dot{F}_E(t)$ and bounded T_0.

Now we exploit a result in (Nešić et al., 1999) that says that if the discrete-time dynamics is ISS and the intersample behavior is uniformly globally bounded over T,then the corresponding sampled-data is ISS. The fact that the intersample behavior is uniformly globally bounded over T directly follows from (3),(4) with $\dot{F}_E$ bounded, since

$$e(t) = \begin{cases} \begin{array}{l} e^{-\frac{K_0}{m}(t-kT)} e_F(kT) \\ + \int_{kT}^{t} e^{-\frac{K_0}{m}(t-\tau)} \dot{F}_E(\tau)d\tau \end{array} & ,kT \leq t < kT+T_0 \\ \begin{array}{l} e_F(kT+T_0) \\ - \int_{kT+T_0}^{t} \dot{F}_E(\tau)d\tau \end{array} & ,kT+T_0 \leq t < (k+1)T \end{cases}. \tag{10}$$

3.2.2 Input-to-state Stability of the Tracking Error Dynamics

Similarly to the study of the force estimation error dynamics, we evaluate the input-to-state stability property of the tracking error dynamics with respect to the inputs F_H, F_E and e_F.

The solution of system (5) at time $t = kT + T_0$, with $k \in \mathbb{N}$, is:

$$\begin{aligned} \varepsilon(kT+T_0) = e^{A_1T_0}\varepsilon(kT) + \int_0^{T_0} e^{A_1(T_0-\tau)} B_{11} u(kT+\tau)d\tau \\ + \int_0^{T_0} e^{A_1(T_0-\tau)} B_{12} e_F(kT+\tau)d\tau, \end{aligned} \tag{11}$$

where $A_1 = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$, $B_{11} = \begin{pmatrix} 0 & 0 \\ \frac{1}{m} & \frac{1}{m} \end{pmatrix}$, $B_{12} = \begin{pmatrix} 0 \\ \frac{2}{m} \end{pmatrix}$ and $u(t) = \begin{pmatrix} F_H(t) \\ F_E(t) \end{pmatrix}$.

The solution of system (6) at time $t = (k+1)T$, with $k \in \mathbb{N}$, is:

$$\begin{aligned} \varepsilon((k+1)T) &= e^{A_2(T-T_0)}\varepsilon(kT+T_0) \\ &+ \int_0^{T-T_0} e^{A_2(T-T_0-\tau)} B_{21} F_H(kT+T_0+\tau)d\tau \\ &+ \int_0^{T-T_0} e^{A_2(T-T_0-\tau)} B_{22} e_F(kT+T_0+\tau)d\tau, \end{aligned} \tag{12}$$

where $A_2 = \begin{pmatrix} 0 & 1 \\ -\frac{K_1}{m} & -\frac{K_2}{m} \end{pmatrix}$, $B_{21} = \begin{pmatrix} 0 \\ \frac{1}{m} \end{pmatrix}$ and $B_{22} = \begin{pmatrix} 0 \\ \frac{1}{m} \end{pmatrix}$.

Let us define

$$\begin{aligned} \omega_k := e^{A_2(T-T_0)} \big(\int_0^{T_0} e^{A_1(T_0-\tau)} B_{11} u(kT+\tau)d\tau \\ + \int_0^{T_0} e^{A_1(T_0-\tau)} B_{12} e_F(kT+\tau)d\tau \big) \\ + \int_0^{T-T_0} e^{A_2(T-T_0-\tau)} B_{21} F_H(kT+T_0+\tau)d\tau \\ + \int_0^{T-T_0} e^{A_2(T-T_0-\tau)} B_{22} e_F(kT+T_0+\tau)d\tau \end{aligned} \tag{13}$$

and $\varepsilon_k = \varepsilon(kT)$, with $k \in \mathbb{N}$. Combining relations (11) and (12), we obtain the discrete-time system:

$$\varepsilon_{k+1} := e^{A_2(T-T_0)} e^{A_1T_0} \varepsilon_k + \omega_k, \tag{14}$$

where ω_k is bounded for all k, since T, T_0 are bounded, F_E, F_H are bounded by assumption and e_F is bounded due to the fact that the force estimation error dynamics is ISS with respect to $\dot{F}_E$.

Next, we study the input-to-state stability property of the system (14) with respect to the input ω_k. But before we carry on this step, we need to evaluate the matrix $Q = e^{A_2(T-T_0)} e^{A_1T_0}$. Namely, input-to-state stability of (14) implies, firstly, the global uniform asymptotic stability of $\varepsilon = 0$ when the input ω_k is zero and the boundness of the error ε for bounded input.

For the evaluation of the matrix Q, two exponential matrices must be determined; as the matrix A_1T_0 depends only on known parameters, we can easily determine its exponential:

$$E_1 := e^{A_1T_0} = \begin{pmatrix} 1 & T_0 \\ 0 & 1 \end{pmatrix}. \tag{15}$$

In order to compute the exponential of matrix $P = A_2(T - T_0)$, we are using a procedure similar to the one introduced in (Gielen et al., 2008), which employs the Cayley-Hamilton theorem, which says that if $p(\lambda) = \det(\lambda I_n - A)$, with I_n the $n \times n$ identity matrix, is the characteristic polynomial of a matrix $A \in \mathbb{R}^{n\times n}$ then $p(A) = 0$. This means that given the matrix P, for any $i \geq 2$, there exists a set of coefficients $a_i, b_i \in \mathbb{R}$ such that the i^{th} power of P can be expressed in terms of its first two powers:

$$P^i = a_i I_2 + b_i P. \tag{16}$$

Let us now exploit (16) to determine the exponential of the matrix P:

$$e^P = \sum_{i=0}^{\infty} \frac{P^i}{i!} = \sum_{i=0}^{\infty} \frac{1}{i!}(a_i I_2 + b_i P), \tag{17}$$

or

$$e^P = \left(\sum_{i=0}^{\infty} \frac{a_i}{i!}\right) I_2 + \left(\sum_{i=0}^{\infty} \frac{b_i}{i!}\right) P. \tag{18}$$

Using the expression of A_2, we can decompose P as follows: $P = U + \frac{1}{m}L$, where

$$U = \begin{pmatrix} 0 & T - T_0 \\ 0 & 0 \end{pmatrix} \tag{19}$$

and

$$L = \begin{pmatrix} 0 & 0 \\ -K_1(T - T_0) & -K_2(T - T_0) \end{pmatrix}. \tag{20}$$

Consequently, the expression for the exponential matrix becomes:

$$e^P = \left(\sum_{i=0}^{\infty} \frac{a_i}{i!}\right) I_2 + \left(\sum_{i=0}^{\infty} \frac{b_i}{i!}\right) U + \frac{1}{m}\left(\sum_{i=0}^{\infty} \frac{b_i}{i!}\right) L. \tag{21}$$

Let us now define the following scalars:

$$\underline{\alpha} = \min_{m \in [M_{min}, M_{max}]} \left(\sum_{i=0}^{\infty} \frac{a_i}{i!}\right), \tag{22}$$

$$\overline{\alpha} = \max_{m \in [M_{min}, M_{max}]} \left(\sum_{i=0}^{\infty} \frac{a_i}{i!}\right), \tag{23}$$

$$\underline{\beta} = \min_{m \in [M_{min}, M_{max}]} \left(\sum_{i=0}^{\infty} \frac{b_i}{i!}\right), \tag{24}$$

and

$$\overline{\beta} = \max_{m \in [M_{min}, M_{max}]} \left(\sum_{i=0}^{\infty} \frac{b_i}{i!}\right). \tag{25}$$

Given the fact that $m \in [M_{min}, M_{max}]$, we can define the scalars $\underline{\gamma} = \frac{1}{M_{max}}$ and $\overline{\gamma} = \frac{1}{M_{min}}$.

Then there always exist $\zeta_1, \zeta_2, \zeta_3 \in [0,1]$ such that:

$$\left(\sum_{i=0}^{\infty} \frac{a_i}{i!}\right) = \zeta_1 \underline{\alpha} + (1 - \zeta_1)\overline{\alpha}, \tag{26}$$

$$\left(\sum_{i=0}^{\infty} \frac{b_i}{i!}\right) = \zeta_2 \underline{\beta} + (1 - \zeta_2)\overline{\beta} \tag{27}$$

and

$$\frac{1}{m} = \zeta_3 \underline{\gamma} + (1 - \zeta_3)\overline{\gamma}. \tag{28}$$

Introducing relations (26), (27) and (28) into expression (21) leads to:

$$\begin{aligned} e^P = {} & (\zeta_1 \underline{\alpha} + (1 - \zeta_1)\overline{\alpha}) I_2 + \left(\zeta_2 \underline{\beta} + (1 - \zeta_2)\overline{\beta}\right) U \\ & + \left(\zeta_3 \underline{\gamma} + (1 - \zeta_3)\overline{\gamma}\right) \left(\zeta_2 \underline{\beta} + (1 - \zeta_2)\overline{\beta}\right) L, \end{aligned} \tag{29}$$

for some $\zeta_1, \zeta_2, \zeta_3 \in [0,1]$.

Let us define the matrices $\Gamma_1 = 3\underline{\alpha}E_1$, $\Gamma_2 = 3\overline{\alpha}E_1$, $\Gamma_3 = 3\underline{\beta}UE_1$, $\Gamma_4 = 3\overline{\beta}UE_1$, $\Gamma_5 = 3\underline{\beta\gamma}LE_1$, $\Gamma_6 = 3\underline{\beta}\overline{\gamma}LE_1$, $\Gamma_7 = 3\overline{\beta}\underline{\gamma}LE_1$ and $\Gamma_8 = 3\overline{\beta}\overline{\gamma}LE_1$, and the scalars $\rho_1 = \frac{\zeta_1}{3}$, $\rho_2 = \frac{1-\zeta_1}{3}$, $\rho_3 = \frac{\zeta_2}{3}$, $\rho_4 = \frac{1-\zeta_2}{3}$, $\rho_5 = \frac{\zeta_2\zeta_3}{3}$, $\rho_6 = \frac{\zeta_2(1-\zeta_3)}{3}$, $\rho_7 = \frac{(1-\zeta_2)\zeta_3}{3}$, $\rho_8 = \frac{(1-\zeta_2)(1-\zeta_3)}{3}$. This means that the expression of matrix Q is equivalent to:

$$Q = \sum_{i=1}^{8} \rho_i \Gamma_i, \tag{30}$$

with $\sum_{i=1}^{8} \rho_i = 1$.

Thus we have now found the generators for a convex set that overapproximates the matrix Q, with the uncertain parameter m. Notice that $\sum_{i=0}^{\infty} \frac{a_i}{i!}$ and $\sum_{i=0}^{\infty} \frac{b_i}{i!}$ are infinite sums and will in practice be approximated by finite sums of length N. Next, we provide an explicit upper bound on the 2-norm of the approximation error induced by such truncation.

Theorem 1. *Consider an integer $N \in \mathbb{N}$ and a real positive scalar ϑ such that*

- $\mu = \sqrt{\frac{\lambda_{max}}{\vartheta}} < 1$, *where*

$$\lambda_{max} = \max_{m \in [M_{min}, M_{max}]} \left\{eig(P^T P)\right\}, \tag{31}$$

- $\forall i \geq N$, $\sqrt{\vartheta^i} < i!$.

Then:

$$\left\|\sum_{i=N}^{\infty} \frac{P^i}{i!}\right\|_2 \leq \frac{\mu^N}{1 - \mu}. \tag{32}$$

Proof.

$$\begin{aligned} \left\|\textstyle\sum_{i=N}^{\infty} \frac{P^i}{i!}\right\|_2 & \leq \textstyle\sum_{i=N}^{\infty} \left\|\frac{P^i}{i!}\right\|_2 \\ \leq \textstyle\sum_{i=N}^{\infty} \frac{\left\|P^i\right\|_2}{i!} & \leq \textstyle\sum_{i=N}^{\infty} \frac{\sqrt{(\lambda_{max})^i}}{i!}, \end{aligned} \tag{33}$$

where the inequality $\left\|A^i\right\|_2^2 \leq \|A\|_2^2 \times \ldots \times \|A\|_2^2 = \max(eig((A^T A))^i$ has been used. Using the property that $\forall a \in \mathbb{R}^+$, $\exists N \in \mathbb{N}$ such that $\forall i \geq N$, $\sqrt{a^i} < i!$, inequality (33) becomes:

$$\left\|\sum_{i=N}^{\infty} \frac{P^i}{i!}\right\|_2 \leq \sum_{i=N}^{\infty} \frac{\sqrt{(\lambda_{max})^i}}{i!} \leq \sum_{i=N}^{\infty} \mu^i. \tag{34}$$

Let us now employ the known result of convergence of geometric series which states that $\forall a \in [0,1)$, $\lim_{n \to \infty} \sum_{i=0}^{n} a^i = \lim_{n \to \infty} \frac{1-a^{n+1}}{1-a} = \frac{1}{1-a}$.

$$\left\|\sum_{i=N}^{\infty} \frac{P^i}{i!}\right\|_2 \leq \frac{\mu^N}{1 - \mu}. \tag{35}$$

Using Theorem 1, we can choose N such that the approximation error is small (even as low as the machine accuracy).

In the next theorem , we provide a LMI-based stability conditions for the discrete-time tracking error dynamics to be ISS with respect to the input ω_k.

Theorem 2. *Consider the discrete-time system (14). If there exists a matrix $\Omega = \Omega^T > 0$ and scalar $\varsigma \in (0,1)$, such that the following linear matrix inequalities are satisfied:*

$$\Gamma_i{}^T \Omega \Gamma_i - \Omega \leq -\varsigma\Omega, i \in \{1,\ldots,8\} \tag{36}$$

where Γ_i are defined above, then the system (14) is ISS with respect to the input ω_k.

Proof. Using the Schur complement, relations (36) can be written as:

$$\begin{pmatrix} -\Omega & \Gamma_i{}^T\Omega \\ \Omega\Gamma_i & -\Omega \end{pmatrix} \leq -\varsigma\Omega, i \in \{1,\ldots,8\}. \tag{37}$$

Multiplying every inequality (37) with ρ_i and summing them up, we obtain:

$$\begin{array}{c} \begin{pmatrix} -\Omega\sum_{i=1}^{8}\rho_i & \sum_{i=1}^{8}\rho_i\Gamma_i{}^T\Omega \\ \Omega\sum_{i=1}^{8}\rho_i\Gamma_i & -\Omega\sum_{i=1}^{8}\rho_i \end{pmatrix} \\ \leq -\varsigma\Omega\sum_{i=1}^{8}\rho_i, \end{array} \tag{38}$$

which according to equation (30) is:

$$\begin{pmatrix} -\Omega & Q^T\Omega \\ \Omega Q & -\Omega \end{pmatrix} \leq -\varsigma\Omega, \tag{39}$$

or

$$Q^T\Omega Q - \Omega \leq -\varsigma\Omega. \tag{40}$$

Let the candidate Lyapunov function be $V_k = (\varepsilon_k)^T\Omega\varepsilon_k$. We compute $\triangle V_k = V_{k+1} - V_k$:

$$\begin{array}{c} \triangle V_k = (\varepsilon_k)^T Q^T \Omega Q \varepsilon_k - (\varepsilon_k)^T \Omega \varepsilon_k \\ +2(\varepsilon_k)^T Q^T \Omega \omega_k + (\omega_k)^T \Omega \omega_k, \end{array} \tag{41}$$

which according to (40) gives:

$$\triangle V_k \leq -\varsigma(\varepsilon_k)^T\Omega\varepsilon_k + 2(\varepsilon_k)^T Q^T \Omega\omega_k + (\omega_k)^T\Omega\omega_k \tag{42}$$

After some straightforward computations, we can show that:

$$\|\varepsilon\|_2 \geq \frac{2}{\varsigma}\sqrt{\frac{\lambda_{max}}{\lambda_{min}}} \sup_{k\in\mathbb{N}}(\omega_k) \Rightarrow \triangle V \leq -\frac{\varsigma}{2}\|\varepsilon\|_2^2, \tag{43}$$

where λ_{max} and λ_{min} are the largest and the smallest eigenvalues of matrix Ω, respectively.

(43) implies that system (14) is input-to-state stable with respect to the input ω_k; see (Jiang and Wang, 2001) for sufficient condition for the ISS of discrete-time systems.

Remark 1. *For the sake of simplicity, Theorem 2 is based on a common quadratic ISS Lyapunov function $V = \varepsilon^T\Omega\varepsilon$. Alternatively, a parameter-dependent Lyapunov function approach could straight-forwardly be exploited to formulate less conservative stability conditions.*

The LMIs (36) are defined for the non-truncated Γ_i, but in practice we evaluate the vertex matrices using a truncation after N iterations as provided by Theorem 1. The errors can be as low as the machine accuracy, just as the errors obtained from the numerical solver of the LMIs. Moreover, we can gain some robustness for these evaluation errors if the scalar ς is chosen greater than $\varepsilon_\varsigma > 0$.

The last part of the study of the ISS property of the tracking error dynamics is to analyze the intersample behavior. Using Theorem 2, we can prove that the error dynamics are ISS on the sampling instance $t = kT$, with $k \in \mathbb{N}$. Given the choice of the parameters K_1 and K_2 such that the system (6) is Hurwitz for all $m \in [M_{min}, M_{max}]$, we can conclude that during the second phase ($t \in [kT + T_0, (k+1)T)$) the tracking error dynamics are bounded. In order to prove the stability of the overall continuous-time system, we need to show that the position error dynamics are also bounded for $t \in (kT, kT + T_0)$.

The solution of system (5), for $t \in (kT, kT + T_0)$ is:

$$\begin{array}{c} \varepsilon(kT+t) = \begin{pmatrix} 1 & t \\ 0 & 1 \end{pmatrix} \varepsilon(kT) \\ +\int_0^t \begin{pmatrix} 1 & t-\tau \\ 0 & 1 \end{pmatrix} B_{11}u(kT+\tau)d\tau \\ +\int_0^t \begin{pmatrix} 1 & t-\tau \\ 0 & 1 \end{pmatrix} B_{12}e_F(kT+\tau)d\tau. \end{array} \tag{44}$$

As the human force and the environmental force are bounded, we can define $F = \max_{t\in(kT,kT+T_0)}(|F_H(t)| + |F_E(t)|)$. In the previous section, we have proven that the force estimation error dynamics are ISS and consequently are bounded; therefore there exists $E_F = \max_{t\in(kT,kT+T_0)}(|e_F(t)|)$. Considering the three terms from relation (44), we can conclude that the first one is bounded due to the boundness of the discrete-time error dynamics, the second term:

$$\left|\int_0^t \begin{pmatrix} 1 & t-\tau \\ 0 & 1 \end{pmatrix} B_{11}u(kT+\tau)d\tau\right| \leq \left|\frac{F}{m}\begin{pmatrix} \frac{T_0^2}{2} \\ T_0 \end{pmatrix}\right|, \tag{45}$$

and the third:

$$\left|\int_0^t \begin{pmatrix} 1 & t-\tau \\ 0 & 1 \end{pmatrix} B_{12}e_F(kT+\tau)d\tau\right| \leq \left|\frac{2E_F}{m}\begin{pmatrix} \frac{T_0^2}{2} \\ T_0 \end{pmatrix}\right|. \tag{46}$$

Therefore, we can conclude that the position error dynamics are also bounded for $t \in (kT, kT + T_0)$. Similarly, to the force estimation error dynamics, we can employ the result from (Nešić et al., 1999) to prove that the tracking error dynamics is ISS because the discrete-time tracking error dynamics is ISS and the intersample behavior is uniformly globally bounded.

Since the force estimation error dynamics (e_F) is ISS with respect to the input $\dot{F}_E$ and the tracking error dynamics (ε) is ISS with respect to the inputs F_H, F_E and e_F, we use the result introduced by (Jiang et al., 1996) concerning the series connection of ISS systems to conclude that the closed-loop system from Figure 1 with the controller C with the block diagram representation from Figure 3 is ISS with respect to the inputs F_H, F_E and $\dot{F}_E$.

Remark 2. *By studying the ISS property of the system, one can observe that the steady-state force estimation and tracking errors can be influenced by tuning parameters T, T_0, K_0, K_1 and K_2. The algorithm provides a deeper insight into these relations. If we consider the converging manifold that bounds the error signal we can determine these parameters in accordance with the desired convergence rate.*

Remark 3. *In case the environmental force F_E is constant, i.e. $\dot{F}_E = 0$, the force estimation dynamics are globally exponentially stable and the tracking error dynamics is ISS with respect to the inputs F_H and F_E. this means that "perfect" haptic feedback is provided and that bounded tracking error remain; therefore the closed loop is stable.*

Remark 4. *The exact "tracking" regulation with respect to what the human has in mind is up to the human (since the human is in charge of the ultimate positioning).*

4 ILLUSTRATIVE EXAMPLE

In this section, we will apply the control design proposed in the previous section to a master-slave teleoperation setup consisting of two 1-DOF robots. The inertia of the robots is considered to be in the range $m \in [0.1, 10]kg$.

The "human" controller has been emulated by a linear transfer function:

$$H(s) = \frac{K_d(T_d s + 1)}{T_{PL} s + 1} = \frac{500(1+s)}{0.1s+1}, \qquad (47)$$

with saturation at $\pm 100N$. Here we use real human parameters, since the human movement is lower than $6Hz$. Also to comply with the human sensing range, which is between $0Hz$ and $40-400Hz$ depending on the amplitude of the input signal, we have chosen the parameters are the cycle period of the controller $T = 0.01s$ and the duration of the first stage $T_0 = T/2 = 0.005s$. The force estimator acting in the first phase of the algorithm is defined by parameter $K_0 = 10^5$. The tracking PD controller which is active during the second phase has the parameters $K_1 = 200$ and $K_2 = 1$.

In Figure 4, we simulated the position tracking of the algorithm when the "human" is performing a movement from $0m$ to $0.25m$ on the master robot and a sinusoidal external force with amplitude $0.5N$ and frequency $1Hz$ is disturbing the slave robot. The dotted line is the position of the master and the solid line si the position of the slave.

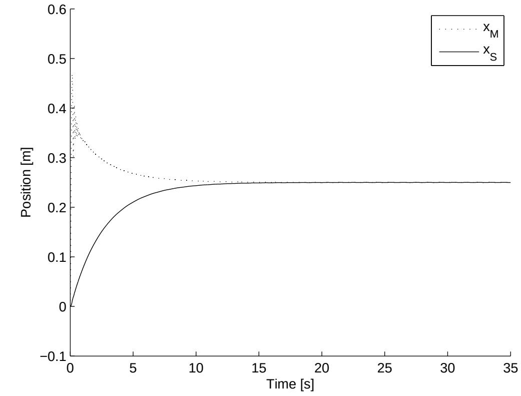


Figure 4: Position tracking.

One can observe that because no disturbance rejection controller is implemented, the external force is stopping the position signal to settle at $0.25m$. In Figure 5, a zoomed in version of the Figure 4 that emphasizes this aspect is presented.

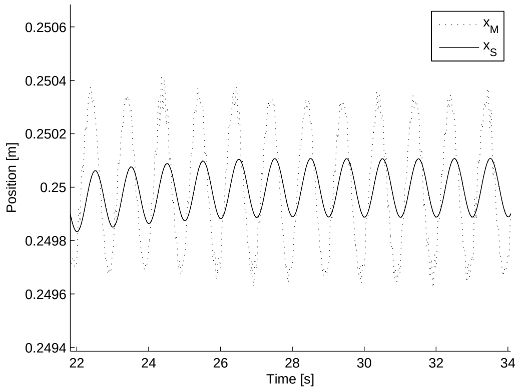


Figure 5: Position tracking.

5 CONCLUSIONS AND PERSPECTIVES

In this paper, we have introduced a new control algorithm for bilateral teleoperation of 1-DOF robots in force-sensorless setups using a switching strategy between a force estimating controller and a tracking controller. This switching algorithm guarantees both the estimation of the environmental force acting upon the slave robot (to be used in haptic feedback) in the absence of force sensors and the convergence of the tracking errors in the case of external perturbations. We note that the ultimate position setting is the responsibility of the human, as he is in charge of the position of the master robot. Finally, we remark that the proposed algorithm is robust for unknown loads to be carried by the slave robot.

Future perspectives of this work we will mainly focus on an extension to multi-degree-of-freedom robots and also to robots with nonlinear dynamics.

REFERENCES

Alcocer, A., Robertsson, A., Valera, A., and Johansson, R. (2003). Force estimation and control in robot manipulators. *Proceedings of the* 7^{th} *Symposium on Robot Control (SYROCO'03)*, pages 31–36.

Chen, X., Komada, S., and Fukuda, T. (2000). Design of a nonlinear disturbance observer. *IEEE Transactions on Industrial Electronics*, 47:429–436.

Eom, K. S., Suh, I. H., and Chung, W. K. (2000). Disturbance observer based path tracking of robot manipulator considering torque saturation. *Mechatronics*, 11:325–343.

Eom, K. S., Suh, I. H., Chung, W. K., and Oh, S. R. (1998). Disturbance observer based force control of robot manipulator without force sensor. *Proceedings of IEEE International Conference on Robotics and Automation*, pages 3012–3017.

Fujiyama, K., Katayama, R., Hamaguchi, T., and Kawakami, K. (2000). Digital controller design for recordable optical disk player using disturbance observer. *Proceedings of the International Workshop on Advanced Motion Control*, pages 141–146.

Garcia, J. G., Robertsson, A., Ortega, J. G., and Johansson, R. (2008). Sensor fusion for compliant robot motion control. *IEEE Transations on Robotics*, 24:430–441.

Gielen, R., Olaru, S., and Lazar, M. (2008). On polytopic approximations as a modelling framework for networked control systems. *Proceedings of the International Workshop on Assesment and Future Direction of NMPC, Pavia, Italy*.

Güvenc, B. A. and Güvenc, L. (2001). Robustness of disturbance observers in the presence of structured real parametric uncertainty. *Proceedings of the American Control Conference*, pages 4222–4227.

Hokayem, P. F. and Spong, M. W. (2006). Bilateral teleoperation: A historical survey. *Automatica*, 42:2035–2057.

Iwasaki, M., Shibata, T., and Matsui, N. (1999). Disturbance-observer-based nonlinear friction compensation in table drive system. *IEEE Transactions on Mechatronics*, 4:3–8.

Jiang, Z. P., Mareels, I. M. Y., and Wang, Y. (1996). A Lyapunov formulation of the nonlinear small-gain theorem for interconnected ISS systems. *Automatica*, 32:1211–1215.

Jiang, Z. P. and Wang, Y. (2001). Input-to-state stability for discrete-time nonlinear systems. *Automatica*, 37:857–869.

Kempf, C. J. and Kobayashi, S. (1999). Disturbance observer and feedforward design for high-speed direct-drive position table. *IEEE Transactions on Control System Technology*, 7:513–526.

Komada, S., Machii, N., and Hori, T. (2000). Control of redundant manipulators considering order of disturbance observer. *IEEE Transactions on Industrial Electronics*, 47:413–420.

Kröger, T., Kubus, D., and Wahl, F. M. (2007). Force and acceleration sensor fusion for compliant manipulation control in 6 degrees of freedom. *Advanced Robotics*, 21:1603–1616.

Lawrence, D. (1993). Stability and transparency in bilateral teleoperation. *IEEE Transactions on Robotics and Automation*, 9:624–637.

Lichiardopol, S., van de Wouw, N., and Nijmeijer, H. (2008). Boosting human force: A robotic enhancement of a human operator's force. *Proceedings of International Conference on Decision and Control*, pages 4576–4581.

Liu, C. S. and Peng, H. (2000). Disturbance observer based tracking control. *Journal of Dynamic Systems, Measurement and Control*, 122:332–335.

Nešić, D., Teel, A. R., and Sontag, E. D. (1999). Formulas relating $\mathscr{KL}$-stability estimates of discrete-time and sampled-data nonlinear systems. *Systems & Control Letters*, 38:49–60.

Ohishi, K., Miyazaki, M., Fujita, M., and Ogino, Y. (1991). H^∞ observer based force control without force sensor. *Proceedings of International Conference on Industrial Electronics, Control and Instrumentation*, pages 1049 – 1054.

Ohishi, K., Miyazaki, M., Fujita, M., and Ogino, Y. (1992). Force control without force sensor based on mixed sensitivity H^∞ design method. *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1356 – 1361.

Ryoo, J. R., Doh, T. Y., and Chung, M. J. (2004). Robust disturbance observer for the track-following control system of an optical disk drive. *Control Engineering Practice*, 12:577–585.

White, M. T., Tomizuka, M., and Smith, C. (1998). Improved track following in magnetic disk drives using a disturbance observer. *IEEE Transactions on Mechatronics*, 5:3–11.

Yokokohji, Y. and Yoshikawa, T. (1994). Bilateral control of master-slave manipulators for ideal kinematics coupling formulation and experiment. *IEEE Transactions on Robotics and Automation*, 10:605–620.

GRASPING WITH VISION DESCRIPTORS AND MOTOR PRIMITIVES

Oliver Kroemer
Max Planck Institute for Biological Cybernetics, Spemannstr. 38, 72076 Tuebingen, Germany
oliverkro@tuebingen.mpg.de

Renaud Detry, Justus Piater
University of Liege, Grande Traverse 10, 4000 Liege, Belgium
renaud.detry@ulg.ac.be, Justus.Piater@ulg.ac.be

Jan Peters
Max Planck Institute for Biological Cybernetics, Spemannstr. 38, 72076 Tuebingen, Germany
jan.peters@tuebingen.mpg.de

Keywords: Dynamical Motor Primitives, Early Cognitive Vision Descriptors, Grasping.

Abstract: Grasping is one of the most important abilities needed for future service robots. Given the task of picking up an object from betweem clutter, traditional robotics approaches would determine a suitable grasping point and then use a movement planner to reach the goal. The planner would require precise and accurate information about the environment and long computation times, both of which may not always be available. Therefore, methods for executing grasps are required, which perform well with information gathered from only standard stereo vision, and make only a few necessary assumptions about the task environment. We propose techniques that reactively modify the robot's learned motor primitives based on information derived from Early Cognitive Vision descriptors. The proposed techniques employ non-parametric potential fields centered on the Early Cognitive Vision descriptors to allow for curving hand trajectories around objects, and finger motions that adapt to the object's local geometry. The methods were tested on a real robot and found to allow for easier imitation learning of human movements and give a considerable improvement to the robot's performance in grasping tasks.

1 INTRODUCTION

Consider the scenario wherein you want to have a humanoid robot grasp an object in a cluttered space. The first stage of most grasp planners determines a suitable grasp location on the object (Saxena et al., 2008; Arimoto, 2008; Bicchi and Kumar, 2000). Having selected a final location and orientation for the hand, the robot must then determine how to execute the grasp so as not to collide with the object or any of the surrounding objects.

The traditional solution for this scenario involves supplying the robot with a CAD model of the objects and a laser scanner or other means (ERFID, previous position, etc.) for obtaining their precise positions. These tools give the robot ample knowledge to apply a planning algorithm that determines a suitable path to the goal. This process relies on precise sensor information and can be very time consuming given a complex scene with numerous possible object collisions to test for at each step. In contrast, humans can perform successful grasps of objects in the periphery of their vision, where visual information is limited.

Taking inspiration from human movements, we propose a reactive method for robots grasping objects in cluttered environments using potential fields based on only a small amount of visual information. Specifically, we present methods for incorporating information derived from Early Cognitive Vision Descriptors (ECVD) (Pugeault, 2008) into the dynamical system motor primitives (DMP) (Schaal et al., 2003) framework. The Early Cognitive Vision system (see Appendix and Figure 2) was chosen since it makes only a few assumptions about the object being grasped, while the motor primitives (see Appendix) were chosen because they generalize well to new situations and can be learned through imitation (Ijspeert et al., 2002). The two frameworks are also compatible

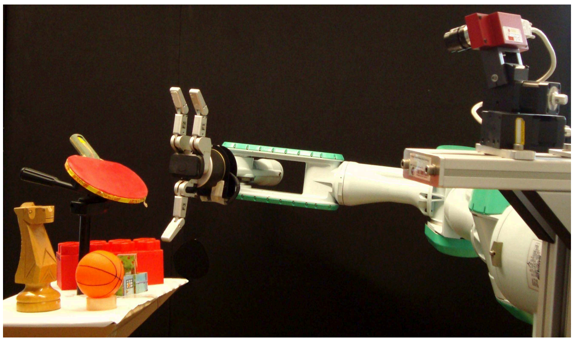


Figure 1: The robot used in our experiments and an example of a grasping task in a cluttered environment.

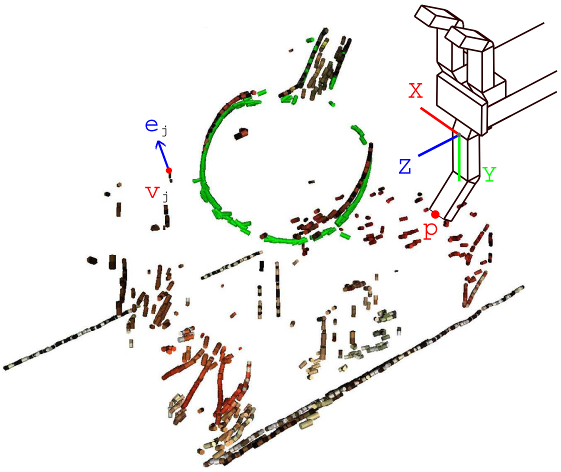


Figure 2: The green ECVD represent the object to be grasped, while the surrounding ECVDs in the scene are clutter. The coordinate frame of the third finger of the Barrett hand (the lower finger in the image) and variables used in section 2 are shown. The *x*-*y*-*z* coordinate system is located at the base of the finger, with *z* orthogonal to the palm, and *y* in the direction of the finger. The marked ECVD on the left signifies the j^{th} descriptor, with its position at $\mathrm{v}_j = (v_{jx}, v_{jy}, v_{jz})^T$, and edge direction $\mathrm{e}_j = (e_{jx}, e_{jy}, e_{jz})^T$ of unit length. The position of the finger tip is given by $\mathrm{p} = (p_x, p_y, p_z)^T$.

with each other and thus straightforward to combine.

The ECVDs were used to elegantly augment the DMPs for grasping tasks, resulting in the robot being able to avoid obstacles, curve its reaching trajectories around the object to grasp, and adapting the fingers to the local geometry of the object.

2 METHODS FOR REACTIVE GRASPING

The methods proposed in this section were inspired by human movements. Human grasping movements can be modeled as two linked components, transportation and finger posture, synchronized by a shared timer or canonical system (Chieffi and Gentilucci, 1993; Oztop and Kawato, 2009). Transportation refers to the actions of the arm in moving the hand, while the finger posture aspect relates to the preshaping and closing of the fingers (Jeannerod, 1997).

Humans perform the reaching/transportation component in a task-specific combination of retina and hand coordinates (Graziano, 2006), which allows for easier specification of object trajectories in a manipulation task than joint coordinates would and results in a reduction in dimensionality. These movements also have curved trajectories that are needed for avoiding obstacles and reaching around objects, which mainly occurs in a planar subspace (Wank et al., 2004).

Similar to the transportation component, the main purpose of the finger posture component is to preshape the hand by extending the fingers sufficiently for them to pass around the object upon approach, and then close on the object simultaneously for a good grasp. Over-extending the fingers is undesirable as it makes collisions with the environment more likely and is therefore usually restricted to situations that present large uncertainties about the object (Oztop et al., 2004; Chieffi and Gentilucci, 1993).

Curved reaching trajectories and preshaping of the hand were incorporated into the robot via a potential field, as described in Sections 2.1 and 2.2. Subsequently, a higher level controller is proposed in Section 2.3, which allows the grasping movements to be interpolated better to new target grasp locations.

2.1 DMP based Attractor Field

The first step towards specifying the grasping movements is to define an attractor field as a DMP that encodes the desired movements given no obstacles. The principal features that need to be defined for these DMPs are 1) the goal positions, and 2) the generic shape of the trajectories to reach the goal.

Determining the goal posture of the hand using the ECV descriptors has been previously investigated in. (Detry et al., 2009). In this work, possible grasp locations were hypothesized from the geometry and color features of the ECVDs, and used to create a kernel density estimate of suitable grasps, which is then refined by attempting grasps to test them.

However, this grasp synthesizer only gives the desired location and orientation of the hand, but leaves finger placement to a secondary finger controller, e.g., (Hsiao et al., 2009; Steffen et al., 2007). Using the ECVDs, the goal position of each finger is approximated by first estimating a contact plane for the object in the finger coordinate system shown in Figure 2. To make it a local approximation, the influence of the i^{th} ECVD is weighted by $w_i = \exp(-\sigma_x^{-2} v_{ix}^2 - \sigma_y^{-2} v_{iy}^2 - \sigma_z^{-2} v_{iz}^2)$, where σ_x, σ_y, and σ_z are length scale constants, and v_i is the position of the ECVD in the finger reference frame. The hand orientation was chosen such that the Z direction of the finger is parallel to the approximated contact plane, which reduces the problem to describing the plane as a line in the 2D X-Y space. The X-Y gradient of the plane is approximated by $\phi = (\sum_{i=1}^N w_i)^{-1} \sum_{i=1}^N w_i \arctan(e_{iy}/e_{ix})$, where N is the number of vision descriptors, and e_i is the direction of the i^{th} edge. The desired Y position of the fingertip is then given by

$$\tilde{p}_y = \frac{\sum_{i=1}^N (w_i v_{iy} - \tan(\phi) w_i v_{ix})}{\sum_{i=1}^N w_i},$$

which can be easily converted to a joint parameter using the inverse kinematics of the hand.

Having determined the goals of both transportation and finger-posture components, the next step is to define the trajectories used to reach these goals. Many of the beneficial traits of human movements, as described earlier, can be transferred to the robot through imitation learning. Learning by imitation involves a human demonstrating a motion and the robot then mimicking the movement. Details for imitation learning with DMPs can be found in (Ijspeert et al., 2002).

We can now combine the goals and imitation learned trajectories to specify the DMPs and thus the attractor fields.

2.2 ECVD based Detractor Fields

Having specified the rudimentary grasping movements, a detractor field is employed to refine the motions in order to include obstacle avoidance for the transportation and ensure that the finger tips do not collide with the object during the hand's approach.

The detractor field will be based on ECVDs, which can be envisioned as small line segments of an object's edges localized in 3D, as shown in Figure 2 for a scene as shown in Figure 1. The detractive potential fields for ECVDs are characterized by two main features; i.e., the detractive forces of multiple ECVDs describing a single line do not superimpose, and the field does not stop DMPs from reaching their ultimate goals. The system therefore uses a Nadaraya-Watson model (Bishop, 2006) of the form

$$u = -s(x) \frac{\sum_{i=1}^N r_i c_i}{\sum_{j=1}^N r_j},$$

to generate a suitable detractor field, where r_i is a weight assigned to the i^{th} ECVD, s is the strength of the overall field, x is the state of the DMPs' canonical system, and c_i is the detracting force for a single descriptor.

The weight of an ECVD for collision avoidance is given by $r_i = \exp(-(\mathrm{v}_i - \mathrm{p})^{\mathrm{T}} \mathrm{h}(\mathrm{v}_i - \mathrm{p}))$, where v_i is the position of the i^{th} ECVD in the local coordinate system, h is a vector of positive length scale hyperparameters, and p is the finger tip position, as shown in Figure 2. The detractor therefore puts more importance on ECVDs in the vicinity of the finger.

The strength factor ensures that the detractor forces always tend to zero at the end of a movement and thus it can not obstruct the attractor from achieving its goal at the end. Therefore, the strength of the detractors is coupled to the canonical system of the DMP; i.e., $s(x) = (\sum_{j=1}^M \psi_j)^{-1} \sum_{i=1}^M \psi_i w_i x$, where x is the value of the canonical system, ψ are its basis functions, and w specify the varying strength of the field during the trajectory.

The transportation and finger-posture movements react differently to edges and thus employ different types of basis functions c_i for their respective potential fields. For the fingers, the individual potential fields are logistic sigmoid functions about the edge of each ECVD of the form $\rho(1 + \exp(d_i \sigma_c^{-2}))^{-1}$, where $d_i = \left\| (\mathrm{p} - \mathrm{v}_i) - \mathrm{e}_i (\mathrm{p} - \mathrm{v}_i)^{\mathrm{T}} \mathrm{e}_i \right\|$ is the distance from the finger to the edge, $\rho \geq 0$ is a scaling parameter, and $\sigma_c \geq 0$ is a length parameter. Differentiating the potential field results in a force term of

$$c_i = \rho \frac{\exp\left(d_i \sigma_c^{-2}\right)}{\left(1 + \exp\left(d_i \sigma_c^{-2}\right)\right)^2}.$$

As the logistic sigmoid is monotonically increasing, the detractor always forces the fingers open further to move their tips around the ECVDs and thus ensure that they always approach the object from the outside.

The hand uses instead the Gaussian basis functions of the form $\rho \exp(-0.5 \mathrm{d}_i^{\mathrm{T}} \mathrm{d}_i \sigma_d^{-2})$, where $\mathrm{d}_i = (\mathrm{q} - \mathrm{v}_i) - \mathrm{e}_i (\mathrm{q} - \mathrm{v}_i)^{\mathrm{T}} \mathrm{e}_i$ is the distance from the end effector position, q, to the edge, and $\rho \geq 0$ and $\sigma_d \geq 0$ are scale and length parameters respectively. Differentiating the potential with respect to d_i gives a force term in the Y direction of

$$c_i = \left[\rho \mathrm{d}_i \sigma_d^{-2} \exp(-0.5 \mathrm{d}_i^{\mathrm{T}} \mathrm{d}_i \sigma_d^{-2})\right]_Y,$$

which can be interpreted as a radial force from the edge with an exponentially decaying magnitude.

The detractor fields, of both the grasping and reaching components, have now been defined, and can be superimposed into the DMP framework as

$$\ddot{y} = \left(\alpha_z(\beta_z\tau^{-2}(g-y) - \tau^{-1}\dot{y}) + a\tau^{-2}f(x)\right) - \tau^{-2}u,$$

which then represents the entire ECVD and MP based potential field.

2.3 High Level DMP Controller for Grasping

Having defined the potential field for a single grasping motion, we interpolate the movements to new target grasps. Having a motion representation that can be interpolated to new targets is crucial for imitation learning. Given such a representation, the number of example trajectories required from the demonstrator can be greatly increased, making learning easier. While DMPs can interpolate to arbitrary goal positions, they have two drawbacks for grasping tasks; i.e., 1) the approach direction to the grasp can not be arbitrarily defined, and 2) the amplitude of the trajectory is unneccessarily sensitive to changes in the start position y_0 and the goal positon g if $y_0 \approx g$ during training, which can cause the robot to reach the limits of its workspace.

These difficulties can be overcome by including a supervisory controller that modifies the hyperparameters of the DMPs appropriately. The supervisor can maintain the correct approach direction by using a task-specific coordinate system. Due to the translation invariance of DMPs, only a rotation, $\mathrm{R} \in \mathbb{SO}(3)$, between the two coordinate systems needs to be determined. The majority of the motions will lie in a plane defined by the start and goal locations, and the final approach direction.

The first new in-plane axis x_p is set to be along the approach direction of the grasp; i.e., $\mathrm{x}_p = -\mathrm{a}$ as shown in Figure 3. As a result, the approach direction is easily defined and only requires that the Y_p and Z_p primitives reach their goal before the X_p primitive. The second axis, y_p, must be orthogonal to x_p and also in the plane, as shown in Figure 3. It is set to $\mathrm{y}_p = b^{-1}((\mathrm{g}-\mathrm{s}) - \mathrm{x}_p(\mathrm{g}-\mathrm{s})^T\mathrm{x}_p)$, where b^{-1} is a normalization term, and s and g are the motion's 3D start and goal positions respectively. The third vector, z_p, is orthogonal to the plane, and is derived by completing the right-handed coordinate system, i.e., $\mathrm{z}_p = \mathrm{x}_p \times \mathrm{y}_p$. The DMPs can now be specified by the supervisor in the X_p-Y_p-Z_p coordinate system, and mapped to the X_w-Y_w-Z_w world reference frame by multiplying by $\mathrm{R}^T = [\mathrm{x}_p, \mathrm{y}_p, \mathrm{z}_p]^T$.

The second problem relates to the scaling of motions with ranges greater than $y_0 - g$, which both com-

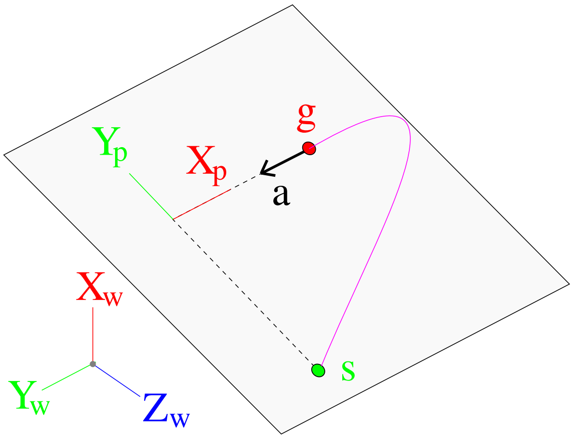


Figure 3: The above diagram shows the the change in coordinate systems for the transportation DMPs. The axes X_w-Y_w-Z_w are the world coordinate system, while X_p-Y_p-Z_p is the planar right handed coordinate system in which the DMP is specified. The trajectory of the DMP is shown by the **pink** line, starting at the **green point**, and ending at the **red point**. Note that X_p is parallel to the approach direction of the hand, which is shown by the **black arrow** a. The planar axis Y_p is perpendicular to X_p, and pointing from the motor primitive's starting location s towards the goal g.

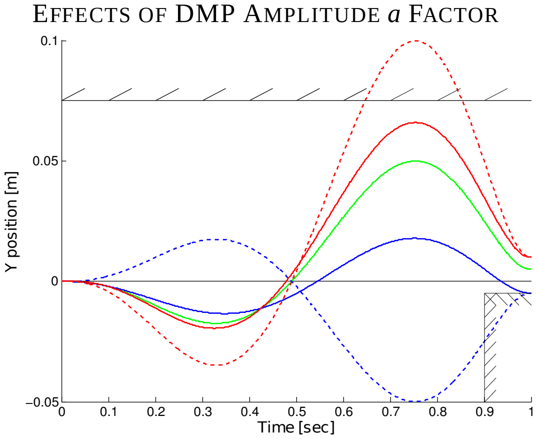


Figure 4: This is a demonstration of the effects of augmenting the amplitude variable a of DMPs. The black lines represent boundaries. The **green** plot shows the trained trajectory of the DMP going to 0.05, and is the same for all amplitude values. Now consider the scenario wherein the goal is placed at 0.1, but the workspace is limited to ±0.75 (top boundary). The **dashed red** line is the standard generalization to a larger goal, while the **red** plot uses the new amplitude. Notice how the new amplitude restricts the range of the trajectory to the workspace. In a different scenario, we move the goal to -0.05, but require the goal to be reached from above (lower right boundary), e.g., a finger placed on a surface. The **dashed blue** line is the standard generalization to a negative goal, and the **blue** trajectory uses the new amplitude. Note that the trajectory is not flip in the case of the new amplitude and thus stays within the restricted region. Both of the new trajectories were generated with $\eta = 0.25$, and maintain shapes close to that of the training trajectory.

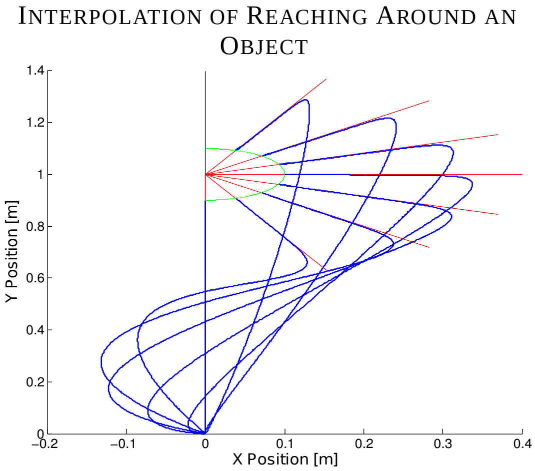


Figure 5: The plot shows workspace trajectories, wherein the x and y values are governed by two DMPs sharing a canonical system. The **red** lines indicate the desired approach direction while the **green** semicircle indicates the goal positions along them. The **blue** lines show the trajectories for the different goals. They make use of the higher level controller of Subsection 2.3, with $\eta = 0.25$. The approach direction DMP was trained on an amplitude of one.

ponents require to move around the outside of objects. In the standard form $a = g - y_0$ (Ijspeert et al., 2003), which can lead to motions that easily exceed the robot's workspace if $g \approx y_0$ during the training, but not during the motion reproduction. The supervisor can control these trajectories by scaling the shaping force (see Appendix), and thus we propose the amplitude term

$$a = \|\eta(g - y_0) + (1 - \eta)(g_T - y_{0T})\|,$$

where g_T and y_{0T} are the goal and start positions of the training data respectively, and $\eta \in [0,1]$ is a weighting hyperparameter. The resulting trajectory amplitude is in the convex hull of the training amplitude and the standard interpolation value ($a = g - y_0$) (Ijspeert et al., 2003) and thus only affects how conservative the generalization to new points is, as can be seen in Figure 4. By taking the absolute value of the amplitude, the approach direction is not reversed, giving a result similar to the use of a constant amplitude proposed by Park et al. (Park et al., 2008), which corresponds to the special case of $\eta = 0$.

Example interpolations of a transportation trajectory can be seen in Figure 5.

3 GRASPING EXPERIMENTS

The methods described in Section 2 were implemented and evaluated on a real robot platform. The robot consists of a Videre stereo camera mounted on a pan-tilt unit, a Barrett hand, and a Mitsubishi PA10 arm. The robot was given the task of grasping an object amongst clutter using only an ECVD model of the object. The results of these trials were then compared to trials of the same grasps using other standard robotics methods for comparison. We hypothesize that our method will result in significantly more successful grasps than the other methods.

3.1 Grasping Experiment Procedure

Before the robot can perform a grasping task, its motions must be initialized. Determining the finger goal state and specifying the detractor fields introduces several new hyperparameters that have simple geometrical interpretations. For instance, $\mathrm{h} = 2[w\ l\ l]^{\mathrm{T}}$, where w and l are the width and length of the finger respectively. To reflect the human tendency towards more precise movements during the last 30% of a motion (Jeannerod, 2009), the strength function, $s(x)$, was set to give the highest strengths during the first 70% of the motion for the transportation, and the last 30% for the finger posture.

A VICON$^{\mathrm{TM}}$ motion tracking system was used to record the movements of a human test subject during a grasping task, which used a different object to the one used by the robot. As the reaching trajectories are encoded in task space rather than joint space, the correspondence problem was not an issue for the imitation learning. Similarly, the DMPs of the fingers are homogeneous, which circumvents the correspondence problem of mapping the five human fingers onto the three fingers of the robot. The imitation learning was performed using locally weighted regression in the the X_p-Y_p-Z_p coordinate system, as proposed by Ijspeert et al. (Ijspeert et al., 2002).

Having defined the basic motions, the robot was then given the task of grasping an object without hitting surrounding obstacles (see Figure 1). Each trial begins with an estimate of the pose of the object relative to the robot (Detry et al., 2008) and sets its grasp location accordingly. The model's ECVD are then projected into the scene, and the robot attempts to perform the grasp and lift the object 15cm so that it is clear of the stand. The trial is a success if the robot can detect the object in its hand at this point. If the hand collides with an obstacle or knocks the object down, the trial is marked as a failure. Grasps were varied to include different approach directions and locations around the object. The experiment consisted of 45 trials.

Two alternative approaches were compared with our proposed method. The first represents a standard robotics approach of specifying a trajectory by

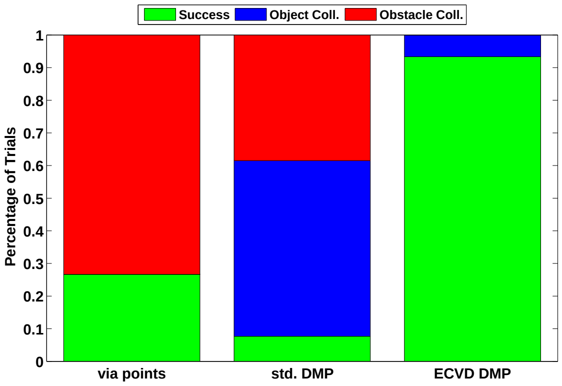


Figure 6: The occurrences of successes and collision types for the different methods are shown. The first column presents the results for the traditional robotics method of specifying trajectories by via points. The second column corresponds to using standard DMPs, while the final column incorporates the ECVD based potential field and supervisory DMP controller. The occurrences are given as the percentage of trials. Trials that collided multiple times, are classified by their first collision.

straight lines between via points and uses fully extended fingers with no preshaping of the hand. The other approach is to use standard DMPs learned from the same human demonstrated movements as our proposed methods, but without the proposed detractor field and supervisory controller. The same grasp locations were proposed to the different methods, and obstacles were placed in similar positions for the the different trials to allow for a fair comparison between the methods.

3.2 Experimental Results

From the three tested methods, the proposed method acquired the highest success rate, as can be seen in Figure 6. The task was not trivial, and all of the methods encountered both successes and problems during the trials.

The standard DMP method encountered the most problems (success rate of only 7%) a majority of which were caused by collisions with the object. This high failure rate can be attributed to the method not specifically incorporating a desired approach direction. In successful trials, the approach direction was close to that of the initial imitation learning. Therefore the proposed DMP supervisor improved the generalization of the movement to new target grasps, and the system would benefit from it even in uncluttered environments. Similarly, the open-loop preshaping of the hand helped avoid obstacles, but occasionally prevented the hand from being sufficiently open to accept the object. The proposed detractor field successfully overcame this problem for the ECVD DMPs.

The via points method encountered no collisions with the object, and would have worked well in an uncluttered environment. The method still encountered collisions with the obstacles for 73% of the trials, but this is more reflective of the difficulty of the task rather than the via point method. The method can therefore be considered as a good approach if it were combined with a suitable path planning method for obstacle avoidance. However, the path planner would need additional information and assumptions about the scene and possibly even extra hardware to acquire it.

The proposed method had a success rate of 93%, with no occurrences of collisions with obstacles. The trials that did fail were the result of the object falling down while the fingers were closing and thus do not indicate problems with the approach used to reaching the grasp location. The method does have certain restrictions though. The magnitude of the detractor fields needs to be calibrated based on the density of ECVDs for common objects, but some obstacles encountered may present lower densities. As the current set of ECVD relies on object edges, smooth objects can lead to noisy or very sparse descriptors, and therefore not create a suitable basis for obstacle avoidance. As the number of descriptor types increases (e.g., corner and plane descriptors), this will become less of a problem. Occluded obstacles will also need to rely on additional information (e.g., force feedback) to be avoided, although this is a source of error for all vision based planners.

Given a few restrictions, the results still show that our hypothesis was correct and the proposed methods represent a suitable basis for avoiding obstacles without relying on a complicated path planner and using only a small amount of vision information compared to standard robot systems.

4 CONCLUSIONS

The proposed methods augment dynamical system motor primitives to incorporate Early Cognitive Vision descriptors by using a potential field. These methods represent important tools that a robot needs to reactively execute grasps of an object in a cluttered environment without relying on a complex planner. The techniques allow for preshaping the fingers to match the shape and size of the object and curving the trajectory of the hand around objects(Wank et al., 2004). These modifications were tested on a real robot, and it was discovered that the methods were not only successful at performing the task, but also allowed for easier imitation learning, better interpolation of the learned trajectories, and significantly better chances of a success of a grasp in cluttered environments than standard motor primitives. Although the experiments were performed within a grasping task scenario, the proposed methods can be beneficial for other manipulation tasks, such as pressing buttons and pushing objects.

REFERENCES

Arimoto, S. (2008). *Control Theory of Multi-fingered Hands*. Springer London.

Bard, C., Troccaz, J., and Vercelli, G. (1991). Shape analysis and hand preshaping for grasping. In *Proceedings of IROS'91*.

Bicchi, A. and Kumar, V. (2000). Robotic grasping and contact: a review. In *ICRA 2000 proceedings*.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

Chieffi, S. and Gentilucci, M. (1993). Coordination between the transport and the grasp components during prehension movements.

Detry, R., Kroemer, O., Popovic, M., Touati, Y., Baseski, E., Krueger, N., Peters, J., and Piater, J. (2009). Object-specific grasp affordance densities. In *ICDL*.

Detry, R., Pugeault, N., and Piater, J. (2008). Probabilistic pose recovery using learned hierarchical object models. In *International Cognitive Vision Workshop*.

Graziano, M. S. (2006). Progress in understanding spatial coordinate systems in the primate brain. *Neuron*.

Hartley, R. and Zisserman, A. (2000). *Multiple View Geometry in Computer Vision*. Cambridge University Press.

Hsiao, K., Nangeroni, P., Huber, M., Saxena, A., and Ng, A. (2009). Reactive grasping using optical proximity sensors. In *ICRA 2009 Proceedings*.

Iberall, T. (1987). Grasp planning for human prehension. In *Proceedings of ICAI'87*.

Ijspeert, A. J., Nakanishi, J., , and Schaal, S. (2002). Movement imitation with nonlinear dynamical systems in humanoid robots. In *ICRA*.

Ijspeert, A. J., Nakanishi, J., , and Schaal, S. (2003). Learning attractor landscapes for learning motor primitives. In *NIPS*.

Jeannerod, M. (1997). *Perspectives of Motor Behaviour and Its Neural Basis*, chapter Grasping Objects: The Hand as a Pattern Recognition Device.

Jeannerod, M. (2009). *Sensorimotor Control of Grasping: Physiology and Pathophysiology*, chapter The study of hand movements during grasping. A historical perspective. Cambridge University Press.

Krueger, N., Lappe, M., and Woergoetter, F. (2004). Biologically motivated multimodal processing of visual primitives. *The Interdisciplinary Journal of Artificial Intelligence and the Simulation of Behaviour*.

Oztop, E., Bradley, N. S., and Arbib, M. A. (2004). Infant grasp learning: a computational model.

Oztop, E. and Kawato, M. (2009). *Sensorimotor Control of Grasping: Physiology and Pathophysiology*, chapter Models for the control of grasping. Cambridge University Press.

Park, D.-H., Hoffmann, H., Pastor, P., and Schaal, S. (2008). Movement reproduction and obstacle avoidance with dynamic movement primitives and potential fields. In *IEEE International Conference on Humanoid Robots(HUMANOIDS)*.

Pugeault, N. (2008). *Early Cognitive Vision: Feedback Mechanisms for the Disambiguation of Early Visual Representation*. Vdm Verlag Dr. Mueller.

Saxena, A., Dreimeyer, J., Kearns, J., Osondu, C., and Ng, A. (2008). *Experimental Robotics*, chapter Learning to Grasp Novel Objects using Vision. Springer Berlin.

Schaal, S., Peters, J., Nakanishi, J., and Ijspeert, A. (2003). Learning movement primitives. In *Proceedings of ISRR'03*.

Steffen, J., Haschke, R., and Ritter, H. (2007). Experience-based and tactile-driven dynamic grasp control. In *IRS proceedings*.

Wank, V., Fischer, A., Bos, K., Boesnach, I., Moldenhauer, J., and Beth, T. (2004). Similarities and varieties in human motiontrajectories of predefined grasping and disposing movements. *International Journal of Humanoid Robotics*.

APPENDIX

Dynamical Systems Motor Primitives

The dynamical systems motor primitives (DMPs) proposed by Ijspeert et al. (Ijspeert et al., 2003) were inspired by the simple, but highly adaptive, motions that animals employ, and combine to obtain more complex motions. The primitives are implemented as a passive dynamical system with an external force, and represented as

$$\ddot{y} = \alpha_z(\beta_z\tau^{-2}(g-y) - \tau^{-1}\dot{y}) + a\tau^{-2}f(x), \qquad (1)$$

where α_z and β_z are constants, τ controls the duration of the primitive, a is an amplitude, $f(x)$ is a nonlinear function, and g is the goal for the state variable y.

By selecting α_z and β_z appropriately, and setting $a = 0$, the system reduces to $\ddot{y} = \alpha_z(\beta_z\tau^2(g-y) - \tau\dot{y})$ and becomes a critically damped global attractor. It can be visualized as a spring and damper system that ensures state y will always end at the goal value g.

The function $f(x)$ is a shaping function based on the state, $x \in [0,1]$, of the canonical system that synchronizes the DMPs $\dot{x} = -\alpha_x\tau x$, where α_x is a time constant. The function takes the form

$$f(x) = \frac{\sum_{j=1}^{M} \psi_j(x) w_j x}{\sum_{i=1}^{M} \psi_i(x)},$$

where M is the number of basis functions, $\psi(x)$ are Gaussian basis functions, and w are weights acquired through locally weighted regression (Ijspeert et al., 2003). This function has the effect of introducing a non-linearity that can affect the spring-damper system to output any arbitrary trajectory specified by the user. Due to the dependence of $f(x)$ on x, the shaping term decays to zero with x, so that the spring and damper beneficial properties of the attractor are maintained.

The resulting primitives can encode arbitrary trajectories, and still ensure that the goal state is always achieved. The trajectories can also be scaled in time and space by setting the τ and g variables appropriately and thus generalize to a range of situations.

Early Cognitive Vision System

The entire prehensile process effectively occurs before the hand has even touched the object and thus the vision system plays a very important role (Bard et al., 1991; Iberall, 1987). Our system uses the Early Cognitive Vision methods of Pugeault et al. (Pugeault, 2008; Hartley and Zisserman, 2000), which makes a minimal number of assumptions about the object, and has been successfully implemented to determine good grasp locations (Detry et al., 2009). A principal idea of this vision system is to store additional low level information and perform perceptual grouping on it to later aid the higher level stereo matching and 3D constructions.

The methods extract local features of a scene, which it localizes and orientates in space (Krueger et al., 2004). Each descriptor is a symbolic representation for an edge in 3D. The resulting features are called early cognitive vision descriptors (ECVD) (Pugeault, 2008), and can be used in generating models of objects for pose estimation (Detry et al., 2008), and for symbolically describing 3D scenes. By using a large amount of small ECVDs, any arbitrary object can be represented.

When performing a grasping task, the robot uses a hierarchical Markov model of the object's ECVD geometry (Detry et al., 2008) to determine its pose, which can then be used to superimpose the ECVDs of the model back into the scene. The grasping techniques can therefore use geometric information of a partially occluded object.

REMOTE CONTROL OF A MOBILE ROBOT SUBJECT TO A COMMUNICATION DELAY

A. Alvarez-Aguirre, H. Nijmeijer
Department of Mechanical Engineering, Eindhoven University of Technology
P.O Box 513, 5600 MB Eindhoven, The Netherlands
{*a.a.alvarez, h.nijmeijer*}*@tue.nl*

T. Oguchi, K. Kojima
Department of Mechanical Engineering, Tokyo Metropolitan University
1-1, Minami-osawa, Hachioji-shi, Tokyo 192-0397 Japan
t.oguchi@tmu.ac.jp, kojima-koutarou@ed.tmu.ac.jp

Keywords: Mobile robotics, Remote control, Time-delay systems, Controlled synchronization, Nonlinear predictor.

Abstract: This paper addresses the remote tracking control of a mobile robot subject to a bilateral time-delay. The delay affects the system since the controller and the robot are linked via a delay inducing communication channel, such as the Internet, and consequently, the performance and stability of the system are compromised. Based on the notion of anticipating synchronization, a state estimator which stabilizes the system when it is affected by a bilateral time-delay is proposed. A stability analysis including the system, tracking controller and estimator is provided, and the applicability of the proposed delay compensation strategy is demonstrated by means of experiments between multi-robot platforms located in Eindhoven, The Netherlands and Tokyo, Japan.

1 INTRODUCTION

In the increasingly fast and diverse technological developments of the last decades the duties and tasks conferred to control systems have become much more complex and decisive. Requirements now encompass flexibility, robustness, ubiquity, transparency and balancing tradeoffs, among others.

Specifically, the study of systems embedded with time-delays and the control methodologies that can be applied to them has become significatively important as a way to undertake remote, dangerous or distributed tasks. As a matter of fact, the remote control, or the control of a system subject to a bilateral time-delay, is part of the underlying problem in two of control engineering's fundamental topics, namely teleoperation strategies and Networked Control Systems (NCS). The problem remains a central issue even though there are many more considerations in addition to this problem when considering teleoperated systems and NCS. Examples of aspects to consider would be transparency and force reflection in teleoperation (Niemeyer et al., 2008); and varying transmission delays and sampling/transmission intervals, packet loss, communication constraints and quantization effects in NCS (Heemels et al., 2010).

Several techniques have been proposed so far in order to cope with bilateral time-delay in this setting, e.g. the use of the scattering transformation, wave variables formulation, and queuing methodologies to name a few. A detailed description of such techniques and many others, together with further references, can be found in (Hokayem and Spong, 2006) and (Tipsuwan and Chow, 2003).

In this work, a control strategy which allows the remote control of a unicycle-type mobile robot is proposed. The bilateral time-delay is compensated by means of a state estimator inspired on a predictor based on synchronization presented in (Oguchi and Nijmeijer, 2005a) and (Oguchi and Nijmeijer, 2005b). The main idea behind the state estimator is to reproduce the system's behavior without time-delay in order to drive an anticipating controller. The problem presents various challenges since the system is nonlinear and subject to a non-holonomic constraint. Additionally, the difficulties faced when implementing the ideas proposed in an experimental setting using the Internet as the communication channel should be taken into account and are also discussed in depth. In (Kojima et al., 2010) a similar state estimator has been applied to a mobile robot subject to a communication delay, and the necessary conditions for the estimator's convergence have been derived. In this paper an alternative approach is taken in order to prove the stability of the whole system, including the mobile robot, the tracking controller and the state estimator.

The paper is organized as follows. Section 2 recalls the tracking control of a delay-free mobile robot. In Section 3, a control scheme intended to control a mobile robot subject to a bilateral time-delay is proposed together with its corresponding stability analysis. Section 4 provides an overview of the experimental platform used to validate the control strategies proposed, explains how the most critical implementation issues were addressed, and presents the experimental results. Conclusions and ideas for future work are provided in Section 5.

2 CONTROL OF A UNICYCLE

The tracking control of a unicycle-type mobile robot is presented in this section. To begin with, consider the posture kinematic model of a unicycle,

$$\begin{aligned}\dot{x}(t) &= v(t)\cos\theta(t),\\ \dot{y}(t) &= v(t)\sin\theta(t),\\ \dot{\theta}(t) &= \omega(t),\end{aligned} \tag{1}$$

in which $x(t)$ and $y(t)$ denote the robot's position in the global coordinate frame X-Y, $\theta(t)$ defines its orientation w.r.t to the X axis, and $v(t)$ and $\omega(t)$ describe the robots' translational and rotational velocities respectively, regarded as its control inputs. The system's state is defined as $q(t) = [x(t)\ y(t)\ \theta(t)]^T$ and the non-slip condition on the unicycle's wheels impose a non-holonomic constraint (Brockett, 1983).

The control objective for the robot is to track the reference trajectory generated by the exosystem,

$$\begin{aligned}\dot{x}_r(t) &= v_r(t)\cos\theta_r(t),\\ \dot{y}_r(t) &= v_r(t)\sin\theta_r(t),\\ \dot{\theta}_r(t) &= \omega_r(t),\end{aligned} \tag{2}$$

with state $q_r(t) = [x_r(t)\ y_r(t)\ \theta_r(t)]^T$. The exosystem's reference velocities $v_r(t)$ and $\omega_r(t)$ are defined in terms of its Cartesian velocities $\dot{x}_r(t)$, $\dot{y}_r(t)$ and accelerations $\ddot{x}_r(t)$, $\ddot{y}_r(t)$, i.e.,

$$\begin{aligned}v_r(t) &= \sqrt{\dot{x}_r^2(t)+\dot{y}_r^2(t)},\\ \omega_r(t) &= \frac{\dot{x}_r(t)\ddot{y}_r(t)-\ddot{x}_r(t)\dot{y}_r(t)}{\dot{x}_r^2(t)+\dot{y}_r^2(t)}.\end{aligned} \tag{3}$$

The difference between the exosystem's and the system's states may be expressed w.r.t the system's local coordinate frame X'–Y' in order to define the error coordinates $q_e(t) = [x_e(t)\ y_e(t)\ \theta_e(t)]^T$, as proposed by (Kanayama et al., 1990) and shown in Figure (1). These coordinates are given by the clockwise rotation of the position differences between $q_r(t)$ and $q(t)$, resulting in,

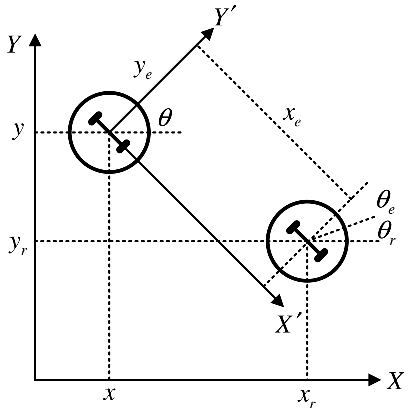


Figure 1: Mobile robot, reference exosystem, and error coordinates.

$$\begin{bmatrix}x_e(t)\\ y_e(t)\\ \theta_e(t)\end{bmatrix} = \begin{bmatrix}\cos\theta(t) & \sin\theta(t) & 0\\ -\sin\theta(t) & \cos\theta(t) & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}x_r(t)-x(t)\\ y_r(t)-y(t)\\ \theta_r(t)-\theta(t)\end{bmatrix}. \tag{4}$$

Differentiating (4) w.r.t. time yields the following error dynamics,

$$\begin{aligned}\dot{x}_e(t) &= \omega(t)y_e(t)+v_r(t)\cos\theta_e(t)-v(t),\\ \dot{y}_e(t) &= -\omega(t)x_e(t)+v_r(t)\sin\theta_e(t),\\ \dot{\theta}_e(t) &= \omega_r(t)-\omega(t).\end{aligned} \tag{5}$$

A tracking controller which results in closed-loop error dynamics which have a cascaded structure has been proposed in (Jakubiak et al., 2002), (Panteley et al., 1998), and is given by,

$$\begin{aligned}v(t) &= v_r(t)+c_2x_e(t)-c_3\omega_r(t)y_e(t),\\ \omega(t) &= \omega_r(t)+c_1\sin\theta_e(t),\end{aligned} \tag{6}$$

with c_1, $c_2 > 0$ and $c_3 > -1$ ensuring stability.

3 BILATERAL TIME-DELAY COMPENSATION

In this section, a state estimator with a predictor-like structure similar to the one proposed in (Kojima et al., 2010) is applied to a unicycle-type mobile robot subject to a bilateral time-delay. The origin of this type of predictor can be traced back to the appearance of the notion of anticipating synchronization in coupled

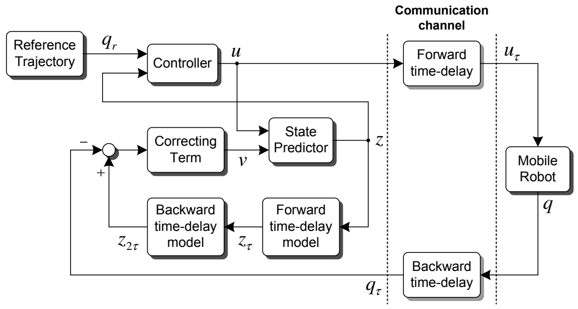


Figure 2: Bilateral time-delay compensation scheme block diagram representation.

chaotic systems, which was first observed by (Voss, 2000). After the same behavior was observed in certain physical systems such as specific electronic circuits and lasers, it was studied for more general systems in (Oguchi and Nijmeijer, 2006). As a result of this generalization, a state predictor based on synchronization for nonlinear systems with input time-delay was proposed in (Oguchi and Nijmeijer, 2005a). The same concept, which can be seen as a state estimator with a predictor-like structure, is proposed for a mobile robot subject to a bilateral time-delay in the block diagram in Figure 2.

3.1 Controller Structure

When considering a bilateral time-delay the system's output is also delayed. In this case the mobile robot is subject to a forward τ_f and backward τ_b time-delay, as denoted in (Hokayem and Spong, 2006). Given the mobile robot (1) subject to a bilateral time-delay, the robot's posture kinematic model after the forward delay is given by,

$$\begin{aligned}\dot{x}(t) &= v(t-\tau_f)\cos\theta(t),\\ \dot{y}(t) &= v(t-\tau_f)\sin\theta(t),\\ \dot{\theta}(t) &= \omega(t-\tau_f).\end{aligned} \tag{7}$$

From the controller's side point of view, the robot's kinematic model is also affected by the backward time-delay τ_b, resulting in the following model,

$$\begin{aligned}\dot{x}(t-\tau_b) &= v(t-\tau_f-\tau_b)\cos\theta(t-\tau_b),\\ \dot{y}(t-\tau_b) &= v(t-\tau_f-\tau_b)\sin\theta(t-\tau_b),\\ \dot{\theta}(t-\tau_b) &= \omega(t-\tau_f-\tau_b).\end{aligned} \tag{8}$$

To improve the robot's performance when subject to the time-delay, the dynamics of the estimator, with state $z(t) = [z_1(t)\ z_2(t)\ z_3(t)]^T$, are proposed as,

$$\begin{aligned}\dot{z}_1(t) &= v(t)\cos z_3(t) + \nu_x(t),\\ \dot{z}_2(t) &= v(t)\sin z_3(t) + \nu_y(t),\\ \dot{z}_3(t) &= \omega(t) + \nu_\theta(t),\end{aligned} \tag{9}$$

with $\nu(t) = [\nu_x(t)\ \nu_y(t)\ \nu_\theta(t)]^T$ defining a correcting term relating the estimator's and the system's states.

The correcting term $\nu(t)$ is intended to bring the estimator's and the system's states closer, since the robot's initial conditions are assumed to be unknown. There is complete freedom in the design of this term, with the simplest choice being,

$$\begin{aligned}\nu_x(t) &= -K_x(z_1(t-\tilde{\tau}_f-\tilde{\tau}_b) - x(t-\tilde{\tau}_b)),\\ \nu_y(t) &= -K_y(z_2(t-\tilde{\tau}_f-\tilde{\tau}_b) - y(t-\tilde{\tau}_b)),\\ \nu_\theta(t) &= -K_x(z_3(t-\tilde{\tau}_f-\tilde{\tau}_b) - \theta(t-\tilde{\tau}_b)),\end{aligned} \tag{10}$$

where $\tilde{\tau}_f$ and $\tilde{\tau}_b$ model the robot's forward and backward delays. Hereinafter the forward and backward time-delays are assumed to be equal and constant, i.e. $\tau_b = \tau_f = \tau$, and modeled perfectly, i.e. $\tilde{\tau}_f = \tilde{\tau}_b = \tau$. The feasibility of this assumption will be discussed before the experimental results are presented.

In order to ease the computations in the stability analysis, a different correcting term will be proposed. Two new sets of error coordinates are defined, namely $z_e(t)$ and $p_e(t)$. The first coordinates relate the estimator's state with the reference trajectory, i.e.

$$\begin{bmatrix} z_{1_e}(t)\\ z_{2_e}(t)\\ z_{3_e}(t)\end{bmatrix} = \begin{bmatrix}\cos z_3(t) & \sin z_3(t) & 0\\ -\sin z_3(t) & \cos z_3(t) & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix} x_r(t)-z_1(t)\\ y_r(t)-z_2(t)\\ \theta_r(t)-z_3(t)\end{bmatrix}. \tag{11}$$

while the second set relates the delayed estimator's state with the current system's state, i.e.,

$$\begin{bmatrix} p_{1_e}(t)\\ p_{2_e}(t)\\ p_{3_e}(t)\end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & 0\\ p_{21} & p_{22} & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix} x(t-\tau)-z_1(t-2\tau)\\ y(t-\tau)-z_2(t-2\tau)\\ \theta(t-\tau)-z_3(t-2\tau)\end{bmatrix}, \tag{12}$$

where $p_{11} = p_{22} = \cos z_3(t-2\tau)$ and $p_{12} = -p_{21} = \sin z_3(t-2\tau)$.

Given the error coordinates (11) and (12), the following correcting term is proposed,

$$\begin{aligned}\nu_x(t) &= -K_x p_{1_e}(t)\cos z_3(t) + K_y p_{2_e}(t)\sin z_3(t),\\ \nu_y(t) &= -K_x p_{1_e}(t)\sin z_3(t) - K_y p_{2_e}(t)\cos z_3(t),\\ \nu_\theta(t) &= -K_\theta \sin p_{3_e}(t).\end{aligned} \tag{13}$$

Since the control scheme of Figure 2 uses the state estimator's output as the controller's input, a new control law is proposed based on the tracking control (6),

$$\begin{aligned} v(t) &= v_r(t) + c_2 z_{1_e}(t) - c_3\omega_r(t) z_{2_e}(t),\\ \omega(t) &= \omega_r(t) + c_1\sin z_{3_e}(t).\end{aligned} \tag{14}$$

Remark 1. *Due to the input time-delay, the system's control action is given by the delayed controller, i.e.,*

$$v(t-\tau) = v_r(t-\tau) + c_2 z_{1_e}(t-\tau) - c_3\omega_r(t-\tau) z_{2_e}(t-\tau),$$
$$\omega(t-\tau) = \omega_r(t-\tau) + c_1 \sin z_{3_e}(t-\tau).$$

The resulting control action provides an idea of how the system will behave. Intuitively, the robot will track the reference trajectory after a time τ*. This will be examined in detail during the stability analysis.*

Remark 2. *Although the stability analysis is unique for each control law, the delay compensation scheme is controller independent, so in this sense the generality of the scheme holds. Consider for example a controller which accounts for actuator saturation and collision avoidance such as in (Kostic et al., 2009).*

3.2 Stability Analysis

In order to describe the system's performance it becomes necessary to establish stability criteria. The system's control objectives are defined as follows,

- $q(t) \to q_r(t-\tau)$, the system converges to a delayed version of the reference trajectory;
- $z(t) \to q(t+\tau)$, the state estimator anticipates the system;
- $z(t) \to q_r(t)$, the state estimator converges to the reference trajectory.

Considering the control objectives and Remark 1, the following proposition is formulated.

Proposition 1. *Given the unicycle-type mobile robot (7) subject to a bilateral delay* 2τ*, the state estimator (9)-(13), and the control law (14), the robot will track a delayed version* $q_r(t-\tau)$ *of the reference trajectory.*

In accordance to Proposition 1 it follows that proving that the equilibrium point $(z_e(t), p_e(t)) = 0$ is stable satisfies the control objectives.

The error coordinates (11)-(12) are grouped and differentiated w.r.t time, and the tracking control law (14) is substituted in them. Given $\xi_1 = [z_{1_e}\ z_{2_e}\ p_{1_e}\ p_{2_e}]^T$ and $\xi_2 = [z_{3_e}\ p_{3_e}]^T$, the resulting closed-loop error dynamics are rearranged in the following cascaded form,

$$\begin{aligned}\dot{\xi}_1(t) = {} & A_0(t,t-2\tau)\xi_1(t) + A_1\xi_1(t-2\tau) \\ & + g(t,t-2\tau,\xi_1(t),\xi_2(t),\xi_1(t-2\tau),\xi_2(t-2\tau)),\end{aligned} \tag{15}$$

$$\dot{\xi}_2(t) = f_2(t,\xi_2(t),\xi_2(t-2\tau)), \tag{16}$$

where,

$$A_0(t,t-2\tau) = \begin{bmatrix} -c_2 & f_{12} & K_x & 0 \\ f_{21} & 0 & 0 & K_y \\ 0 & 0 & 0 & f_{34} \\ 0 & 0 & f_{43} & 0 \end{bmatrix},$$

$$A_1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & K_x & 0 \\ 0 & 0 & 0 & K_y \end{bmatrix},$$

$$g(t,t-2\tau,\xi_1(t),\xi_2(t),\xi_1(t-2\tau),\xi_2(t-2\tau)) =$$
$$\begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \\ 0 & g_{32} \\ 0 & g_{42} \end{bmatrix} \xi_2(t) + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ h_{31} & h_{32} \\ h_{41} & h_{42} \end{bmatrix} \xi_2(t-2\tau),$$

$$f_2(t,\xi_2(t),\xi_2(t-2\tau)) = \begin{bmatrix} -c_1 \sin z_{3_e}(t) + K_\theta \sin p_{3_e}(t) \\ K_\theta \sin p_{3_e}(t-2\tau) \end{bmatrix},$$

with,

$$f_{12} = (1+c_3)\omega_r(t),$$
$$f_{21} = -\omega_r(t),$$
$$f_{34} = -f_{43} = \omega_r(t-2\tau),$$
$$g_{11} = c_1 z_{2_e}(t)\int_0^1 \cos(s z_{3_e}(t))ds + v_r(t)\int_0^1 \sin(s z_{3_e}(t))ds,$$
$$g_{21} = (v_r(t) - c_1 z_{1_e}(t))\int_0^1 \cos(s z_{3_e}(t))ds,$$
$$g_{12} = -K_\theta z_{2_e}(t)\int_0^1 \cos(s p_{3_e}(t))ds,$$
$$g_{22} = K_\theta z_{1_e}(t)\int_0^1 \cos(s p_{3_e}(t))ds,$$
$$\begin{aligned} g_{32} = {} & (v_r(t-2\tau) + c_2 z_{1_e}(t-2\tau) - c_3\omega_r(t-2\tau) z_{2_e}(t-2\tau)) \\ & \cdot \int_0^1 \sin(s p_{3_e}(t))ds,\end{aligned}$$
$$\begin{aligned} g_{42} = {} & (v_r(t-2\tau) + c_2 z_{1_e}(t-2\tau) - c_3\omega_r(t-2\tau) z_{2_e}(t-2\tau)) \\ & \cdot \int_0^1 \cos(s p_{3_e}(t))ds,\end{aligned}$$
$$h_{31} = c_1 p_{2_e}(t)\int_0^1 \cos(s z_{3_e}(t-2\tau))ds,$$
$$h_{41} = -c_1 p_{1_e}(t)\int_0^1 \cos(s z_{3_e}(t-2\tau))ds,$$
$$h_{32} = -K_\theta p_{2_e}(t)\int_0^1 \cos(s p_{3_e}(t-2\tau))ds,$$
$$h_{42} = K_\theta p_{1_e}(t)\int_0^1 \cos(s p_{3_e}(t-2\tau))ds.$$

The definition of a persistently exciting (PE) signal is required in order to establish the stability of the cascaded system (15)-(16).

Definition 1. *A continuous function* $\omega : \mathbb{R}^+ \to \mathbb{R}$ *is said to be persistently exciting (PE) if* $\omega(t)$ *is bounded, Lipschitz, and constants* $\delta_c > 0$ *and* $\varepsilon > 0$ *exist such that,*

$$\forall t \geq 0,\ \exists s :\ t - \delta_c \leq s \leq t \ \ \textit{such that}\ \ |\|\omega(s)| \geq \varepsilon.$$

It is assumed that the desired rotational velocity ω_r, $i = 1,2$ is PE. Consequently, the time varying term $\omega_r(t-2\tau)$, in the matrix function $A_0(t,t-2\tau)$ may be renamed as $\bar{\omega}_r(t)$, resulting in a matrix $\bar{A}_0(t)$. Note that the matrix function, although time-varying, will always have entries in the same positions. From a practical viewpoint the matrix's entry values will change with time, but not its structure.

The following theorem establishes the local stability of the equilibrium point $(z_e(t), p_e(t)) = 0$, and hence the fulfillment of the control objectives.

Theorem 1. *Consider a unicycle-type mobile robot subject to a bilateral time-delay 2τ with a posture kinematic model given by (7). Suppose that the robot's reference trajectory is generated by an exosystem with dynamics (2). The estimator (9) together with the correcting term (13) are used to generate the system's control input (14), which is applied to the robot after a delay τ. Suppose that the reference rotational velocity $\omega_r(t)$ is PE, that the controller gains satisfy $c_1 > 0$, $c_2 > 0$, and $c_3 > -1$, and that the correcting term gains satisfy $K_x, K_y, K_\theta < 0$. If*

- *subsystem $\dot{\xi}_1(t) = \bar{A}_0(t)\xi_1(t) + A_1\xi_1(t-2\tau)$ in (15) is at least locally exponentially stable (LES);*
- *function g in (15) is bounded;*
- *system (16) is LES;*

then the equilibrium point $(z_e(t), p_e(t)) = 0$ of the closed-loop error dynamics (15)-(16) is LES.

Proof. For the sake of brevity only a sketch of the proof will be provided.

The theorem is derived from the results for cascaded systems presented in (Panteley and Loría, 1998), which have been successfully applied to the tracking control of a mobile robot in (Jakubiak et al., 2002) and to the mutual synchronization of two robots in (van den Broek et al., 2009).

3.2.1 Stability of $\xi_1(t) = 0$

Subsystem $\dot{\xi}_1(t)$ in (15) is a linear time-varying (LTV) system which can be separated as follows,

$$\begin{bmatrix} \dot{z}_{1_e}(t) \\ \dot{z}_{2_e}(t) \end{bmatrix} = \begin{bmatrix} -c_2 & (1+c_3)\omega_r(t) \\ -\omega_r(t) & 0 \end{bmatrix} \begin{bmatrix} z_{1_e}(t) \\ z_{2_e}(t) \end{bmatrix} + \begin{bmatrix} K_x & 0 \\ 0 & K_y \end{bmatrix} \begin{bmatrix} p_{1_e}(t) \\ p_{2_e}(t) \end{bmatrix}, \tag{17}$$

$$\begin{bmatrix} \dot{p}_{1_e}(t) \\ \dot{p}_{2_e}(t) \end{bmatrix} = \begin{bmatrix} 0 & \bar{\omega}_r(t) \\ -\bar{\omega}_r(t) & 0 \end{bmatrix} \begin{bmatrix} p_{1_e}(t) \\ p_{2_e}(t) \end{bmatrix} + \begin{bmatrix} K_x & 0 \\ 0 & K_y \end{bmatrix} \begin{bmatrix} p_{1_e}(t-2\tau) \\ p_{2_e}(t-2\tau) \end{bmatrix}. \tag{18}$$

Note that the output of system (18) perturbs system (17) in a similar way as in the cascaded structure (15)-(16). Nevertheless, the coupling term in (17) will vanish if (18) converges to zero. The reason for this is that the coupling term does not depend on $z_{1_e}(t)$ or $z_{2_e}(t)$, which is not the case of the coupling term in (15)-(16), which includes ξ_1 and ξ_2. Consequently, the stability of $\dot{\xi}_1(t) = \bar{A}_0(t)\xi_1(t) + A_1\xi_1(t-2\tau)$ can be established by proving the stability of (17) and (18) separately.

The time-delay in (18) is approximated by a Taylor series expansion and the stability of the subsystem is established using the resulting characteristic polynomials, which require that $K_x, K_y < 0$.

Knowing that the coupling term vanishes, the stability of the first term in (17) has already been studied in (van den Broek, 2008) and (van den Broek et al., 2009), and can be established by a well-known result in the adaptive control field for LTV systems. The requirement is that that $\omega_r(t)$ be PE and that the controller gains satisfy $c_2 > 0$ and $c_3 > -1$.

3.2.2 Assumption on g

The approach taken here is to express the indeterminate forms in g_{ij},

$$\int_0^1 \cos(sx)ds = \begin{cases} \frac{\sin(x)}{x} & \text{for} \quad x \neq 0 \\ 1 & \text{for} \quad x = 0 \end{cases},$$

$$\int_0^1 \sin(sx)ds = \begin{cases} \frac{1-\cos(x)}{x} & \text{for} \quad x \neq 0 \\ 0 & \text{for} \quad x = 0 \end{cases},$$

as,

$$\lim_{x\to 0} \tfrac{\sin(x)}{x} = 1, \quad \lim_{x\to 0} \tfrac{1-\cos(x)}{x} = 0. \tag{19}$$

by applying L'Hopital's rule. Using (19) and the properties of vector norms, and given that the Frobenius norm and the 2-norm are consistent, it is possible to obtain explicit expressions for the bounds on g_{ij}.

3.2.3 Local Exponential Stability of $\xi_2(t) = 0$

Consider subsystem $\dot{\xi}_2(t) = f_2(t, \xi_2(t), \xi_2(t-2\tau))$. Clearly, $\dot{p}_{3_e}(t) = K_\theta \sin p_{3_e}(t-2\tau)$ can be linearized and will be exponentially stable provided $K_\theta < 0$. On the other hand, $\dot{z}_{3_e}(t) = -c_1 \sin z_{3_e}(t) + K_\theta \sin p_{3_e}(t)$ can also be linearized. As with $\dot{\xi}_1(t)$, $\dot{z}_{3_e}(t)$ is perturbed through a coupling term that will vanish as $p_{3_e}(t)$ converges to zero. Requiring that $c_1 > 0$ will ensure exponential stability of the remaining linearized dynamics. This completes the proof. □

4 EXPERIMENTAL RESULTS

Two equivalent multi-robot platforms exist at the Eindhoven University of Technology (TU/e) and at the Tokyo Metropolitan University (TMU). The bilateral time-delay compensation scheme is implemented in them, meaning that a mobile robot located at TU/e can be controlled from TMU and viceversa.

4.1 Experimental Platform Description

The experimental platforms' design objectives encompass cost, reliability and flexibility. The hardware and software choices together with the implementation of the setup at TU/e are discussed in greater detail in (van den Broek, 2008) (cf. Fig. 3). The setup has already been used to implement cooperation, coordination, collision avoidance and servo vision algorithms. The platform at TMU has similar characteristics, only differing from the one at TU/e in its size and vision calibration algorithm.

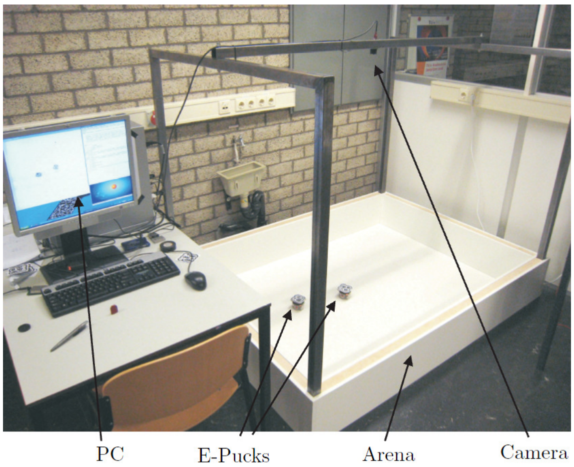


Figure 3: Experimental setup at TU/e.

Mobile Robot. The unicycle selected is the e-puck, (Mondada et al., 2009), whose wheels are driven by stepper motors that receive velocity control commands over a BlueTooth connection.

Vision. Each robot is fitted with a fiducial marker of 7 by 7 cm, collected by an industrial FireWire camera, interpreted in the program reacTIVision (reacTIVision, 2009), and calibrated by means of a global transformation (TU/e) or a grid (TMU).

Driving Area. The driving area is of 175 × 128 cm for TU/e and 100 × 50 cm for TMU, and is determined based on the required accuracy, the camera lens, and the height at which the camera is positioned.

Software. The e-puck robots and reacTIVision's data stream can be managed in C, Python, or Matlab script. In this work, the controller implementation and signal processing is carried out in Python, (Python, 2009).

Bandwidth and Sampling Rate. Using vision as the localization technique diminishes the system's bandwidth and results in a sampling rate of 25Hz.

4.2 Data Exchange over the Internet

Due to its widespread availability and low cost, the Internet is chosen as the communication channel to exchange data between TU/e and TMU.

Data Exchange. A Virtual Private Network (VPN) is established between TU/e and TMU in order to implement a reliable and secure data exchange.

Socket Configuration. Data is exchanged between TU/e and TMU as soon as it becomes available using non-blocking Transmission Control Protocol (TCP) sockets running the Internet Protocol (IP). The system's low bandwidth allows the use of the TCP, which guarantees reliable and orderly data delivery.

Data Payload. The variables exchanged amount to the current time instant and control signals from the control side to the system, and to the position and orientation values from the system side to the controller.

4.3 Implementation Issues

One of the main implementation issues of the proposed time-delay compensation strategy is the accurate modeling and characterization of the time-delay induced by the communication channel. The use of predictor-like schemes is often discouraged because of their sensitivity to delay model mismatches, (Hokayem and Spong, 2006), specially when considering nonlinear systems and a communication channel such as the Internet. To this end, three methods that ease the implementation of the proposed compensation strategy are suggested. Their objective it to bring $\tilde{\tau}$ as close as possible to τ in practice.

Delay Measurement. The round trip delay between TU/e and TMU (and viceversa) has been measured during different times of the day, for a variable amount of time, and for a total time of around 60min. The mean delay value is approximately 265ms for

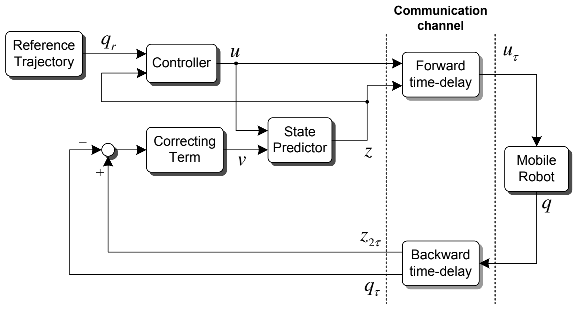


Figure 4: Time-delay compensation scheme block diagram representation without time-delay models.

both cases (267.4917ms TU/e-TMU, 269.5307ms TMU-TU/e). Occurrences of delays greater than 300ms where of 0.27% for TU/e-TMU and 0.34% for TMU-TU/e. Thus, the round trip delay can be modeled with enough accuracy even if the Internet is considered as the communication channel.

Time-stamping. Outgoing and incoming data on the controller side can be time-stamped in order to estimate the round trip delay for each pair of control signals and sensor data, setting the estimator's delay model accordingly.

Signal Bouncing. The estimator's output may be sent together with the control signals to the mobile robot, and then sent back to the controller without being modified. By using the communication channel itself to delay the estimator's output, modeling the time-delay is no longer necessary (cf. Figure 4).

4.4 Experiments

In the first experiment a mobile robot at TMU is controlled from TU/e. The reference trajectory is a lemniscate with center at [0.5m, 0.25m], a length and width of 0.2m, and a velocity multiplier of 0.2m/s. The scenario repeats in the second experiment, where a sinusoid with origin at [0.1m, 0.25m], an amplitude of 0.15m, an angular frequency of 0.3rad/s, and a velocity multiplier of 0.01m/s constitutes the reference.

The system's initial condition is $q(0) = [0.3235\text{m } 0.1882\text{m } 0.2851\text{rad}]^T$ for the first experiment and $q(0) = [0.0225\text{m } 0.1821\text{m } 0.3916\text{rad}]^T$ for the second one. In both cases the estimator's initial condition is set to $z(0) = [0\ 0\ 0]^T$, the controller gains to $c_1 = 1.0$, $c_2 = c_3 = 2.0$ and the correcting term gains to $K_x = K_y = K_\theta = -0.6$. The sampling rate is 25Hz and the experiments' duration is 60sec and 120sec respectively. The round trip delay is modeled as 265ms based on measurements, although the estimator's output is in fact delayed 280ms since only delay models which are multiples of 0.04 are allowed due to the setup's sampling time, meaning data is displaced 7 locations within the storage buffer.

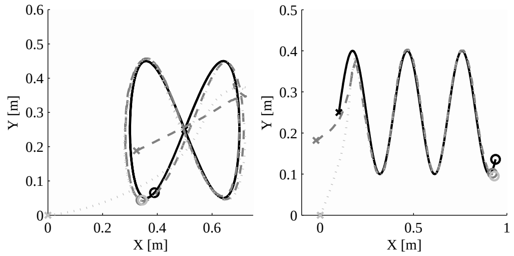


Figure 5: Reference, robot and predictor behavior in the X-Y plane for two different trajectories.

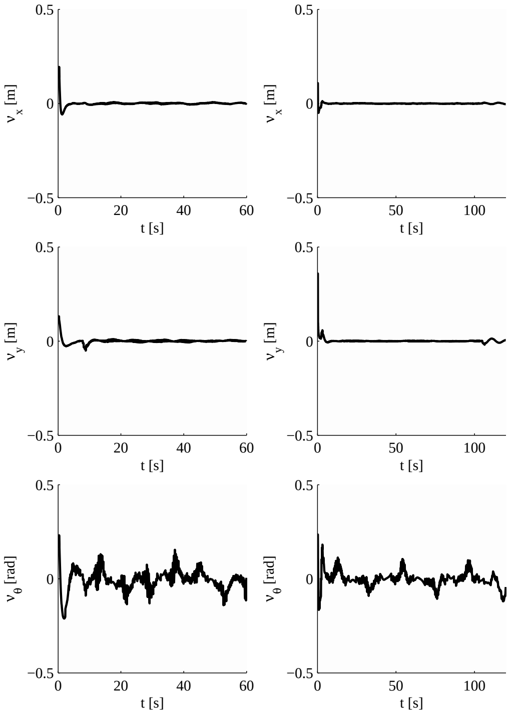


Figure 6: Practical convergence of the correcting terms.

The experimental results are shown in Figure 5 and 6 for both experiments. The first plots show the reference (black), robot (gray) and predictor (light gray) trajectories in the X-Y plane, with their initial and final position marked with a cross and a circle respectively. The plots in Figure 6 show the evolution of the correcting terms $\nu_x(t)$, $\nu_y(t)$, and $\nu_\theta(t)$ and how they practically converge to zero even in the presence of a delay model mismatch and considering a time-varying communication channel. The behavior of the proposed delay compensation strategy is consistent with the stability analysis presented and the tracking performance of the robot can be ensured even under a bilateral time-delay.

5 DISCUSSION

A compensation strategy to account for the negative effects of a bilateral time-delay affecting a unicycle-type mobile robot has been proposed. Several techniques to accurately reproduce the time-delay in the estimator have been presented, since this is required and often a cause of concern in predictor-like control strategies. Experiments show that the delay compensation strategy is robust to small delay model mismatches and delay variations using the Internet as a communication channel.

Future work includes a robustness analysis of the delay compensation scheme and a comparison with other predictor-like strategies. Additionally, techniques to synchronize as precisely as possible the controller and system sides are being studied in order to obtain a more accurate measurement of the system's performance. Integrating the concept of remote control of mobile robots with notions related to the long distance synchronization of robotic networks and extending the concepts of the estimator to a more general setting (other mechanical systems) also remain topics to be addressed in the future.

ACKNOWLEDGEMENTS

The first author would like to dedicate this paper to his father, Prof. Jaime Alvarez-Gallegos of CINVESTAV, México, who was recently honored because of his 60th birthday. Additionally, the authors acknowledge the help of Hisashi Katsumata in conducting experiments. This work has been supported by CONACYT and SEP, México, by the Graduate School of Science and Engineering at TMU and the Japan Society for the Promotion of Science (JSPS) through Grant-in-Aid for Scientific Research (No. 20560424).

REFERENCES

Brockett, R. (1983). *Differential geometric control theory*, chapter Asymptotic stability and feedback stabilization, pages 181–191. Birkhäuser.

Heemels, W. P. M. H., Teel, A. R., van de Wouw, N., and Nesic, D. (2010). Networked control systems with communication constraints: Tradeoffs between sampling intervals, delays and performance. *IEEE Trans. Automat. Contr.*, Accepted.

Hokayem, P. F. and Spong, M. W. (2006). Bilateral teleoperation: An historial survey. *Automatica*, 42(12):2035–2057.

Jakubiak, J., Lefeber, E., Tchón, K., and Nijmeijer, H. (2002). Two observer-based tracking algorithms for a unicycle mobile robot. *Int. J. Appl. Math. Comput. Sci.*, 12(4):513–522.

Kanayama, Y., Kimura, Y., Miyazaki, F., and Noguchi, T. (1990). A stable tracking control method for an autonomous mobile robot. In *Proc. IEEE Int. Conf. Rob. Automat. (ICRA)*, pages 384–389.

Kojima, K., Oguchi, T., Alvarez-Aguirre, A., and Nijmeijer, H. (2010). Predictor-based tracking control of a mobile robot with time-delays. In *Proc. 8th IFAC Symposium on Nonlinear Control Systems (NOLCOS)*. (Accepted).

Kostic, D., Adinandra, S., Caarls, J., van de Wouw, N., and Nijmeijer, H. (2009). Collision-free tracking control of unicycle mobile robots. In *Proc. 48th IEEE Conf. Dec. Control (CDC/CCC)*, pages 5667–5672.

Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., Klaptocz, A., Magnenat, S., Zufferey, J.-C., Floreano, D., and Martinoli, A. (2009). The e-puck, a robot designed for education in engineering. In *Proc. 9th Conference on Autonomous Robot Systems and Competitions*, pages 59–65.

Niemeyer, G., Preusche, C., and Hirzinger, G. (2008). *Springer Handbook of Robotics*, chapter 31: Telerobotics, pages 741–758. Springer-Verlag.

Oguchi, T. and Nijmeijer, H. (2005a). Control of nonlinear systems with time-delay using state prediction based on synchronization. In *Proc. EUROMECH Nonlinear Dynamics Conference (ENOC)*, pages 1150–1156.

Oguchi, T. and Nijmeijer, H. (2005b). Prediction of chaotic behavior. *IEEE Trans. on Circ. and Syst. I*, 52(11):2464–2472.

Oguchi, T. and Nijmeijer, H. (2006). Anticipating synchronization of nonlinear systems with uncertainties. In *6th IFAC Workschop on Time-Delay Systems*.

Panteley, E., Lefeber, E., Loría, A., and Nijmeijer, H. (1998). Exponential tracking control of a mobile car using a cascaded approach. In *IFAC Workshop on Motion Control*.

Panteley, E. and Loría, A. (1998). On global uniform asymptotic stability of nonlinear time-varying systems in cascade. *Syst. Contr. Lett.*, 33(2):131–138.

Python (2009). *Python Programming Language*. http://www.python.org.

reacTIVision (2009). *reacTIVision 1.4: A toolkit for tangible multi-touch surfaces*. http://reactivision.sourceforge.net/.

Tipsuwan, Y. and Chow, M.-Y. (2003). Control methodologies in networked control systems. *Contr. Eng. Pract.*, 11(10):1099–1111.

van den Broek, T. (2008). *Formation Control of Unicycle Mobile Robots: Theory and Experiments*. Master's thesis, Eindhoven University of Technology.

van den Broek, T., van de Wouw, N., and Nijmeijer, H. (2009). Formation control of unicycle mobile robots: A virtual structure approach. In *Proc. 48th IEEE Conf. Dec. Control (CDC/CCC)*, pages 3264–3269.

Voss, H. U. (2000). Anticipating chaotic synchronization. *Phys. Rev.*, 61(15):5115–5119.

PERFORMANCE OF HIGH-LEVEL AND LOW-LEVEL CONTROL FOR COORDINATION OF MOBILE ROBOTS

Sisdarmanto Adinandra, Jurjen Caarls, Dragan Kostić and Henk Nijmeijer
Dept. of Mechanical Engineering, Eindhoven University of Technology, PO Box 513 5600MB, Eindhoven, The Netherlands
{s.adinandra, j.caarls, d.kostic, h.nijmeijer}@tue.nl

Keywords: Coordinated Control, Non-holonomic Systems, High-level and Low-level Control, Collision Avoidance, Performance Evaluation.

Abstract: We analyze performance of different strategies for coordinated control of mobile robots. By considering an environment of a distribution center, the robots should transport goods from place A to place B while maintaining the desired formation and avoiding collisions. We evaluate performance of two collision avoidance strategies, namely a high-level and low-level collision avoidance approach, each using different feedback information and update rate. As performance measure we take into account the time to accomplish the transportation task and the tracking errors of the robots. Evaluation is done in several experiments with seven mobile robots.

1 INTRODUCTION

A group of mobile robots can be used to realize spatially distributed transportation tasks in a distribution center. When several robots are employed in a shared environment, then motion coordination and cooperation between these robots can be introduced in order to increase robustness in the execution of their tasks.

Transportation in a distribution center is typically carried out by means of conveyors. Unfortunately, a failure in a single conveyor can disable a part of the transportation. To increase robustness of transportation, one can introduce redundant conveyers. This solution requires extra investments and occupies additional space. An appealing alternative is to substitute conveyers with mobile robots (Giuzzo, 2008). Unlike conveyers, the robots can dynamically alter their trajectories to avoid obstacles and complete assigned tasks. In addition, if one robot fails, its task can be delegated to another one. While engaging a number of mobile robots simultaneously, efficient robot coordination and cooperation control strategies are required to achieve high throughput in transportation of goods without collisions.

In the distribution center, planning and scheduling tasks are normally done in a centralized way. A high-level planning system decides which customers orders must be executed, together with the decision on how the tasks must be accomplished, see e.g. (van den Berg, 1999) and (Gu, *et al.,* 2007). Whereas this centralized approach can provide the optimal throughput in the absence of uncertainties, it can show quite some weaknesses in exceptional situations, such as when unexpected obstacles appear or when unknown disturbances start affecting the robots. In such situations, re-planning must be accomplished with time limitations that often lead to transportation plans that are not optimal in terms of the throughput. A viable alternative to the centralized approach is to facilitate negotiations among the robots and the supervisor which assigns the tasks. Through negotiation, the robots and the supervisor can dynamically adapt the transportation plans such as to make them less sensitive to different types of failures. The result might not have the optimal throughput, but it will likely bring higher robustness in comparison with the centralized approach. The high level coordination can be solved, for instance, using the holonic approach (Giret and Botti, 2004), (Moneva, *et al.,* 2009). Another example of the high-level coordination is decentralized control of Automated Guided Vehicles for distribution centers (Farahvash and Boucher, 2004), (Weyns, *et al.,* 2005*)*. In both works, a multi agent system is proposed for transportation planning, distribution of tasks, and collision avoidance.

At the layer of low-level control of individual mobile robots, various techniques of robot coordination and cooperation can be used to realize the transportation, such as leader-follower, behavior based, virtual structure, etc, (Ren and Beard, 2004), (Mastellone, *et al.*, 2008), (Sun *et al.*, 2009), (van den Broek *et al.*, 2009). In these approaches, the motion controllers achieve tracking of individual robot trajectories, on the one hand, and maintain the desired spatial formation between the robots, on another.

To the best of our knowledge, little research is devoted to performance comparison between the high-level and low-level control techniques. Some research have been conducted on performance comparison of different high-level techniques, see e.g. (Vis, 2004), (Le-Anh and De Koster, 2006), (Gu, *et al.*, 2010) and references therein. Furthermore, a quantitative evaluation of robustness of high-level control is still lacking, and no data are reported that illustrate how complicated it can be for the high-level control to find the optimal solution. Finally, there is scarce research on appropriate combination of high- and low-level control that can handle both effectiveness and robustness at the same time.

The lack of information has motivated us to quantitatively compare performance of high-level and low-level control of a group of mobile robots that perform a task which resembles coordinated transportation of goods in a distribution center. An ideal situation is simulated as a basis for comparison. We evaluate performance using relevant indicators, such as time to accomplish the task and errors in tracking the desired robot trajectories.

Two main contributions of this paper are: (*i*) suboptimal solutions for motion coordination of mobile robots, namely pure high-level and pure low-level coordination and (*ii*) experimental performance evaluation of these two strategies for completing transportation tasks in distribution center like environments.

This paper is organized as follows. In Section 2 we present necessary mathematical models and tools. Section 3 explains strategies to coordinate the motion of the mobile robots. Section 4 reports on experimental results and highlights the main findings of our performance analysis. Conclusions and discussion on future work are given in Section 5.

2 PRELIMINARIES

2.1 Unicycle Mobile Robots

We consider a group of *m* mobile robots that are described by the non-holonomic kinematic model of a unicycle, as depicted in Figure 1:

$$\begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\theta}_i \end{bmatrix} = \begin{bmatrix} v_i \cos\theta_i \\ v_i \sin\theta_i \\ \omega_i \end{bmatrix}. \qquad (1)$$

Here, v_i and ω_i are the forward and steering velocities, respectively, x_i and y_i are the Cartesian coordinates of the robot midpoint $\mathbf{O_i}$ in the world coordinate frame $\mathbf{O_{xy}}$, θ_i is the heading angle relative to the *x*-axis of the world frame, and $i \in \{1,2,3,\dots,m\}$. The reference trajectory of each robot i is given in the frame $\mathbf{O_{xy}}$:

$$\boldsymbol{P}_{ri}(t) = [x_{ri}(t) \quad y_{ri}(t) \quad \theta_{ri}(t)]^{\boldsymbol{T}}. \qquad (2)$$

The trajectories of all mobile robots constitute a time-varying formation. An example is shown in Figure 2, where a platoon-like formation is adopted.

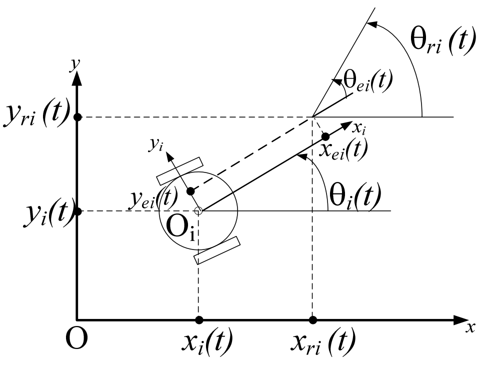


Figure 1: Configuration and error coordinates of a unicycle mobile robot.

2.2 Trajectory Tracking Control

To follow its own reference trajectory and to maintain the assigned formation, for each robot we propose the following control laws v_i and ω_i

$$v_i = v_{ri}\cos\theta_{ei} + k_{xi}x_{ei} + \sum\nolimits_{j=1,j\neq i}^{m} s_x^{ij} k_x^{ij} x_{ej} \frac{(x_{ei} - x_{ej})}{\sqrt{1 + x_{ei}^2 + x_{ej}^2}}, \qquad (3a)$$

$$\omega_i = \omega_{ri} + k_{yi}v_{ri}y_{ei}\frac{\sin\theta_{ei}}{\theta_{ei}} + k_{\theta i}\theta_{ei} \qquad (3b)$$

$$+\sum\nolimits_{j=1,j\neq i}^{m} s_y^{ij} k_y^{ij} \left(y_{ei} - y_{ej}\right) \frac{\sin\theta_{ei}}{\theta_{ei}} \sin\theta_{ej} + \sum\nolimits_{j=1,j\neq i}^{m} s_\theta^{ij} k_\theta^{ij} \theta_{ej} \frac{\left(\theta_{ei} - \theta_{ej}\right)}{\sqrt{1 + \theta_{ei}^2 + \theta_{ej}^2}}.$$

where x_{ei}, y_{ei}, θ_{ei} are the tracking errors represented in the robot coordinate frame $\mathbf{O}_{\mathbf{x_i y_i}}$

$$\begin{bmatrix} x_{ei} \\ y_{ei} \\ \theta_{ei} \end{bmatrix} = \begin{bmatrix} \cos\theta_i & \sin\theta_i & 0 \\ -\sin\theta_i & \cos\theta_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{ri} - x_i \\ y_{ri} - y_i \\ \theta_{ri} - \theta_i \end{bmatrix}, \tag{3c}$$

k_x, k_y, k_θ are the design parameters that influence the performance of trajectory tracking, while $k_x^{ij}, k_y^{ij}, k_\theta^{ij}$ are the design parameters that influence formation keeping. Furthermore, $s_\alpha^{ij}, \alpha \in \{x, y, \theta\}$ is defined as follows:

$$s_\alpha^{ij} = \begin{cases} \operatorname{sgn}(\alpha_{ei}) & |\alpha_{ei}| \geq |\alpha_{ej}| \\ \operatorname{sgn}(\alpha_{ej}) & |\alpha_{ei}| < |\alpha_{ej}| \end{cases}. \tag{3d}$$

The control laws (3a,b) guarantee globally asymptotic tracking of $\boldsymbol{P}_{ri}(t)$ and represent a non-saturated version of the controller proposed in (Kostić, *et al.*, 2009) and (Kostić, *et al.*, 2010).

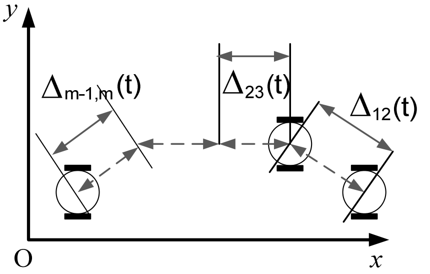


Figure 2: An example of a convoy-like, time-varying formation.

2.3 Penalty Functions

We introduce a set P_γ of continuous, monotone and bounded penalty functions indexed by a constant parameter $\gamma \in \mathbb{R}^+$ (Kostić, *et al.*, 2009):

$$\begin{aligned} P_\gamma = \{\delta_\gamma : \mathbb{R} \to \mathbb{R} \,|\, &\delta \text{ is continous, monotone},\\ &\delta_\gamma = \delta_{\gamma,min} \text{ if } x \in \mathbb{R}^+,\\ &\delta_{\gamma,min} \leq \delta_\gamma(x) \leq \delta_{\gamma,max}\\ &\text{if } \gamma_{min} \leq x \leq \gamma_{max},\\ &\delta_\gamma(x) = \delta_{\gamma,min} \text{ if } x \geq \gamma_{max}\}. \end{aligned} \tag{4}$$

An example of a function in P_γ is

$$\delta_\gamma(x) = \begin{cases} \delta_{\gamma,max}, & x < \gamma_m \\ 1 - \Delta_\gamma\left(\lambda\left(x - \gamma_{min} - \frac{\sin\psi\,(x - \gamma_{min})}{\psi}\right)\right), & \gamma_{min} \leq x \leq \\ \delta_{\gamma,min}, & x > \gamma_m \end{cases} \tag{5}$$

where $\Delta_\gamma = \gamma_{max} - \gamma_{min}$ and $\psi = 2\pi / (\gamma_{max} - \gamma_{min})$.

3 HIGH-LEVEL AND LOW-LEVEL COORDINATION

To achieve transportation with high throughput and increase robustness to uncertainties, such as disturbances, all engaged robots have to cooperate. These robots have to coordinate their motions such as to avoid collisions and keep the sequence as assigned by the given formation.

In this research, we investigate the performance of two coordination methods, namely high-level coordination and low-level coordination. The coordination takes care of collision avoidance and keeping the desired robot sequence. Both aspects are very relevant for realization of transportation tasks in a distribution center environment.

3.1 High-level Coordination

The high-level method of coordination is implemented at the level of generating the reference trajectory (2) for each robot. This method assumes that the robots move from waypoint to waypoint (nodes) on a network of fixed path segments (edges). This network enables us to define spatial reference trajectories.

To accomplish a given transportation task, a timed desired trajectory along the desired path segments is generated by each robot. For collision avoidance it is needed to coordinate (predicted) intervals of appearance of the robots at intersecting segments. If these time intervals are well coordinated, then the absence of collisions is guaranteed as long as each robot accurately follows its own reference trajectory. In this case, the coupling gains k_x^{ij}, k_y^{ij} and k_θ^{ij} between robots in the control laws (3) have to be set to zero. Figure 3 illustrates this coordination method in a situation where trajectories of three robots intersect at junction J1.

The junction is represented by a group of intersecting path segments. Each segment brings a robot from one side of the junction to another. Therefore, the interval a robot occupies the junction is marked by passing the beginning and ending waypoints of such a segment.

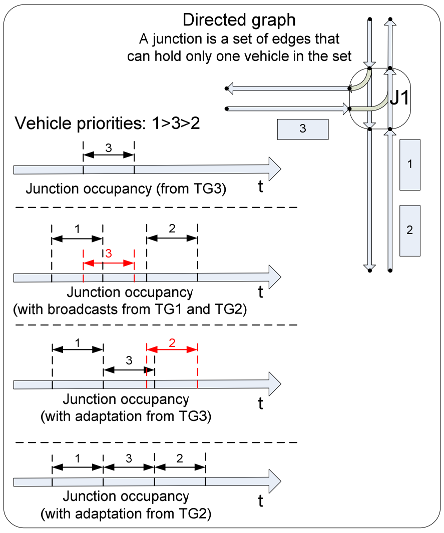

Directed graph
A junction is a set of edges that can hold only one vehicle in the set
J1
Vehicle priorities: 1>3>2
3
1
2
3
Junction occupancy (from TG3)
t
1
2
3
Junction occupancy
(with broadcasts from TG1 and TG2)
t
1
2
3
Junction occupancy
(with adaptation from TG3)
t
1
3
2
Junction occupancy
(with adaptation from TG2)
t

Figure 3: High-level coordination for collision avoidance.

At regular intervals, currently 1s, each robot broadcasts its occupation interval to other robots headed to J1. If overlap is detected, the robot of the highest priority will rebroadcast its occupation interval immediately to notify the other robots.

The robot of lower priority will postpone its arrival time at the junction until the leaving time of the robot of higher priority. Each robot adjusts its speed immediately such as to reach the junction at the correct time. The necessary speed is calculated and all passing times at the waypoints between the robot and the junction are adapted accordingly.

Similarly, the passing times after the junction should be changed too. From the junction segment on, the passing times are only adapted when the robot would require to exceed its speed limits. Once both the entering and leaving waypoint passing times on the junction are known, the robot broadcasts its new non-overlapping occupation interval, and waits again for incoming occupation intervals.

The priority of a robot is determined as the sum of the total time this robot has to wait for other ones, excluding those in front of it, and the expected time to enter the junction. This is similar to the first come first served priority scheme, which minimizes the queues in front of the junctions as well as the waiting times of the individual robots.

3.2 Low-level Coordination

The low-level coordination is implemented at the level of trajectory tracking control. To keep the correct robot sequence according to the desired spatial pattern, the coupling gains k_x^{ij}, k_y^{ij}, and k_θ^{ij} are set to positive non-zero values. With such gains, the control law (3) will be able to track the desired formation, e.g. as shown in Figure 2. Due to this coupling, the robots can recover their formation faster than their individual desired paths, thereby maintaining the desired sequence of robots better than without the coupling.

To gain more robustness to uncertainties, the reference forward velocity $v_{ri}, i \in \{1,2,3,\dots,m\}$, of each robot is also adjusted using the penalty functions from the set P_γ defined by (4). For each robot we determine the distance between its reference and actual position:

$$\Delta_{rp,i} = \sqrt{(x_{ri} - x_i)^2 + (y_{ri} - y_i)^2}. \tag{6}$$

If $\Delta_{rp,i} > \gamma_{min}$, then the desired forward velocity of each robot i is penalized as follows:

$$v_{ri} = v_{des,i} \prod_{j=1}^{m} \delta_{\gamma j}(\Delta_{rp,j}). \tag{7}$$

Here, $\delta_{\gamma j}$ is a penalty function from the set (4) and $v_{des,i}$ is the desired forward velocity. Using (7), if one robot is far from the assigned path, then the forward velocities of all robots are decreased in order to give the perturbed robot time to get back on its track. In this way, it will be easier to keep the correct sequence of the robots.

To avoid collisions, we make use of an Artificial Potential Field (APF) (Latombe, 1991). The reference trajectories of the robots facing collisions are online adapted using the APF, mimicking the approach in (Kostić, *et al.*, 2010). If $\boldsymbol{q}_i = (x_i, y_i)$ and $\boldsymbol{q}_j = (x_j, y_j)$, $i,j \in \{1,2,\dots,m\}$, are the Cartesian coordinates of the robots i and j, respectively, then an APF of robot i is:

$$V_i = \sum_{j=1, j\neq i}^{m} V_{i,j}(\boldsymbol{q}_i, \boldsymbol{q}_j) + \rho_{att}(\boldsymbol{q}_i, \boldsymbol{P}_{ri}), \tag{8}$$

where

$$\rho_{att}(\boldsymbol{q}_i, \boldsymbol{P}_{ri}) = \frac{1}{2} K_a((x_i - x_{ri})^2 + (y_i - y_{ri})^2), \tag{9}$$

$$V_{i,j}(\boldsymbol{q}_i, \boldsymbol{q}_j) = \begin{cases} K_o e^{-\frac{1}{2}\left(\frac{(x_i - x_j)^2}{\alpha} + \frac{(y_i - y_j)^2}{\beta}\right)}, & \|\boldsymbol{q}_i - \boldsymbol{q}_j\|_2 - rob_d < d_s \, . \\ 0, & \text{elsewhere.} \end{cases} \tag{10}$$

Here, K_o and K_a are the gains of the repulsive and attractive fields, respectively, α and β are positive real numbers that determine the size of the detection region for the repulsive function, rob_d is the robot diameter, and d_s is the threshold of the detection region. When all obstacles are outside the detection region of robot i, its tracking controller tracks the trajectory $\boldsymbol{P}_{ri}(t)$. Inside the detection region, the low-level coordination modifies the trajectory as follows:

- Determine the Cartesian velocities that move the robot in the direction of the steepest descent of V_i (away from the obstruction):

$$\begin{bmatrix}\delta v'_{x,i}\\ \delta v'_{y,i}\end{bmatrix} = -\begin{bmatrix}\frac{\partial V_i}{\partial x_i}\\ \frac{\partial V_i}{\partial y_i}\end{bmatrix}, \delta v_i = \sqrt{\left(\delta v'_{x,i}\right)^2+\left(\delta v'_{y,i}\right)^2}. \qquad (11)$$

- Update the reference trajectory such that collision avoidance is achieved at time-instant t_k:

$$x_{ri}(t_k) = x_i(t_{k-1}) + (t_k - t_{k-1})\delta v_i, \qquad (12a)$$

$$y_{ri}(t_k) = y_i(t_{k-1}) + (t_k - t_{k-1})\delta v_i, \qquad (12b)$$

$$\theta_{ri}(t_k) = \begin{cases}\operatorname{atan}(\delta v'_{y,i}/\delta v'_{x,i}), & \text{if } \delta v_i > 0,\\ \theta_{ri}(t_{k-1}), & \text{if } \delta v_i = 0,\end{cases} \qquad (12c)$$

$$v_{ri}(t_k) = \sqrt{\left(\delta v'_{x,i}\right)^2+\left(\delta v'_{y,i}\right)^2}, \qquad (12d)$$

$$\omega_{ri}(t_k) = \frac{\theta_{ri}(t_k)-\theta_{ri}(t_{k-1})}{t_k - t_{k-1}}. \qquad (12e)$$

4 EXPERIMENTS AND PERFORMANCE ANALYSIS

4.1 Scenario

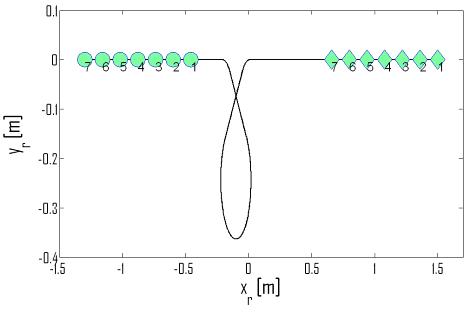


Figure 4: The experimental robot path. Symbols “o” and “◊” indicate the start and end positions of the robots, respectively.

To mimic a realistic transportation task in a distribution center, experiments are conducted according to the following scenario: a convoy of seven robots delivers goods along a path depicted in Figure 4. At one segment of this path, the front part of the convoy intersects with the part at the back; consequently, robot coordination is needed to avoid collisions between the robots and to keep the correct robot sequence. The desired forward velocity of each robot is 0.08 [m/s].

For performance evaluation, we use the following indicators:

1. The average of the travel times, $\overline{t_{tvrl}}$, that the robots consume to reach their target locations:

$$\overline{t_{tvrl}} = \frac{\sum_{i=1}^{7} t_{tvrl,i}}{7}. \qquad (13)$$

2. The normalized total tracking error of all robots:

$$e_{xy,tot} = \sum_{i=1}^{7}\sqrt{\frac{1}{l}\sum_{k=1}^{l}\left(x_{ro,i}(t_k) - x_i(t_k)\right)^2 + \left(y_{ro,i}(t_k) - y_i(t_k)\right)^2}. \qquad (14)$$

where t_k is the moment of collecting data, l is the number of data points in the experiment, $x_{ro,i}$ and $y_{ro,i}$ are the originally assigned robot reference, while x_i and y_i are the actual positions.

3. The normalized total formation error. In the experiments we use a platoon-like formation, which has a spatial pattern described by a time-varying Euclidean distance between the neighboring robots. For seven robots, the pattern can be described for $i \in \{1,2,3,4,5,6\}$ and $j = i + 1$, as follows:

$$\Delta_{rij}(t) = \sqrt{\left(x_{ro,i}(t) - x_{ro,j}(t)\right)^2 + \left(y_{ro,i}(t) - y_{ro,j}(t)\right)^2}. \qquad (15)$$

We define the individual formation error by

$$\delta_{ij}(t) = \Delta_{rij}(t) - \Delta_{ij}(t), \qquad (16)$$

where $\Delta_{ij}(t)$ is the actual Euclidean distance between the robots i and j. Thus, the normalized total formation error is given by:

$$\delta_{ij,tot} = \sum_{i=1}^{6}\sum_{j=i+1}^{7}\sqrt{\frac{1}{l}\sum_{k=1}^{l}\left(\delta_{ij}(t_k)\right)^2}. \qquad (17)$$

For a total performance indicator, we take the summation of the three indicators proposed above with equal weight:

$$\sum\nolimits_{perf} = \overline{t_{tvrl}} + e_{xy,tot} + \delta_{ij,tot}. \qquad (18)$$

4.2 Experimental Set-up

Our experimental setup is depicted in Figure 5. We use mobile robots, model e-puck (Mondala and Bonani, 2007), a camera as a localization device for getting the position and orientation of all robots, and a PC. The PC generates robot trajectories, processes camera images to get the actual pose of the robots, and runs the collision avoidance algorithms and tracking control laws for all the robots. The PC sends the control velocities to the robots via a BlueTooth protocol. This way of implementation is chosen due to the limiting processing power of the onboard robot processors and due to the limited bandwidth of the BlueTooth communication.

4.3 Experimental Results and Analysis

In the experiments, we use the following design parameters:

$$k_x = 0.4,\ k_y = 100,\ k_\theta = 0.5, \tag{19a}$$

$$k_x^{ij} = 0.06, k_y^{ij} = 10, k_\theta^{ij} = 0.00001, \tag{19b}$$

$$K_0 = 20,\ K_a = 10, \alpha = \beta = 0.05, \tag{19c}$$

$$d_s = 0.03\ [\mathrm{m}],\ \Delta_{rij} = 0.14\ [\mathrm{m}], \tag{19d}$$

$$\gamma_{min} = 0.05\ [\mathrm{m}], \gamma_{max} = 0.5\ [\mathrm{m}], \tag{19e}$$

$$\delta_{\gamma i}(\Delta_{rp,i})_{min} = 0.3,\ \delta_{\gamma i}(\Delta_{rp,i})_{max} = 1. \tag{19f}$$

The values of k_x, k_y, k_θ are chosen such that we have an accurate trajectory tracking. As for $k_x^{ij}, k_y^{ij}, k_\theta^{ij}$, the values will be zero if high-level coordination is active. When low-level coordination is active, the values are chosen such that we have strong coupling between the robots, especially in x and y direction.

As a basis for comparison, we simulate the case where the coupling terms in the control law (3) are enabled and collisions are allowed. In this unrealistic case, all the robots can travel without perturbation from the start to the end position, while keeping the formation. To account for realistic imperfections of the vision system used in experiments, we add simulated noise, drawn from a normal distribution with zero mean and standard deviation of ±0.005 [m] for x and y, and $\pm 0.5^0$ for θ, in accordance with the measurement noise of the real camera. The following results are obtained:

$$\overline{t_{tvrl}} = 36.06\ [\mathrm{s}], e_{xy,tot} = 0.0024[\mathrm{m}], \tag{20a}$$

$$\delta_{ij,total} = 0.0019[\mathrm{m}], \Sigma_{perf} = 36.06\,43[-]. \tag{20b}$$

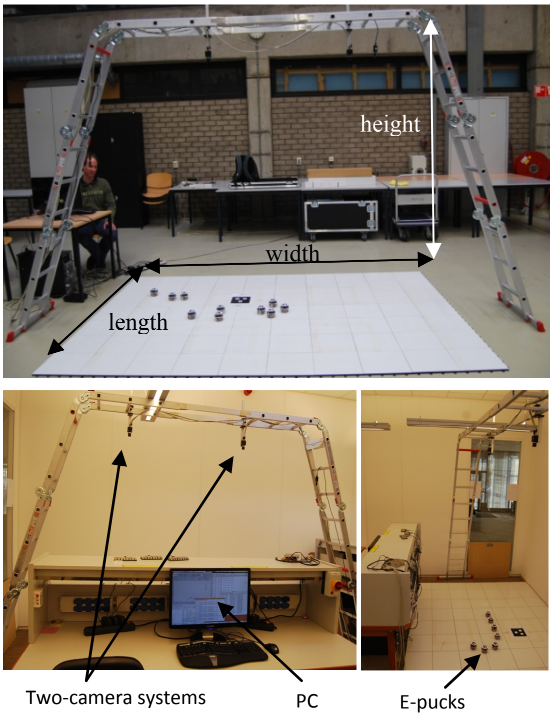


Figure 5: The experimental set-up.

The simulation results are used as a reference for comparison with data obtained from experiments. Tables 1a and 1b show statistics of all performance indicators obtained in ten repeated experiments, while Figure 6 shows a graphical representation of these statistics.

ANALYSIS. From the second column of Table 1a and Figure 6, we can see that the high-level coordination achieves the shortest average travel time with respect to other strategies. Thanks to negotiations among the robots in the high-level coordination, collision avoidance is achieved time-efficiently, which leads to short travel times.

Table 1a: Mean value and standard deviation of $\overline{t_{tvrl}}$ and $e_{xy,tot}$ in ten repeated experiments.

Strategies	Indicators			
	$\overline{t_{tvrl}}$[s]		$e_{xy,tot}$[m]	
	mean	Std	mean	Std
[1]HL	**36.06**	0	**0.105**	**0.00088**
[2]LL-1	38.939	1.920	0.907	0.4501
[3]LL-2	38.110	1.184	0.737	0.3277
[4]LL-3	38.633	1.024	0.885	0.2796
[5]LL-4	40.204	2.277	1.228	0.3470

[1]HL: high-level; [2]LL-1: low level, no coupling between robots [3]LL-2: low-level, all robots are mutually coupled; [4]LL-3: low-level, each robot is coupled to the leader robot, i.e. the first robot of the convoy, not vice versa; [5]LL-4: low-level, each robot is coupled to the robot in front of it, not vice versa.

Moreover, the travel time is identical to the one achieved under the ideal condition (20a). The given experimental result suggests that the high-level control yields time-optimal transportation while taking care of collision avoidance.

Table 1b: Mean value and standard deviation of $\delta_{ij,tot}$ in ten repeated experiments and Σ_{perf}.

Strategies	Indicators		
	$\delta_{ij,tot}$[m]		Σ_{perf}
	mean	std	
^{1}HL	**0.0782**	0.00045	36.243200
^{2}LL-1	0.0934	0.04390	39.939400
^{3}LL-2	0.0853	0.02390	38.932299
^{4}LL-3	0.0862	0.00770	39.604199
^{5}LL-4	0.1170	0.05540	41.548999

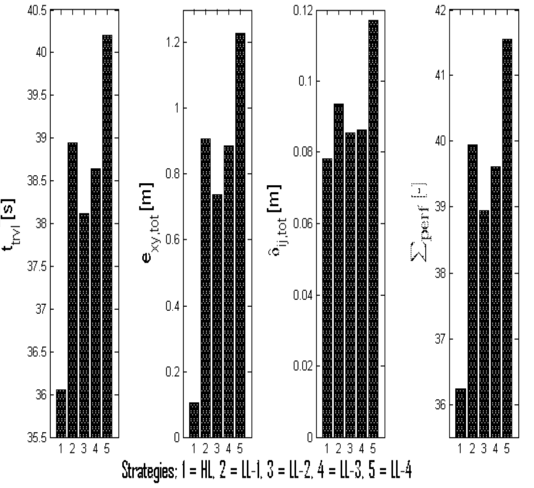


Figure 6: The mean values of $\overline{t_{tvrl}}$, $e_{xy,tot}$, $\delta_{ij,tot}$ from different coordination strategies and Σ_{perf}.

As for the low-level coordination, according to the second column of Table 1a and Figure 6, in all low-level coordination methods, the total travel time is longer than achieved using the high-level coordination. The online APF strategy does not utilize negotiations; a robot facing an imminent collision tries to move away from the obstacle. Time-optimality of such a collision avoidance algorithm is not guaranteed.

In our experiments, the lowest tracking errors are achieved using the high-level coordination. With modest uncertainties, this strategy ensures that each robot follows its own collision-free reference trajectory. Consequently, the difference between the actual and the desired robot trajectories remains small, which results in small tracking errors. This is in line with the observation that the shortest travel time is characteristic for the high level strategy. In addition, the way the high-level strategy solves the collision avoidance is also useful for formation keeping, as depicted in Figure 6.

When comparing the tracking errors, the low-level coordination strategies, all yield larger tracking errors compared to the high-level coordination. As expected, the tracking errors with coupling are slightly larger than without coupling. Having the couplings, a robot adapts its movement to other perturbed robots. In this way, the tracking control can keep the formation but by doing so, it decrease its own tracking performance. Despite the poor tracking error performance, as shown in Figure 6, introducing correct coupling helps the robots keep the formation.

If a decision needs to be made, a single measure is needed to base it on. Comparing the values of the total performance, we can observe that in our experiments, option LL-2 turns out to be the most suitable option for the low-level coordination method.

Overall, it appears that the high-level control is the most promising solution according to the total performance measure. However, this solution may fail in case of perturbations. The high-level strategy requires all robots to be in the correct position for a successful collision-free execution. If one robot is not at the correct position at some time instant, collisions may occur and the correct robot sequencing cannot be guaranteed.

The low-level coordination is inherently more robust to perturbations. Despite perturbations, the low-level coordination achieves collision avoidance and formation keeping. In the presence of perturbations, from the comparison of the indicators and considering the importance of formation, the fully coupled option (LL-2) seems to be the most appealing one. We show in Table 2 the minimum distances between the center points of all robots, measured from an experiment using the LL-2 strategy. During the experiment robot 4 was manually displaced from the platoon. Since the diameter of the robot is 0.07 [m], any value below 0.07 [m] implies a collision between robots. Table 2 shows no distances below 0.08 [m], so no collisions occurred.

Given experimental analysis brings us to the conclusion that it is not easy to find a single solution that scores best in terms of all performance indicators and is robust enough against uncertainties. There will be a set of solutions that are best, i.e. no better solutions exist in one performance indicator, without being worse in another. These solutions are called pareto-optimal solutions. One has to choose a weighing between all indicators, as in the total performance indicator (18), to select the best one for a certain application.

Table 2: Minimum distances between the centers of all robots. $d_{ci,j}$ denotes the distance between robots i and j.

min $d_{ci,j}$[m]							
robot	1	2	3	4	5	6	7
1	-	0.08	0.08	0.09	0.16	0.09	0.08
2		-	0.08	0.17	0.23	0.19	0.14
3			-	0.08	0.17	0.20	0.18
4				-	0.08	0.09	0.17
5					-	0.08	0.08
6						-	0.08

The problem remains to find those pareto-optimal solutions. The high-level and low-level coordination strategies that we propose still can be improved and optimized in terms of all performance indicators, e.g. using larger safety distance for the high level method or having a smoother transition from normal to collision mode for the low level methods. Finding the true pareto-optimal solutions is probably not possible. However, from a set of solutions one can always remove the non pareto-optimal solutions, and choose from the remaining, best, ones.

5 CONCLUSIONS

We have experimentally evaluated the performance of different strategies for coordinated control of mobile robots. We have proposed high-level and low-level coordination methods and presented experimental results that illustrate the superior performance of the high-level method in terms of time efficiency and accuracy of tracking the desired robot trajectories. A serious limitation of the high-level coordination is the requirement for accurate tracking of the reference trajectories.

Even though the performance of the low-level coordination method is worse than that of the high-level coordination method, the low-level one is inherently more robust against uncertainties.

Given the results of our analysis, it seems interesting to analyze performance of combinations of different strategies, e.g., of high-level and low-level coordination methods. By combining more strategies, we may optimize more performance indicators and meet more requirements. Consequently, it needs to be investigated which combinations would lead to the pareto-optimal solutions.

ACKNOWLEDGEMENTS

This work has been carried out as part of the FALCON project under the responsibility of the Embedded Systems Institute with Vanderlande Industries as the industrial partner. This project is partially supported by the Dutch Ministry of Economic Affairs under the Embedded Systems Institute (BSIK03021) program.

REFERENCES

Berg, J. P. van den, 1999, A literature Survey on Planning and Control of Warehousing Systems, in *IIE Transactions*, Vol.31, pp. 751-762.

Broek, T. H. A. van den, van de Wouw, N., Nijmeijer, H. 2009, Formation Control of Unicycle Mobile Robots: a Virtual Structure Approach, in *Proc. IEEE Conf. on Decision and Control,* pp. 8238-8333.

Farahvash, P., and Boucher, T. O., 2004, A Multi-agent Architecture for Control of AGV Systems, in *Robotics and Computer Integrated Manufacturing*, Vol.20, pp. 473-483.

Giret, A., and Botti, V., 2004, Holons and Agents, in *J. of Intelligent Manufacturing*, Vol. 15, pp. 645-659.

Giuzzoa, E., 2008, Three Engineers, Hundred of Robots, One Warehouse, in *IEEE Spectrum,* Vol. 45. No. 7, pp. 26-34.

Gu, J., Goetschalckx, M., McGinnis, L. F., 2007, Research on Warehouse Operation: A Comprehensive review, in in *European Journal of Operational Research,* Vol. 177, pp. 1-21.

Gu, J., Goetschalckx, M., McGinnis, L. F., 2010, Reseach on Warehouse Design and Performance Evaluation: A Comrehensive review, in *European Journal of Operational Research,* Vol. 203, pp. 539-549.

Kostić, D., Adinandra, S., Caarls, J., Nijmeijer, H., 2009, Collision-free Tracking Control of Unicycle Mobile Robots, in *Proc. IEEE Conf. on Decision and Control,* pp. 5667-5672.

Kostić, D., Adinandra, S., Caarls, J., Nijmeijer, H., 2010, Collision-free Tracking Control of Unicycle Mobile Robots, Collision-free Motion Coordination of Unicycle Multi-agent Systems, *Accepted for American Control Conference 2010.*

Latombe, J. C., 1991, *Robot Motion Planning*, Kluwer Academic Publishers, Boston, MA.

Le-Anh, De Koster, M. B. M., 2006, A review of Design and Control of Automated Guided Vehicles Systems, in *European Journal of Operational Research,* Vol. 171, pp. 1-23.

Mastellone, S., Stipanović, D. M., Graunke, C. R., Intlekofer, K. A., and Spong, M. W., 2008, Formation Control and Collision Avoidance for Multi-agent Non-holonomic Systems: Theory and Experiments, in *Int. Journal of Robotics Research*, Vol.27, No. 1, pp. 107-126.

Mondala, F., Bonani, M., 2007, E-puck education robot, www.e-puck.org.

Moneva, H., Caarls, J., and Verriet, J., 2009. A Holonic Approach to Warehouse Control, in *Int. Conf. on Practical Applications of Agents and Multiagent Systems,* pp. 1-10.

Ren, W. and Beard, W., 2004, Decentralized Scheme for Spacecraft Formation via The Virtual Structure Approach, in *IEEE Proc. Control Theory Application 2004,* pp. 268-357.

Sun, D., Wang, C., Shang, W., Feng, G., 2009, A Synchronization Approach to Trajectory Tracking of Multiple Mobile Robots While Maintaining Time-Varying Formations, *IEEE Transactions on Robotics,* Vol.25, pp. 1074-1086.

Vis, I. F. A, 2004, Survey of Research in the Design and Control of Automated Guided Vehicle Systems, in *European Journal of Operational Research,* Vol. 170, pp. 677-709.

Weyns, D., Schelfthout, K., Holvoet, T., and Lefever, T., 2005, Decentralized Control of E'GV Transportation Systems, in *Autonomous Agents and Multiagent Systems, Industry Track (Pechoucek, M. and Steiner, D. and Thompson, S., eds.),* pp. 67-74.

A HIGHLY INTEGRATED LOW PRESSURE FLUID SERVO-VALVE FOR APPLICATIONS IN WEARABLE ROBOTIC SYSTEMS

Michele Folgheraiter, Mathias Jordan, Luis M. Vaca Benitez, Felix Grimminger, Steffen Schmidt
Jan Albiez and Frank Kirchner
German Research Center for Artificial Intelligence (DFKI), Robotics Innovation Center
Robert-Hooke-Strasse 5D-28359 Bremen, Germany
{*Michele.folgheraiter, Mathias.Jorda, luis.vaca_benitez, felix.grimminger, steffen.schmidt, jan.albiez, frank.kirchner*}*@dfki.de*

Keywords: Proportional Valve, Hydraulic Valve, Pneumatic Valve, Mechatronics, Pressure Control, Servo-Mechanism.

Abstract: In this paper an innovative low pressure servo-valve is presented. The device was designed with the main aim to be easily integrable into complex hydraulic/pneumatic actuation systems, and to operate at relatively low pressure ($< 50 \cdot 10^5 Pa$). Characteristics like compactness, lightweight, high bandwidth, and autonomous sensory capability, where considered during the design process in order to achieve a device that fulfills the basic requirements for a wearable robotic system. Preliminary results about the prototype performances are presented here, in particular its dynamic behavior was measured for different working conditions, and a non-linear model identified using a recursive Hammerstein-Wiener parameter adaptation algorithm.

1 INTRODUCTION

State of the art robots that are actuated with a hydraulic system are generally thought to operate with pressures greater than $200 \cdot 10^5 Pa$ (Hayward, 1994), (Jacobsen et al., 1991). This is mainly due to the fact that it is convenient to increase the force/weight ratio of the actuation system by increasing its operational pressure (Yoshinada et al., 1992). If from one side increasing the pressure brings advantages, from the other could represent a limitation. At first the hydraulic components need to be designed to resist the high forces generated by the fluid pressure; this requires therefore to employ thick and heavy materials for pipes and actuators. Secondly, the usage of high pressure could also cause a dangerous situation for the operators that are in the proximity of the robot. The safety issue is even more critical if the robot, in our case an exoskeleton, is strictly coupled with the human being (Pons, 2008). Any failure in the hydraulic system could seriously harm the user.

One of the main goal of VI-Bot project, under development at DFKI Bremen (Robotics Innovation Center), is to design an intrinsically safe, wearable arm exoskeleton for Tele-Robotics applications (Folgheraiter et al., 2008; Folgheraiter et al., 2009a; Folgheraiter et al., 2009b; Folgheraiter et al., 2009c). As requirements the haptic interface should: enable the operator to control complex robotics systems in an intuitive way, implement a multi-points haptic feedback to increase the immersion into the work scenario, be light weight and adaptable to different users, and integrate different levels of safe mechanisms. Furthermore, the kinematics architecture of the system should be designed in order to constrain as less as possible the natural arm movements and its workspace.

To achieve these goals and at the same time to reduce the complexity of the system (number of required DOF), it is necessary to keep the exoskeleton's joints near to the human arm, "ideally overlap them with the human articulations" in order to avoid parallel kinematic loops. It turns out that the necessity to have a compact, light, highly dynamic actuation system here is crucial. The advantages of using hydraulically actuators to operate the exoskeleton's joints, if directly compared with classical DC motors, are represented by their high force/weight ratio, the possibility to use the axes of the actuator as rotational/prismatic axes of the robotic system, and their back-drivability. Furthermore with a proper hydraulic supply and a precise fluid regulation, strength and high dynamic range can be achieved (Raibert et al., 2008), (Kazerooni et al., 2006),(Kahn, 1969).

In figure 1 is shown the actual version of our haptic interface. In total there are 7 actuated joints: 5 located in the shoulder/upper-arm and 2 in the forearm. An additional passive joint allows the wrist supination-pronation. All active joints are hydraulic actuated, valves, sensors and electronics are thought to be mounted directly in proximity of the actuators, this in order to reduce the amount of cables and pipes needed. The hydraulic pump and the primary power supply are located outside the exoskeleton to avoid additional weight to the system.

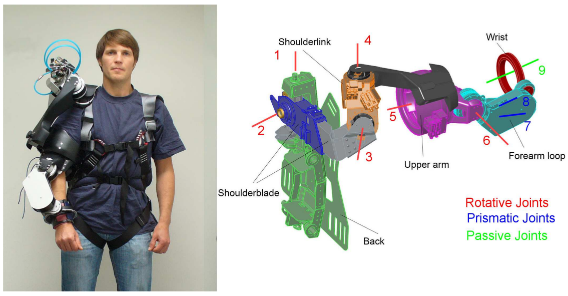


Figure 1: The arm exoskeleton equipped with 8 hydraulic rotative and linear actuators.

A central element within the hydraulic system is represented by the proportional servo-valve. On the market there are plenty of proportional (4/3) hydraulic valves, the problem is that most of them are though to work with high pressure and therefore do not fulfill our needs. According to the authors knowledge, the smallest, light weight and dynamically performing valve on the market, is currently sold by *MOOG Inc.* company (Inc., 2009). The device weights only $92g$, has an hysteresis for the flow characteristics $< 3\%$, and a 90^o phase-lag > 250Hz. Unfortunately the device is thought to work only with pressure in the range of $160-250 \cdot 10^5 Pa$ and it is specifically designed for Formula-1 applications.

Therefore we started to look at the pneumatic components, that generally are light and designed for low pressure (up to $10 \cdot 10^5 Pa$). We adapted them to work with liquids (oil), adding a precise actuation and proper sensory features.

This document is organized as follows: next section describes the experimental setup employed to measure the repeatability and flow-position characteristic of the valve, section 3 presents the dynamic model of the driving system, section 4 introduces a strategy to regulate position and velocity of the valve, section 5 presents a first fully integrated prototype. Finally section 6 draws the conclusions and future developments of this work.

2 EXPERIMENTAL SETUP AND TESTING

In this section the testbed developed to evaluate the performance of the servo valve and first experimental results characterising the valve are presented.
The experimental setup (figure 2) consists of the core parts of a commercial pneumatic valve (*Numatics Inc.* series Micro-Air), a stepper motor, a gear pump providing pressure supply between $0-30 \cdot 10^5 Pa$, a flow-meter, a pressure sensor, and an electronic board equipped with a STM32 μController.

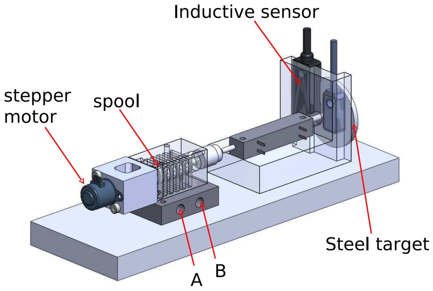


Figure 2: CAD Model of the valve testbed.

The drive system of the valve is a 3.3 V DC stepper motor from Nanotec working in fullstep mode, i.e. 18 degree/step and with an holding torque of $1.6 \cdot 10^{-3} Nm$. The rotor of the stepper motor is a lead screw, driving a cylinder and thus converting the rotational motor movement into a translation. The drive is attached to the valve spool via a permanent magnet, while the actual valve positions are determined using an inductive sensor from *Bahlluff Inc.* by tracking a steel target connected to the extended spool axis.
Control of the testbed and its components is performed by a STM32 μController (series F103VE), programmed with a special toolchain consisting of Matlab/Simulink and Rapidstm32 Blockset, Real-Time-Workshop and Keil Microvision μVision. The scheme in Figure 3 sketches the general dependencies of the testbed, were A and B are the connections of the valve to the actuator chambers, P_s and T are the pressure supply line respectively the tank lines of the gear pump, while the red lines represent the communication between the μController and the experimental setup through sensors and actors.

The two main features characterising a hydraulic valve are:

1. repeatability of the spool movement with respect to a certain input to the drive system.
2. flow through the valve with respect to spool position and the pressure drop over the valve.

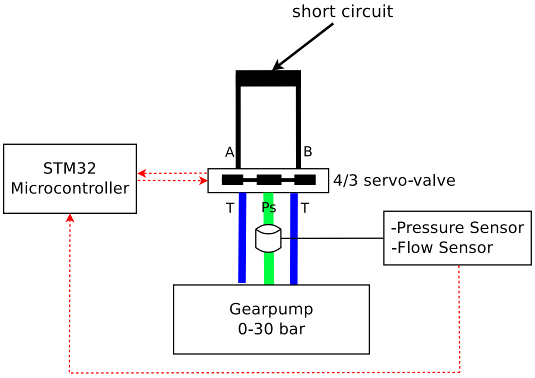


Figure 3: Scheme of experimental test setup.

Thus we focus on showing exemplary results of these two features. Position control of the fluidic valve is presented instead in section 4.

2.1 Performance of the Drive System

In the following, the ability of the valve's drive system to exactly position the spool over a long time intervall is investigated. Therefore the repeatability of valve movement is tested, by applying a special open loop control sequence. In the adjustment phase the spool is driven to the center position, while in the second phase the stepper motor is governed to move the spool 70 steps out of the zero position in both directions, which covers almost the whole working range of the developed prototype. Experiments took place under influence of pressure with $P_s = 15 \cdot 10^5 Pa$. Note that in this experiment the connectors A and B are connected in short circuit, while the speed of the motor is adjusted to its maximum of 1000 steps per second.

Figure 4 shows the open loop response of the valve prototype to the applied control sequence. The duration of the test was 20 minutes, while data was acquired for a time interval of 20 seconds every five minutes.

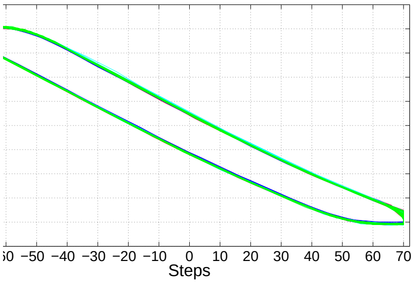


Figure 4: Position of spool while performing 70 steps out of center in both directions ($P_s = 15 \cdot 10^5 Pa$).

As we can obtain from Figure 4 the position response of the drive system to the reference signal is matching for the whole experiment. Differences for the travel of the spool, comparing the movement out of the center in both directions, might be caused by friction effects. The hysteresis is due to a backlash of approximately 10 steps (i.e. 0.1mm) between the spindle of the stepper motor and the thread of the cylinder moving the spool, when the motor changes its direction of rotation. This is due to a mechanical behavior and can only be reduced through higher precision in the manufacturing process of these two elements, or via a proper control action. Alternatively a special ball-screw could be used, which provides nearly zero backlash. Overall the valve shows remarkable repeatability for an open loop control of spool position under influence of pressure for a long time intervall.

2.2 Flow Characteristic of Valve

In this section the resulting static flow characteristic of the valve is presented.

To measure the flow for a fixed pressure drop over the valve, the connectors A and B are again linked in a short circuit, causing the valve to work against the pressure in the tank line T. A flowmeter from *Biotech* (series VZS-007-ALU) is connected to the pressure supply line providing a resolution of 900 pulses/L at a maximum flow of 5 L/min to the digital I/O of the μController. Due to the fact that flow needs a certain time to become constant, the valve is driven at a very low speed of 1 step/2s through the overall working period. Simultaneously the average flow is calculated and sampled each second.

Fig.5 shows the static flow characteristic of the hydraulic valve for a constant pressure drop of $\Delta P = 29 \cdot 10^5 Pa$.

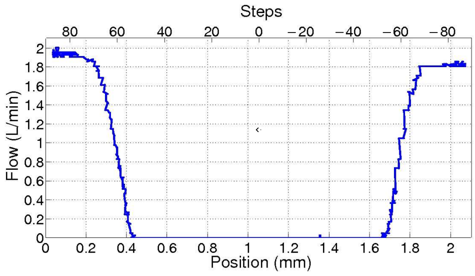


Figure 5: Flow-characteristic of the valve with respect to the spool position ($\Delta P = 29 \cdot 10^5 Pa$).

From Figure 5 we can determine that the valve has a large deadband of approximatively 1.25 mm (between 0.45 mm and 1.7 mm) where it is completely closed. This is caused by the inner structure of the valve, which is adapted from a pneumatic solenoid valve and can neither be influenced nor changed at the moment. Therefore this deadband has to be taken into account in the position control structure presented in section 4.

Having a closer look at the slopes of the flow characteristic, an area of about $0.2mm$ can be attested, where flow regulation should be possible. Driving the spool to the extremes causes flow saturation ($U_a \leq 0.2\,mm$ and $1.9\,mm \leq U_a$), thus the overall working range of the valve prototype can be defined to

$$0.15\,mm \leq U_a \leq 1.95\,mm,$$

whereas the amplitude of the flow depends on the pressure drop over the valve.

The exemplary results shown in this section certify a good repeatability to the drive system of the developed prototype. Furthermore the static flow characteristic promises a possible flow regulation within the defined working range.

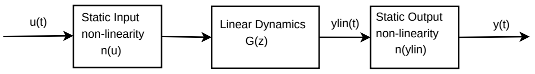
3 DYNAMIC MODEL

In this section a non-linear dynamic model that takes into account the electromechanical behavior of the stepper motor, the mechanical behavior of the spool and the static/dynamic effects of the frictions present within the system, was identified using a recursive Hammerstein-Wiener Parameter Adaptation Algorithm (PAA). The linear part of the model can be considered as an ARX structure, well known from the Recursive Least Squares(RLS) Algorithm described in (I. Landau, 2006). On the other hand, the input and output non-linearities with order nl of a Hammerstein-Wiener model have the form

$$\eta(u) = \sum_{k=2}^{n_l} \beta_k u^k \tag{1}$$

$$\eta(y_{lin}) = \sum_{k=2}^{n_l} \gamma_k y_{lin}^k \tag{2}$$

with y_{lin} as the output of the linear part.


u(t)
Static Input non-linearity n(u)
Linear Dynamics G(z)
ylin(t)
Static Output non-linearity n(ylin)
y(t)

(3)

Figure 6: Block diagram of a non-linear Hammerstein-Wiener model.

Using the equation for a Hammerstein-Wiener models output (Guo, 2004), the structure of the PAA becomes, with u(t) and y(t) being the real input and output signals of the system:

$$\hat{\theta}(t+1) = \theta(t) + F(t+1)\phi(t)\varepsilon^0(t+1) \tag{4}$$

with the Adaptation Gain

$$F(t+1) = F(t) - \frac{F(t)\phi(t)\phi(t)^T F(t)}{1+\phi(t)^T F(t)\phi(t)} \tag{5}$$

and the Prediction Error

$$\varepsilon^0(t+1) = y(t+1) - \hat{\theta}(t)^T \phi(t). \tag{6}$$

$\hat{\theta}$ is the vector of computed parameters, with

$$\hat{\theta}(t)^T = [\hat{a}(t)^T, \hat{b}(t)^T, \hat{\beta}(t)^T, \hat{\gamma}(t)^T], \tag{7}$$

where $\hat{a}(t)^T = [\hat{a}_1(t)...\hat{a}_{na}(t)]$ are the parameters of polynomial A with order na, $\hat{b}(t)^T = [\hat{b}_1(t)...\hat{b}_{nb}(t)]$ the parameters of polynomial B with order nb, $\hat{\beta}(t)^T = [\hat{\beta}_1(t)...\hat{\beta}_{nl}(t)]$ the parameters of the input non-linearity with order nl, $\hat{\gamma}(t)^T = [\hat{\gamma}_1(t)...\hat{\gamma}_{nl}(t)]$ the parameters of the output non-linearity with order nl.

Furthermore $\phi(t)$ is the Predictor Regressor Vector

$$\phi(t)^T = [-y(t), u(t), u(t)^2, ..., u(t)^{nl}, y_{lin}(t)^2, ..., y_{lin}(t)^{nl}]. \tag{8}$$

The output of the linear part y_{lin} cannot be measured, nevertheless it can be calculated by multiplying the parts of the predictor and the parameters vector corresponding to the linear model:

$$y_{lin}(t+1) = \hat{\theta}\,[1:na+nb+nl]\,(t)\phi\,[1:na+nb+nl] \tag{9}$$

Finally the model output $\hat{y}$ is computed

$$\hat{y}(t+1) = \hat{\theta}(t+1)^T \phi(t) \tag{10}$$

The measured signals used for identification are the current absorbed by the stepper motor i_M as input signal, and position of the valvespool x_S in terms of mm as output. In order to characterize the dynamic behavior of the valve we only considered the range of spool positions where the flow can be effectively regulated. As a first step the data was filtered using a bandpass filter allowing frequencies between $0Hz$ and $60Hz$, which does not affect the dynamic range of the model. Different models were identified starting from distinct initial values for the parameter vector, the best data fitting reached an average of 87.49%. The output of the obtained model can be seen in Figure 7, whereas the transfer function of the linear dynamic part is given by:

$$G(z) = \frac{-0.1675z - 0.1039}{z^3 - 0.8512z^2 - 0.1351z - 0.03855} \qquad (11)$$

with the sample time 0.01s. The Bode diagram of G(z) is shown in Figure 8. Finally the static input non-linearity of the model was identified to:

$$\eta(u) = 0.59831 * u^2 - 0.00047669 * u^3, \qquad (12)$$

while the static output non-linearity of the model is given by:

$$\eta(y_{lin}) = 0.030651 * y_{lin}^2 - 0.042948 * y_{lin}^3 \qquad (13)$$

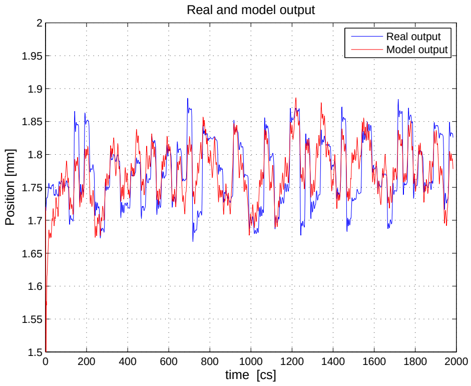


Figure 7: Comparison of the real and the model output.

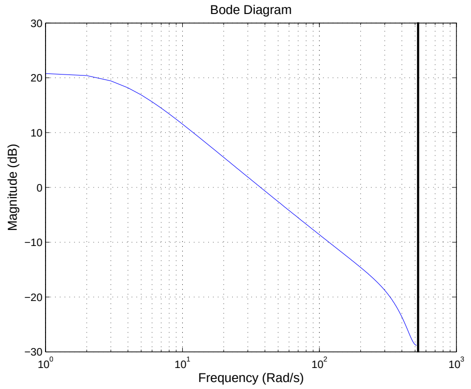


Figure 8: Magnitude Bode Plot of the linear transfer function.

4 VALVE CONTROL

This section introduces a first approach to control the developed fluidic valve.

Regarding the experiments shown in section 2, up to now all movements of the spool have been performed in open loop, i.e. no feedback of the spool position was used. To facilitate precise position control of the valve at a high bandwidth, the drive system is to be controlled in closed loop, using the feedback of the inductive sensor to track the desired trajectories given by the μController. Therefore classical PID-control in combination with a discretisation of stepper motor motion is applied.

A basic requirement for position control is the ability of the drive system to run at different speeds. Due to the fact that the stepper motor is somehow a digital drive, which can only have the states *run* or *stop*, a discretising function is introduced to the control loop. The idea is to vary the number of samples passing between two step commands by defining a variable delay factor τ_d, which causes a step command only every $\tau_d * T_s$, with T_s as basic sampling time of the system. Thus an increase of τ_d results in a slower motion of the valve spool, because less steps are performed by the drive system in a fixed time interval.

The value of τ_d is set by the control action u of the PID-controller, tracking the desired spool position p_d, as follows

$$\tau_d = \begin{cases} 15 - |u| \cdot 500 & , u \leq 0.03 \\ 1 & , u > 0.03 \end{cases} \qquad (14)$$

Figure 9 shows the resulting scheme of the closed loop control.

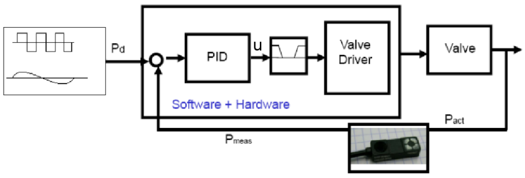


Figure 9: Scheme of discretised closed loop valve control.

Whereas p_{meas} represents the measured and p_d the desired spool position. The discretising function can be found between the PID-controller and the valve driver.

To reject noise from the position measurement, a moving average low-pass filter is realised via software in the μController. The following Figure 10 shows the modulation of the valve speed through the discretising function.

To verify the functionality of the control startegy, the valve is governed to execute a velocity sweep. Starting from $\tau_d = 100$ the delay factor is decreased by an amount of five every 5 ms until the maximum speed at $\tau_d = 1$ is reached. After changing the direction of movement τ_d is reset to its starting value. It is obvious that the speed of the valve is increased

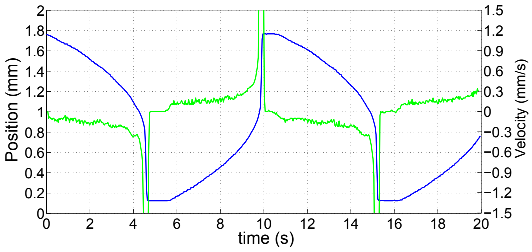


Figure 10: Speed modulated fluidic valve, blue: valve position, green: velocity.

while the delay factor decreases and vice versa.
To test the overall performances of the developed control scheme, sinusoidals with different frequencies and step-like reference trajectories were supplied. The PID's parameters have been determined through classical Ziegler-Nichols method, whereas the critical controller gain and the critical oscillation period were respectively identified as $K_{p,crit} = 0.58$ and $T_{crit} = 0.032\,s$. These results in the following controller gains

$$K_p = 0.35\,, K_i = 0.02\,, K_d = 0.004\,.$$

Figure 11 exemplary presents the tracking results of the position control loop for step-like reference signals.

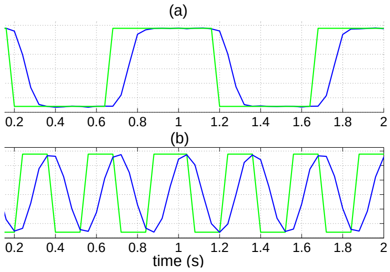


Figure 11: Step-like reference tracking at (a) f=1Hz, (b) f=3Hz, where the green line represents the reference.

As we can obtain from the figure, the controller tracks step-like signals without overshoot up to an actual maximum frequency of f=3Hz. Due to the fact that this test is performed in the maximum working range, a better dynamic response can be expected for smaller movements.
Finally Table 1 sums up the characteristic values for a sinusoidal reference tracking, where f is the frequency of the reference, $e_{p_{max}}$ the maximum error in position and φ the phase shift between the two signals.

Table 1: Chracteristic values for sinusoidal reference tracking.

f in [Hz]	$e_{p_{max}}$ in [V]	φ in [deg]
0.5	0.028	0.81
1	0.032	1.8
3	0.11	3.24

5 THE VALVE PROTOTYPE

After the choice of the proper hardware and electronic components, a new valve was designed and realized (figure 12) using rapid prototyping technique. The final device has a volume of LxWxH=(60mm)x(20mm)x(40mm), and the version with aluminum parts will weight approximately 80g including one position and two pressure sensors, an amplifier board, the stepper motor, and the electrical/hydraulic connectors.

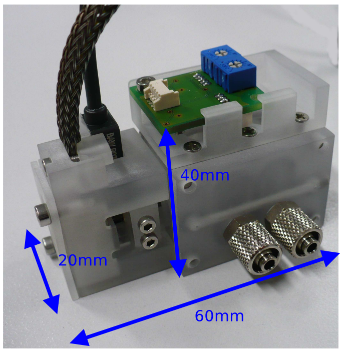


Figure 12: Valve developed using rapid prototyping technique.

In this first prototype the PWM motor driver and the control logic are located in a separated device, nevertheless future versions may also include these components on-board. This will reduce the amount of required cables to only a single power and a data line (e.g. a CAN-BUS). Compared with other state of the art hydraulic valves (Inc., 2009), the one here presented is designed to work with relatively low pressures both for hydraulic and pneumatics purposes, it is extremely compact and light-weight, furthermore it integrates two pressure sensors that are directly connected to the two output lines A and B (see schema in figure 2). This allows a fine tuning of the pressure inside the two actuator chambers, and therefore enables a precise control of the generated torque.

6 CONCLUSIONS AND FUTURE DEVELOPMENTS

In this paper a new fluid servo-valve specifically designed for wearable robotic systems is presented. The work is motivated by the fact that, according to the authors knowledge, no commercial valve exists for precise low pressure hydraulic actuators control. As general requirements, compactness, light-weight, and high dynamics were considered during the design process. A first series of experiments have been performed to test repeatability, flow-position characteristics and dynamic response. A model of the drive system of the servo-valve was identified using a recursive Hammerstein-Wiener parameter adaptation algorithm. The combination of a linear and dynamic part with a non linear static component let to reach a fit of 87%. Finally to test the overall functionality of the valve and to measure its step response characteristics, a proper control algorithm was implemented that allows to regulate the position and the velocity of the valve's spool.

Further work have to be dedicated in order to identify the overall model that will explicitly define the position-flow-pressure relationship. The backlash and deadband problems need to be properly addressed. In particular the employment of a ball-screw for the roto-translation mechanism of the valve, instead of a normal lead screw , will improve the precision in controlling the position. Furthermore, with a customized design of the spool, will be possible to decrease the switching time between the two opening positions, and therefore to improve the dynamic behavior of the servo-valve.

ACKNOWLEDGEMENTS

The work presented in this paper was done within the VI-Bot project, funded by the German Ministry of Science (BMBF), [grant number 01IW07003].

REFERENCES

Folgheraiter, M., Bongardt, B., Albiez, J., and Kirchner, F. (2008). A bio-inspired haptic interface for tele-robotics applications. In *IEEE International Conference on Robotics and Biomemetics (ROBIO-2008)*, Thailand, Bangkok.

Folgheraiter, M., Bongardt, B., Albiez, J., and Kirchner, F. (2009a). Design of a bio-inspired wearable exoskeleton for applications in robotics. In *International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC-2009)*, Portugal, Porto.

Folgheraiter, M., de Gea, J., Bongardt, B., Albiez, J., and Kirchner, F. (2009b). Bio-inspired control of an arm exoskeleton joint with active-compliant actuation system. *Applied Bionics and Biomechanics*, 6(2):193–204.

Folgheraiter, M., Schmidt, B. B. S., de Gea Fernandéz, Albiez, and Kirchner, F. (2009c). Design of an arm exoskeleton using an hybrid motion-capture and model-based technique. In *IEEE International Conference on Robotics and Automation. IEEE International Conference on Robotics and Automation (ICRA-2009), May 12-17, Kobe, Japan*.

Guo, F. (2004). *A New Identification Method for Wiener and Hammerstein System*. PhD thesis, Karlsruhe University.

Hayward, V. (1994). *Experimental Robotics III, Design of a hydraulic robot shoulder based on a combinatorial mechanism*, volume 200. Springer Berlin / Heidelberg.

I. Landau, G. Z. (2006). *Digital Control Systems*. Springer.

Inc., M. (2009). Sub miniature servovalve e024. Technical report, http://www.moog.com/.

Jacobsen, S. C., Smith, F. M., Backman, D. K., and Iversen, E. K. (1991). High performance, high dexterity, force reflective teleoperator ii. In *Proc. ANS Topical Meeting on Robotics and Remote Systems*.

Kahn, M. (1969). *Optimal Control of a Hydraulic Arm*. PhD thesis, Stanford University M.E Dept.

Kazerooni, H., Steger, R., and Huang, L. (2006). Hybrid control of the berkeley lower extremity exoskeleton (bleex). *The International Journal of Robotics Research*, 25(5-6):561–573.

Pons, J. L., editor (2008). *Wereable Robots: Biomechatronixs Exoskeletons*. Wiley.

Raibert, M., Blankespoor, K., Nelso, G., Playter, R., and the BigDog Team (2008). Bigdog, the rough-terrain quaduped robot. Technical report, Waltham, MA 02451 USA.

Yoshinada, H., Yamazaki, T., Suwa, T., and Naruse, T. (1992). Design and control of a manipulator system driven by seawater hydraulic actuator. In *Second Int. Symposium on Measurement and Control in Robotics (ISMCR)*, pages 359–364.

COLLECTIVE LEARNING OF CONCEPTS USING A ROBOT TEAM

Ana Cristina Palacios-García, Angélica Muñoz-Meléndez and Eduardo F. Morales
National Institute for Astrophysics, Optics and Electronics
Luis Enrique Erro No. 1, Santa María Tonantzintla, Puebla, Mexico
cpalacios@ccc.inaoep.mx, {munoz, emorales}@inaoep.mx

Keywords: Robotics and automation mobile robots and autonomous systems, Vision, Recognition and reconstruction, Network robotics.

Abstract: Autonomous learning of objects using visual information is important to robotics as it can be used for local and global localization problems, and for service tasks such as searching for objects in unknown places. In a robot team, the learning process can be distributed among robots to reduce training time and produce more accurate models. This paper introduces a new learning framework where individual representations of objects are learned on-line by a robot team while traversing an environment without prior knowledge on the number or nature of the objects to learn. Individual concepts are shared among robots to improve their own concepts, combining information from other robots that saw the same object, and to acquire a new representation of an object not seen by the robot. Since the robots do not know in advance how many objects they will encounter, they need to decide whether they are seeing a new object or a known object. Objects are characterized by local and global features and a Bayesian approach is used to combine them, and to recognize objects. We empirically evaluated our approach with a real world robot team with very promising results.

1 INTRODUCTION

The design of robot teams is a very active research domain in the mobile robotics community. Robot teams have effectively emerged as an alternative paradigm for the design and control of robotic systems because of the team's capability to exploit redundancy in sensing and actuation.

The research on robot teams has focused on developing mechanisms that enable autonomous robots to perform collective tasks, such as strategies for coordination and communication (Asada et al., 1994; Matarić, 1997); exploration, mapping and deployment (Howard et al., 2006); sensing, surveillance and monitoring (Parker, 2002); and decentralized decision making (Wessnitzer and Melhuish, 2003). In these works, a robot team can reduce time to complete a complex task that is allocated among its members.

Despite constant research on the design of robot teams, very little attention has been paid so far to the development of robot teams capable of learning from their interaction with their environment. In addition to their capability for accelerated learning, learning robot teams can be used to acquire a much richer and varied information compared to the information acquired by single learning robots.

Learning is a key issue to achieve autonomy for both, single robot and robot teams. Learning capabilities can provide robots flexibility and adaptation needed to cope with complex situations. In the context of robot teams, the most common machine learning approach has been reinforcement learning, where the idea is to learn optimal policies using a set of robots to improve the coordination of individual actions in order to reach common goals (Asada et al., 1994; Matarić, 1997; Parker, 2002; Fernández et al., 2005).

In this work we use visual information to learn, with a team of robots, descriptions of objects placed in a particular environment. Learning to recognize particular objects in an environment is important for robotics as it can be used for local and global localization tasks as well as for simple service tasks such as searching for objects in unknown places. Contrary to previous approaches, in our learning setting, the robots are not told the number or nature of the objects to be learned.

Vision is a primary source of perception in robotics and provides different features that can be used to classify objects. In general, using a particular set of features can be adequate for particular tasks but inadequate for other tasks. In this work, objects are characterized by two complementary features: (i) SIFT features (Lowe, 2004) and (ii) information about the silhouettes of objects. Other features could be used as well, but the main objective in this work is to show the different cases and possible confusions that can arise in the recognition of objects and merging of concepts, and how they can be addressed.

Numerous difficulties arise in robot teams when learning as well as sharing concepts that represent concrete objects. Some of these issues are discussed by Ye and Tostsos (1996) and include, how do robots represent their local views of the world, how is the local knowledge updated as a consequence of the robot's own action, how do robots represent the local views of other robots, and how do they organize the knowledge about themselves and about other robots such that new facts can be easily integrated into the representation. This article addresses the individual and collective representation of objects from visual information using a team of autonomous robots.

The rest of the paper is organized as follows. Section 2 reviews related work. Sections 3 y 4 describe, respectively, the stages of individual learning and collective learning of concepts. Section 5 describes our experimental results, and Section 6 provides conclusions and future research work.

2 RELATED WORK

Interesting experiments where physical mobile robots learn to recognize objects from visual information have been reported. First we review significant work developed for individual learning, and then we review learning approaches developed for robot teams.

Steels and Kaplan (2001) applied an instance-based method to train a robot for object recognition purposes. In this work objects are represented by color histograms. Once different representations have been learned from different views of the same object, the recognition is performed by classifying new views of objects using the KNN algorithm (Mitchell, 1997).

Ekvall et al. (2006) used different learning techniques to acquire automatically semantic and spatial information of the environment in a service robot scenario. In this work, a mobile robot autonomously navigates in a domestic environment, builds a map, localizes its position in the map, recognizes objects and locates them in the map. Background subtraction techniques are applied for foreground objects segmentation. Then objects are represented by SIFT points (Lowe, 2004) and an appearance-based method for detecting objects named Receptive Field Co-occurrence Histograms (Ekvall and Kragic, 2005). The authors developed a method for active object recognition which integrates both local and global information of objects.

In the work of Mitri et al. (2004), a scheme for fast color invariant ball detection in the RoboCup context is presented. To ensure the color-invariance of the input images, a preprocessing stage is first applied for detecting edges using the Sobel filter, and specific thresholds for color removal. Then, windows are extracted from images and predefined spatial features such as edges and lines are identified in these windows. These features serve as input to an AdaBoost learning procedure that constructs a cascade of classification and regression trees (CARTs). The system is capable of detecting different soccer balls in RoboCup and other environments. The resulting approach is reliable and fast enough to classify objects in real time.

Concerning the problem of collective learning of objects using robot teams there are, as far as we know, very few works. Montesano and Montano (2003) address the problem of mobile object recognition based on kinematic information. The basic idea is that if the same object is being tracked by two different robots, the trajectories and therefore the kinematic information observed by each robot must be compatible. Therefore, location and velocities of moving objects are the features used for object recognition instead of features such as color, texture, shape and size, more appropriate for static object recognition. Robots build maps containing the relative position of moving objects and their velocity at a given time. A Bayesian approach is then applied to relate the multiple views of an object acquired by the robots.

In the work of O'Beirne and Schukat (2004), objects are represented with Principal Components (PC) learned from a set of global features extracted from images of objects. An object is first segmented and its global features such as color, texture, and shape are then extracted. Successive images in a sequence are related to the same object by applying a Kalman filter. Finally, a 3D reconstructed model of an object is obtained from the multiple views acquired by robots. For that purpose, a Shape From Silhouette based technique (Cheung et al., 2003) is applied.

In contrast to previous works, in our method each member of the robot team learns on-line individual representations of objects without prior knowledge on the number or nature of the objects to learn. Individual concepts are represented as a combination of global and local features extracted autonomously by the robots from the training objects. A Bayesian approach is used to combine these features and used for classification. Individual concepts are shared among robots to improve their own concepts, combining information from other robots that saw the same object, and to acquire a new representation of an unnoticed object.

3 INDIVIDUAL LEARNING OF CONCEPTS

The individual concepts are learned on-line by a robot team while traversing an environment without prior knowledge on the number or nature of the objects to learn. The individual learning of concepts consists of tree parts: object detection, feature extraction and individual training.

Individual concepts of objects are represented by Principal Component (PC) over the information about the silhouettes of objects and Scale Invariant Features (SIFT). Learned concepts are shared among robots.

3.1 Object Detection

Robots move through an environment and learn descriptions of objects that they encountered during navigation. Objects are detected using background substraction. In this paper we assume a uniform and static background. We performed morphological operations (closing - erode) to achieve better segmentation. Once an object is detected, it is segmented and scaled to a fixed size, to make the global PC features robust to changes in scale and position.

3.2 Feature Extraction and Individual Training

The segmented objects are grouped autonomously by the robots in sets of images containing the same object. Robots assume that they are observing to the same object while it can be detected, and they finish to see it when they can not detect objects in the captured images. Only one object can be detected in an image at the same time. For each set of images, the robot obtains an individual concept that represents the object.

Training using Global Features. We applied Principal Component Analysis (PCA) over the average silhouettes that are automatically extracted from the set of images of a particular object. The average provides a more compact representation of objects and reduces segmentation errors. Figure 1 (a) shows an object used in the training phase, Figure 1 (b) shows its silhouette, and Figure 1 (c) illustrates the average silhouette obtained from a set of images that represent the object of Figure 1 (a). Once the robot has obtained an average silhouette, this is added by the robot to a set of known average silhouettes. After that, the robot uses PCA to reduce the dimensionality of all average silhouettes learned to get the PC features that represent them.

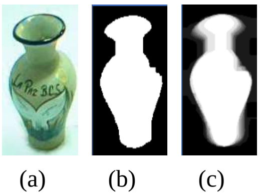


Figure 1: Examples of the silhouette (b) and average silhouette (c) of an object (a).

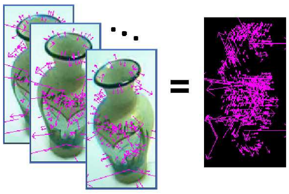


Figure 2: Examples of the SIFT features extracted from a set of images and the final set of SIFT features.

Training using Local Features. Each robot extracts local SIFT features of each image of the set of images, and groups them in a final set which contains all the different SIFT features that represent an object. In Figure 2 we show an example of the SIFT points obtained from a set of images of a *vase* and the final set of SIFT points obtained. The PC features and the SIFT features represent the individual concept of the observed object.

3.3 Sharing Concepts

The concepts learned by robots are shared among them to achieve collective learning. This can be done off-line or on-line. In the case of collective off-line learning the robots share their individual concepts once they have learned all the training objects. On the other hand, in the collective on-line learning the robots share their individual concept as soon as a new object is learned.

4 COLLECTIVE LEARNING OF CONCEPTS

Collective learning of concepts enables robots to improve individual concepts combining information from other robots that saw the same object, and to acquire a new representation of an object not seen by the robot. Therefore, a robot can learn to recognize more objects of what it saw and can improve their own concepts with additional evidence from other robots.

A robot has to decide whether the concept shared by another robot is of a new object or of a previously learned concept. A robot can face three possibilities: coincident, complementary or confused information. The shared concepts are fused depending on the kind of information detected, as described below.

4.1 Pre-analysis of Individual Concepts

The concept learned by a robot is defined as follows:

$$C_k^i = \{Sil_k^i, SIFT_k^i\} \tag{1}$$

where C_k^i is the concept k learned by robot i, Sil_k^i is the average silhouette, and $SIFT_k^i$ is the set of SIFT features that form the concept k.

In order to determine if a shared concept is previously known or not to a robot, it evaluates the probabilities that the PC features and SIFT features are previously known by the robot. The probability vectors of PC features calculated by robot i, v_P^i, indicate the probability that a concept shared by robot j, C_k^j, is similar to the concepts known by robot i, $C_1^i, \ldots, C_{numObjs}^i$, given the global features. *numObjs* is the number of concepts of objects known by robot i. The process to obtain the probability vector PC is described as follows:

- A temporal training set of silhouettes is formed by adding the average silhouettes of concepts known by robot i or actual robot, $Sil_1^i, ..., Sil_{numObjs}^i$, and the average silhouette of the shared concept Sil_k^j.
- The PCA is trained using the temporal set of average silhouettes. The projection of the average silhouettes know by robot i is obtained as a matrix of projections, *matProys*. The projection of the average silhouette Sil_k^j is obtained in a vector, *vectProys*.
- The Euclidean distance (dE) is calculated between each vector of the matrix *matProys* and the vector *vectProys* as shown in formula 2, i.e, we obtain the distance between all the projections already computed and the projection of the new silhouette.

$$dE_l^i = \sqrt{\sum_{r=1}^{nEigens} (matProys_{(l,r)} - vectProys_{(1,r)})^2} \tag{2}$$

where $nEigens$ is the number of eigenvectors used during the PCA training ($nEigens = numObjs^i - 1$), and l is the index of the distance vector, where the maximum size of the vector dE^i is $numObjs^i$.

- The distance value dE^i is divided by a maximum distance value, $ThresholdMax$, determined experimentally to obtain a similarity metric also called the probability vector v_P^i as shown in formula 3.

$$v_{P_l}^i = 1 - \frac{dE_l^i}{ThresholdMax} \tag{3}$$

If dE_l^i is bigger than the $ThresholdMax$ value, then the probability will be fixed as shown in formula 4, which indicates that the projections of the object j and the one of the object i are completely different.

$$v_{P_l}^i = \frac{1}{numObjs} \tag{4}$$

The value of the SIFT similarity metric also called the probability vector SIFT at the position $v_{S_l}^i$, is obtained calculating the number of coincident SIFT, n_{coin}, between the individual SIFT concept $SIFT_l^i$ learned by robot i, and the individual SIFT concept $SIFT_k^j$ shared by robot j. If the number n_{coin} is bigger than an average of coincidences determined experimentally, *AverageCoin*, then the probability will be fixed to $v_{S_l}^i = 1.0$, which means that both concepts contain the same local features SIFT. In other case, the probability will be calculated using formula 5.

$$v_{S_l}^i = \frac{n_{coin}}{AverageCoin} \tag{5}$$

The constant *AverageCoin* represents the average of coincidences between two sets of SIFT points of the same object from different perspectives.

4.2 Analysis and Fusion of Individual Concepts

This section describes how to detect if the shared concept is coincident, complementary or confused, and how the individual concepts are fused to form collective concepts depending on the kind of detected concept.

4.2.1 Coincident Concepts

A coincident concept is detected when two or more robots of the robot team learned individual concepts from similar views of the same object. A shared concept is classified as coincident if $v_{P_l}^i \geq \alpha$ and $v_{S_l}^i \geq \alpha$. That is, if both probabilities (PC and SIFT) of a previously learned concept are greater than a predefined threshold value (α). If a shared concept is determined as coincident it is merged with the most similar known concept as follows:

PCA Fusion. It is obtained by evaluating a new average silhouette from the average of the known Sil_l^i and new Sil_k^j silhouettes. After that, it is necessary to re-train the PCA substituting the concept Sil_l^i with the new average silhouette which contains information of the concept learned by robot j.

SIFT Fusion. It is obtained by adding the complementary SIFT points of concept $SIFT_k^j$ to the set of SIFT points of concept $SIFT_l^i$. Also, each pair of coincident SIFT points of both concepts is averaged in terms of position and their corresponding SIFT descriptors.

The main idea to fuse individual concepts is to improve their representation.

4.2.2 Complementary Concepts

A concept C_k^j contains complementary information if it differs with all known concepts by robot i, i.e., if both shape and local features are different to all known concepts by robot i, $C_1^i, \ldots, C_{numObjs}^i$. That is, if $v_P^i < \alpha$ and $v_S^i < \alpha$.

A complementary concept C_k^j is fused with the collective concepts known by the robot i as follows:

PCA Fusion. The new average silhouette is added and the new PC concepts are obtained by re-training the PCA using the updated set of average silhouettes.

SIFT Fusion. The new SIFT features are simple added to the current set of SIFT concepts known by the robot i.

4.2.3 Confused Concepts

There are two types of confusion that can occur between concepts:

Different Shape and Similar Local Features (type 1). This type of confusion occurs when the new concept C_k^j is complementary by shape, Sil_k^j, to all the concepts known by the robot i, $Sil_1^i, ..., Sil_{numObjs^i}^i$ but it is coincident by local SIFT features, $SIFT_k^j$, with at least one concept known by the robot i. That is, $v_{S_l}^i \geq \alpha$ and if $v_P^i < \alpha$.

Similar Shape and Different Local Features (type 2). This type of information occurs when concept C_k^j is coincident by shape, Sil_k^j, to at least one concept known by the robot i, but it is complementary using its local SIFT features, $SIFT_k^j$. That is, if $v_{P_l}^i \geq \alpha$ and $v_S^i < \alpha$.

In both types of confusion, type 1 or type 2, there can be two options:

a) **Different Objects.** Both concepts correspond to different objects.

b) **Same Object.** Both concepts correspond to the same object but they were learned by robots from different points of view.

In our current approach, both types of confusions are solved as complementary objects. The reason is that robot i cannot distinguish with its current information between both, different objects or same object, using only the individual and the shared concepts. To solve the ambiguity, as future work each robot should build autonomously a map and locate its position in the map. In addition, for each learned object, robots will locate them in the map. For confused objects a robot can move to the position of the object marked in the map to see the object from different perspectives in order to solve the conflict.

5 EXPERIMENTS AND RESULTS

We performed several experiments to demonstrate the proposed algorithm. In section 5.1, we show the results of a general experiment that demonstrates the main features of the proposed approach. In section 5.2 we present the accuracy of the collective concepts versus the individual concepts.

In these experiments we used a robot team consisting of two homogeneous Koala robots equipped with a video camera of 320 × 240 pixels. For more than two robots our method can be applied straightforward. The only difference is that robots will need to consider the information from more than one robot, possibly reducing confused concepts.

5.1 Concept Acquisition and Testing

The mobile robots learn on-line a representation of several objects while following a predefined trajectory without prior knowledge on the number or nature of the objects to learn. The idea of using pre-planned trajectories instead of making the robots wandering randomly, is that we can control the experimental conditions to show different aspects of the proposed methodology.

Each robot shares its individual concept as soon as it is learned to improve the representation of this concept or to include a new concept in the other robot.

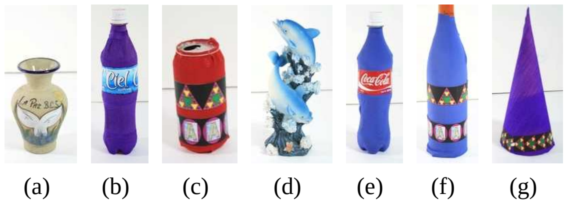


Figure 3: Training objects. a) *vase*, b) *water bottle*, c) *can*, d) *dolphin*, e) *soda bottle*, f) *bottle* and g) *cone*.

Figure 3 shows the training objects used in this experiment. As can be seen in the figure, some objects have the same shape but different texture, some have the same texture but different shape, some others are not symmetric in their shape. The objective of this experiment is to show the performance of the system to detect coincident, complementary and confused information under a wide variety of conditions.

Robot 1 (R1) learned during individual training concepts for: *dolphin*, *can*, *water bottle* and *vase*. Robot 2 (R2) learned individual concepts for: *vase*, *soda bottle*, *bottle* and *cone*. Note that some objects are learned by both robots while others are only learned by one robot.

While learning a new concept, each robot has to decide whether to fuse the current concept with a previously known concept or include it as a new one. Tables 1 and 2 show the probability vectors of the PC features based on shape (v_P^1) and of the SIFT features (v_S^1) obtained by Robot 1. Tables 3 and 4 show the probability vectors of the PC (v_P^2) and SIFT (v_S^2) features obtained by Robot 2. In these tables the coincident information is represented in bold.

Table 1: Probability vectors PC (v_P^1) obtained by R1.

	New (collective concepts R1)						
Objects	*Dolphin*	*Vase*	*Can*	*Soda bottle*	*Water bottle*	*Bottle*	*Cone*
Dolphin$_{R1}$	-	-	-	-	-	-	-
Vase$_{R2}$	0.19	-	-	-	-	-	-
Can$_{R1}$	0.31	0.26	-	-	-	-	-
Soda bottle$_{R2}$	0.36	0.28	0.58	-	-	-	-
Water bottle$_{R1}$	0.43	0.28	0.53	**0.73**	-	-	-
Bottle$_{R2}$	0.31	0.17	0,56	0.61	0.58	-	-
Vase$_{R1}$	0.25	**0.69**	0.42	0.43	0.41	0.32	-
Cone$_{R2}$	0.31	0.01	0.28	0.28	0.33	0.43	-

We used the defined criteria in Section 4.2 to recognize coincident, complement or confused concepts, with α = 0.65 as threshold value, and the probability vectors of Tables 1, 2, 3 and 4.

Table 2: Probability vectors SIFT (v_S^1) obtained by R1.

	New (collective concepts R1)						
Objects	*Dolphin*	*Vase*	*Can*	*Soda bottle*	*Water bottle*	*Bottle*	*Cone*
Dolphin$_{R1}$	-	-	-	-	-	-	-
Vase$_{R2}$	0.09	-	-	-	-	-	-
Can$_{R1}$	0.12	0.12	-	-	-	-	-
Soda Bottle$_{R2}$	0.28	0.11	0.40	-	-	-	-
Water bottle$_{R1}$	0.15	0.59	0.20	0.20	-	-	-
Bottle$_{R2}$	0.08	0.15	**0.65**	0.04	0.12	-	-
Vase$_{R1}$	0.16	**1.00**	0.23	0.10	0.08	0.09	-
Cone$_{R2}$	0.05	0.28	0.43	0.10	0.14	0.09	-

Table 3: Probability vectors PC (v_P^2) obtained by R2.

	New (collective concepts R2)						
Objects	*Vase*	*Dolphin*	*Soda bottle*	*Can*	*Bottle*	*Water bottle*	*Cone*
Vase$_{R2}$	-	-	-	-	-	-	-
Dolphin$_{R1}$	0.19	-	-	-	-	-	-
Soda bottle$_{R2}$	0.28	0.36	-	-	-	-	-
Can$_{R1}$	0.26	0.31	0.58	-	-	-	-
Bottle$_{R2}$	0.17	0.31	0.61	0.56	-	-	-
Water bottle$_{R1}$	0.29	0.44	**0.73**	0.54	0.58	-	-
Cone$_{R2}$	0.01	0.31	0.28	0.28	0.43	0.33	-
Vase$_{R1}$	**0.69**	0.25	0.43	0.42	0.32	0.41	0.01

Tables 5 and 6 show the results of the analysis performed by each robot. As can be seen from these tables, each robot encountered the three types of possible information and fuse its concepts accordingly.

For instance, Table 1 shows how are the probabilities of objects of R1 affected using only PCA over shapes of objects, as both robots encounter and learn concepts while traversing the environment. In the first row, R1 learns about the concept *dolphin* and acquires it. In the second row, R2 then learns about *vase* and shares this concept to R1. The probability, according to the PCA features to be a *dolphin* is 0.19 (second row). R1 learns the object *can*, which has a probability of 0.31 to be a *dolphin* and a probability of 0.26 to be a *vase*, which was learned by R2 and shared to R1 (third row). In the fifth row, R1 learns about a *water bottle* but it confuses with the *soda bottle* learned

Table 4: Probability vectors SIFT (v^2_S) obtained by R2.

	New (collective concepts R2)						
Objects	*Vase*	*Dolphin*	*Soda bottle*	*Can*	*Bottle*	*Water bottle*	*Cone*
Vase$_{R2}$	-	-	-	-	-	-	-
Dolphin$_{R1}$	0.18	-	-	-	-	-	-
Soda bottle$_{R2}$	0.11	0.28	-	-	-	-	-
Can$_{R1}$	0.12	0.12	0.04	-	-	-	-
Bottle$_{R2}$	0.15	0.08	0,04	0.64	-	-	-
Water bottle$_{R1}$	0.59	0.15	0.20	0.20	0.12	-	-
Cone$_{R2}$	0.09	0.05	0.10	0.43	0.09	0.14	-
Vase$_{R1}$	**1.00**	0.16	0.10	0.23	0.09	0.08	0.12

Table 5: Detected information by R1 for each own and shared individual concepts.

Individual concepts	Related objects	v^1_P	v^1_S	Kind of info.
Dolphin$_{R1}$	-	-	-	Complementary
Vase$_{R2}$	All l ($l = 1$ to $numObj^1$)	$v^1_{P_{(2,l)}} < 0.65$	$v^1_{S_{(2,l)}} < 0.65$	Complementary
Can$_{R1}$	All l	$v^1_{P_{(3,l)}} < 0.65$	$v^1_{S_{(3,l)}} < 0.65$	Complementary
Soda bottle$_{R2}$	All l	$v^1_{P_{(4,l)}} < 0.65$	$v^1_{S_{(4,l)}} < 0.65$	Complementary
Water bottle$_{R1}$	*Soda bottle*	$v^1_{P_{(5,4)}} = 0.73$	$v^1_{S_{(5,i)}} < 0.65$	Confuse type 2
Bottle$_{R2}$	*Can*	$v^1_{P_{(6,l)}} < 0.65$	$v^1_{S_{(6,3)}} = 0.65$	Confuse type 1
Vase$_{R1}$	*Vase*	$v^1_{P_{(7,2)}} = 0.69$	$v^1_{S_{(7,2)}} = 1.00$	Coincident
Cone$_{R2}$	All l	$v^1_{P_{(8,l)}} < 0.65$	$v^1_{S_{(8,l)}} < 0.65$	Complementary

and shared before by R2. As can be seen from Figure 3, both objects have the same shape and consequently the PCA features are not able to discriminate between these two objects. This is not the case for the SIFT features, which prevent R1 to consider it as the same object (as explained below). In the seventh row, R1 learns about *vase* which was already learned and shared by R2, and in this case both concepts are merged.

Table 6: Detected information by R2 for each own and shared individual concepts.

Individual concepts	Related objects	v^2_P	v^2_S	Kind of info.
Vase$_{R2}$	-	-	-	Complementary
Dolphin$_{R1}$	All k	$v^2_{P_{(2,k)}} < 0.65$	$v^2_{S_{(2,k)}} < 0.65$	Complementary
Soda bottle$_{R2}$	All k	$v^2_{P_{(3,k)}} < 0.65$	$v^2_{S_{(3,k)}} < 0.65$	Complementary
Can$_{R1}$	All k	$v^2_{P_{(4,k)}} < 0.65$	$v^2_{S_{(4,k)}} < 0.65$	Complementary
Bottle$_{R2}$	All k	$v^2_{P_{(5,k)}} < 0.65$	$v^2_{S_{(5,k)}} < 0.65$	Confuse type 1
Water bottle$_{R1}$	*Soda Bottle*	$v^2_{P_{(6,3)}} = 0.73$	$v^2_{S_{(6,k)}} < 0.65$	Confuse type 2
Cone$_{R2}$	All k	$v^2_{P_{(7,k)}} < 0.65$	$v^2_{S_{(7,k)}} < 0.65$	Complementary
Vase$_{R1}$	*Vase*	$v^2_{P_{(8,1)}} = 0.69$	$v^2_{S_{(8,1)}} = 1.00$	Coincident

To test the performance of the individual concepts and the collective concepts acquired by each robot, the concepts were used in an object recognition task. Each robot followed a predefined trajectory to recognize objects in the environment. The objects were detected by the robot team in the following order: *cone*, *water bottle*, *vase*, *bottle*, *soda bottle* and *dolphin*. Once an object is detected, the robot (i) evaluates its class using the PC (v^i_P) and SIFT (v^i_S) probability vectors and combines both probabilities using a Bayesian approach:

$$P^i_{B_l} = \frac{v^i_{P_l} \times v^i_{S_l} \times P_u}{\left(v^i_{P_l} \times v^i_{S_l} \times P_u\right) + \left((1 - v^i_{P_l}) \times (1 - v^i_{S_l}) \times (1 - P_u)\right)} \quad (6)$$

where P_u is a uniform probability distribution ($P_u = \frac{1}{numObjs^i}$), $v^i_P = p(PC\ projection \mid Class = i)$, $v^i_S = p(SIFT\ matching \mid Class = i)$, P^i_B is the Bayesian probability vector ($p(Class = i \mid PCprojection, SIFTmatching)$, and l is the index of the Bayesian probability vector, where the maximum size of the probability vector is $numObjs^i$.

Figures 4, 5 and 6 show the average probabilities obtained during the object recognition task for each set of images of the same class, using the individual and collective learned concepts. The dotted bars indicate the classification errors. A classification error is produced when a robot classifies an unknown object with a probability ≥ 0.6. The unknown objects for robots 1 and 2 are those which were not learned during their individual training.

The classification errors of Robot 1 in Figure 4 occur when the objects *cone*, *bottle* and *soda bottle* are classified as *dolphin*, *water bottle* and *water bottle*, respectively. The classification errors of Robot 2

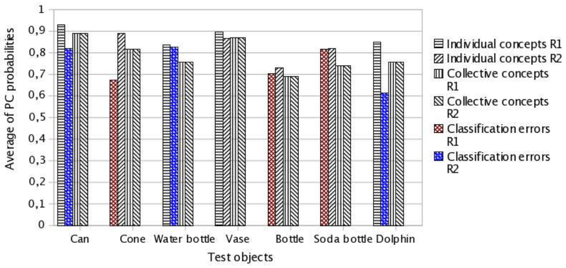


Figure 4: Average PC classification probabilities for the object recognition task using the individual and collective PC concepts.

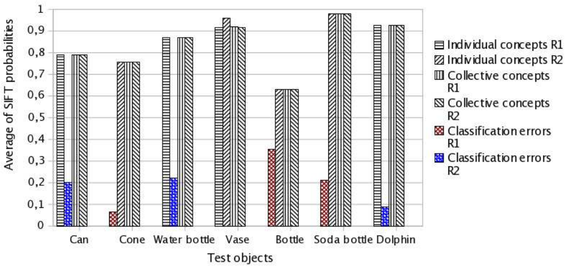


Figure 5: Average SIFT classification probabilities for the object recognition task using the individual and collective SIFT concepts. Any robot makes classification errors.

occur when the objects *can*, *water bottle* and *dolphin* are classified as *bottle*, *soda bottle* and *soda bottle*, respectively. For the *vase* there is no classification error because both robots learn individual concepts of it.

Although the probability bars presented in the previous figures show a higher probability for individual concepts than for collective concepts, in reality the collective concepts are more robust as they represent the probabilities considering a larger number of objects. This will be discussed in Section 5.2.

Table 7: Precision in the object recognition task using the individual and collective concepts acquired by each robot.

	R1	R2	R1-R2	R2-R1
PCA	55.69 % (100.00 %)	49.98 % (94.82 %)	86.15 %	86.16 %
SIFT	48.32 % (86.23 %)	42.89 % (79.11 %)	87.84 %	87.84 %
Bayes	52.59 % (94.20 %)	51.68 % (81.54 %)	80.73 %	80.73 %

We show in Table 7 the precision of the object recognition task using the individual and collective concepts. The precision is presented in two ways, one considering the total number of objects, and the other one taking into a count only the number of objects used during the individual training (reported in parentheses). As can be seen the collective concepts produce a significantly better precision.

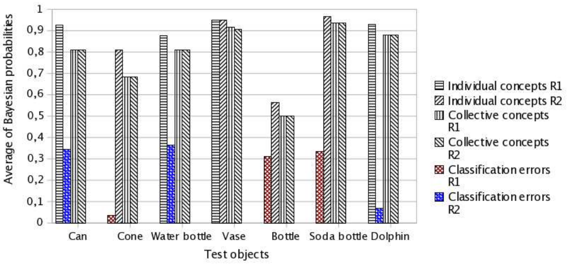


Figure 6: Bayesian classification probabilities which uses the Bayesian fusion of the PCA and SIFT classification probabilities.

5.2 Accuracy of the Individual and Collective Concepts

In this section we compare the results of the individual concepts with that of collective concepts. In each experiment, a different set of objects was used, and both robots learned the same set of objects. Therefore, all the shared concepts were coincident, that is, robots learned both individually and collectively the same number of concepts. At the end of each experiment the robots learned four concepts that were proved by a test sequence. In Figure 7 we present the accuracy obtained by the robots using the PC features of the individual and collective concepts in an object recognition task. Figure 7 shows the averages in accuracy of the number of images well classified under six experiments (*correct*), the average percentages of the number of non detected or non classified images (*no detected*), and the average percentage of false positives for each concept (*false* +). Figure 8 and 9 show, respectively, the accuracy obtained by the robots when using the SIFT vectors and the Bayesian approach.

As it can be observed in Figures 7, 8 and 9, the accuracy that indicate the quantity of well classified images (*correct*) using the collective concepts for the object recognition task, is in general better than the accuracy using the individual concepts. For PC, SIFT and Bayes there is an improvement in the accuracy up to 2.56 %, 13.79 % and **20.62 %**, respectively. This demonstrates that the collective concepts have better coverage than the individual concepts because they contain information acquired from different points of view, which allows a better recognition of test objects. Also, the percentages of the number of non detected images of collective concepts are smaller than the ones of the individual concepts.

In Table 8 we present the average percentages of false positives for both, the individual and the collective concepts acquired by the robots. We conclude that the collective concepts have better quality than the individual concepts.

In general for both, the individual and the collective concepts, we observed an improvement in the accuracy when using the Bayesian approach. In Table 9 we present the average percentages of accuracy using the individual and collective concepts.

The average profit in the percentages of classification using the Bayesian approach using the collective concepts with regard to the individual concepts is of **14.63 %**.

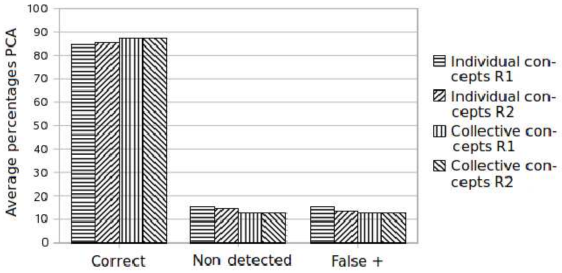


Figure 7: Accuracy in coverage using the part PC of concepts.

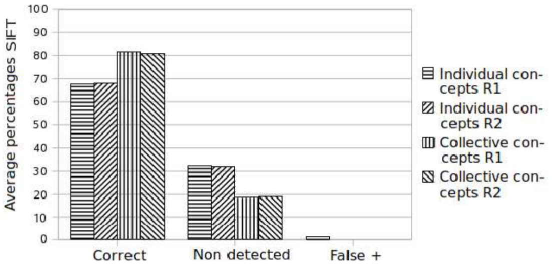


Figure 8: Accuracy in coverage using the part SIFT of concepts.

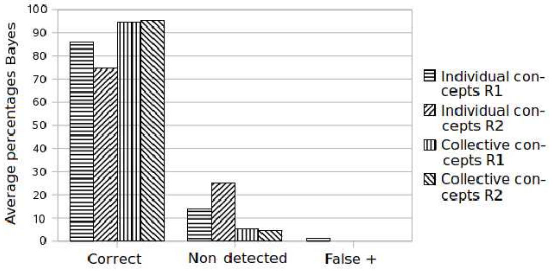


Figure 9: Accuracy in coverage using the Bayesian approach.

Table 8: Average percentages of false positives.

	PCA	SIFT	Bayes
Individual	14.42 %	0.64 %	0.64 %
Collective	13.14 %	0.00 %	0.00 %

Table 9: Average percentages of accuracy.

	PCA	SIFT	Bayes
Individual	84.94 %	67.88 %	80.18 %
Collective	87.18 %	81.12 %	94.81 %

6 CONCLUSIONS AND FUTURE WORK

In this paper we have introduced a new on-line learning framework for a team of robots. Some of the main features of the proposed scheme are:

- The robots do not know in advance how many objects they will encountered. This pose several problems as the robots need to decide if a new seen object or shared concept, is of a previously learned concept or not.
- The representation of objects are learned on-line while the robots are traversing a particular environment. This is relevant for constructing autonomous robots.
- Three possible cases in which to merge concepts and how to merge them were identified.

The detection of coincident concepts avoids producing multiple concepts for the same object. The detection of complementary concepts allows to detect and learned unknown objects not seen by a particular robot. The detection of confused concepts allows to fuse information: 1) when the object have different shape and similar SIFT features, and 2) when the objects have similar shape and different SIFT features. These cases are particularly difficult to deal with because the objects may be genuinely different or may be the same but seen from different points of view by the robots.

In general, the object recognition using the collective concepts had a better performance than using the individual concepts in terms of accuracy. This occurs because the collective concepts consider information from multiple points of view producing more general concepts.

As future work we propose to integrate schemes to object segmentation for dynamic environments. For instance, using an object segmentation based on distance as in Méndez-Polanco et al., 2009. Use a different set of features and identify possible conflicts between more that two kind of features. We also plan to incorporate planning of trajectories to autonomously allocate the environment among robots. We also plan to add strategies to solve some confusions in shared concepts by taking different views from these objects. Finally, we plan to incorporate our algorithm for robot localization and search of objects, and to test our work for robot teams with three or more robots.

ACKNOWLEDGEMENTS

The first author was supported by the Mexican National Council for Science and Technology, CONACYT, under the grant number 212422.

REFERENCES

Asada, M., Uchibe, E., Noda, S., Tawaratsumida, S., and Hosoda, K. (1994). Coordination of multiple behaviors acquired by vision-based reinforcement learning. *Proceedings of the International Conference on Intelligent Robots and Systems*, pages 917–924.

Cheung, G. K. M., Baker, S., and Kanade, T. (2003). Visual hull alignment and refinement across time: A 3-D reconstruction algorithm combining Shape-From-Silhouette with stereo. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '03)*, 2:375–382.

Ekvall, S., Jensfelt, P., and Kragic, D. (2006). Integrating active mobile robot object recognition and SLAM in natural environments. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '06)*, pages 5792–5797.

Ekvall, S. and Kragic, D. (2005). Receptive field cooccurrence histograms for object detection. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '05)*, pages 84–89.

Fernández, F., Borrajo, D., and Parker, L. E. (2005). A reinforcement learning algorithm in cooperative multi-robot domains. *Journal of Intelligent and Robotic Systems*, pages 161–174.

Howard, A., Parker, L. E., and Sukhatme, G. S. (2006). Experiments with a large heterogeneous mobile robot team: Exploration, mapping, deployment and detection. *International Journal of Robotics Research*, 25:431–447.

Lowe, D. G. (2004). Distinctive image features from scale invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110.

Matarić, M. J. (1997). Reinforcement learning in the multi-robot domain. *Autonomous Robots*, 4(1):73–83.

Méndez-Polanco, J. A., Muñoz-Meléndez, A., and Morales, E. F. (2009). People detection by a mobile robot using stereo vision in dynamic indoor environments. *Proceedings of the 8th Mexican International Conference on Artificial Intelligence (MICAI '09)*, pages 349–359.

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill Science/Enginering/Math.

Mitri, S., Pervolz, K., Surmann, H., and Nuchter, A. (2004). Fast color independent ball detection for mobile robots. *Proceedings of the IEEE International Conference Mechatronics and Robotics (MechRob '04)*, pages 900–905.

Montesano, L. and Montano, L. (2003). Identification of moving objects by a team of robots based on kinematic information. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '03)*, 1:284–290.

O'Beirne, D. and Schukat, M. (2004). Exploration and object recognition using cooperative robots. *Proceedings of the International Conference on Imaging Science, Systems and Technology (CISST '04)*, pages 592–598.

Parker, L. E. (2002). Distributed algorithms for multi-robot observation of multiple moving targets. *Autonomous Robots*, 3(12):231–255.

Steels, L. and Kaplan, F. (2001). AIBO's first words. The social learning of languaje and meaning. *Evolution of Comunication*, 4(1):3–32.

Wessnitzer, J. and Melhuish, C. (2003). Collective decision-making and behaviour transitions in distributed ad hoc wireless networks of mobile robots: Target-hunting. *Advances in Artificial Life, 7th European Conference (ECAL '03)*, pages 893–902.

Ye, Y. and Tsotsos, J. K. (1996). On the collaborative object search team: a formulation. *Distributed Artificial Intelligence Meets Machine Learning (ECAI '96 Workshop)*, pages 94–116.

DYNAMIC MODELING AND PNEUMATIC SWITCHING CONTROL OF A SUBMERSIBLE DROGUE

Y. Han, R. A. de Callafon, J. Cortés
Department of Mechanical and Aerospace Engineering, UCSD, 9500 Gilman Drive, La Jolla, CA 92093-0411, U.S.A.
{y3han, callafon, cortes}@ucsd.edu

J. Jaffe
Scripps Institution of Oceanography, UCSD, 9500 Gilman Drive, La Jolla CA, 92093-0238, U.S.A.
jules@mpl.ucsd.edu

Keywords: Underwater robotics, Submersible drogues, Switching control.

Abstract: This paper analyzes the dynamic properties of a submersible drogue for which buoyancy control is implemented by a flexible membrane and pneumatic actuation. It is shown how a simple on/off switching algorithm tuned on the basis of depth measurements and an estimated depth velocity can be used to achieve accurate depth control with small fluctuations. The switching control makes the pneumatically controlled drogue an inherent hybrid system and conditions for contraction and stability are given for the proposed switching control algorithm. Numerical evaluation of the conditions for stability and experimental data of the switching control implemented on a pneumatic submersible drogue further demonstrates the soundness of the proposed switching control algorithm.

1 INTRODUCTION

A compelling and unanswered need in oceanography is to sample the (coastal) environment at high-resolution spatial and temporal scales than presently possible (Davis, 1991; Kunzig, 1996). Although current systems have led to many important discoveries, oceanographers would agree that many fundamental processes are presently unobservable due to the sparseness of current sampling geometries (Perry and Rudnick, 2003; Schofield and Tivey, 2004). The development of an oceanographic observatory system based on small and inexpensive buoyancy-controlled drogues that are able to perform motion control for collective oceanographic measurements could alleviate the problem of data sparseness.

One of the key requirements on (coordinated) motion is to control the individual depth of such buoyancy-controlled drogues. From a motion control point of view, alternating depth profiles can be used for motion planning purposes when using glider based propulsion (Fiorelli et al., 2003; Bhatta et al., 2005; Paley et al., 2008). On the other hand, small form factor buoyant objects can benefit from strong horizontal shear layers typically observed in shallow coastal ocean flows (Helfrich and White, 2007; Ly and Luong, 1997) to perform motion control by depth planning. Buoyancy induced motion should be done with as little control energy as possible to maximize deployment time for scientific data acquisition purposes. Requirements on energy storage are limited due to the desire of a small form factor design to simulate a free floating, low Reynolds Lagrangian-based distributed sensor system.

In this paper we will analyze the design of a single small form factor buoyancy-controlled drogue. Prototypes of buoyancy-controlled drogues are inspired by current activities at Scripps Institute of Oceanography (Colgan, 2006). Compared with these existing designs, we propose a pneumatically controlled drogue with a flexible membrane. As shown in the analysis of the dynamics, compressibility of the flexible membrane leads to an inherent unstable buoyancy equilibrium that requires very little control energy to generate alternating depth profiles. In addition, a separation of battery and pneumatic power reduces electric power and actuator requirement for motion control.

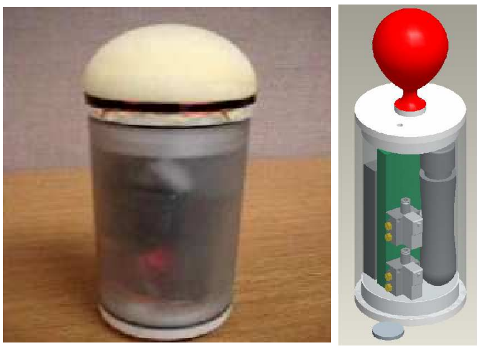
We propose a simple switching (on/off) control algorithm for the buoyancy control of the drogue. It is shown how an on/off switching algorithm tuned

This work was partially supported by NSF Award OCE-0941692.

on the basis of depth measurements and an estimated depth velocity profile can be used to achieve accurate depth control with small fluctuations. The switching control makes the pneumatically controlled drogue an inherent hybrid system (Van der Schaft and Schumacher, 2000) and a formal result on contraction and stability evaluated is given for the proposed control algorithm. The formal result on stability is evaluated numerically and both simulation and experimental data of the switching control implemented on a pneumatic submersible drogue are included in this paper to demonstrate the effectiveness of the proposed pneumatically switching buoyancy-controlled drogues.

2 ILLUSTRATION OF DESIGN CONCEPT

Previous work at Scripps Institute of Oceanography (Colgan, 2006) has illustrated the feasibility of building a standalone ball-shaped free-floating drogue vehicle. A similar, but smaller, less expensive and potentially more energy efficient concept based on a pneumatic controlled flexible membrane attached to the drogue is proposed in this paper for buoyancy control up to a depth of 150 feet. A prototype of the cylindrical shaped (1 liter in volume) membrane controlled drogue is shown in Figure 1.



(a) (b)

Figure 1: Pneumatic membrane controlled cylindrical shaped (a) drogue. Schematic inside view (b) of cylindrical shaped drogue.

Similar flexible membranes for pneumatic control of underwater drogues have also been used in ARGO floats (Gould, 2003; Gould, 2004), but at a much larger form factor. In these design concepts, the membrane actuation is primarily used for providing the buoyancy control for surfacing of the drogues. Instead, to conform to a small form factor design, we use the displaced volume under the flexible membrane for both buoyancy control and the opportunity to create alternating depth profiles that includes periodic surfacing. Due to the compressibility of the membrane, the slightest perturbation in depth will cause a change in volume and the equilibrium state of the drogue is unstable. This is beneficial for low energy periodic surfacing control. As shown in this paper, feedback control can be used to stabilize the drogue to any desired depth.

3 DYNAMIC MODELING OF PNEUMATIC DROGUE

3.1 Rigid Body Dynamics

Assuming a drag parameter d due to a drag force in water, the depth $x(t)$ of the drogue can be described by the second order differential equation

$$m\ddot{x}(t)+\text{sign}(\dot{x}(t))d\dot{x}^2(t)=F_g-F_b(t) \tag{1}$$

where the constant rigid mass of the drogue and added mass due to displacement of fluid (Brennen, 1982) are combined in the mass parameter m. In the equation (1), F_g is the downward (constant) gravitational force

$$F_g=mg$$

and $F_b(t)$ is buoyancy force

$$F_b(t)=\rho g[V_b(t)+V]$$

determined by the fixed drogue volume V and the displaced volume $V_b(t)$ under the membrane. The constant ρ is the density of the water, which we assume to be constant at different depths. The drag parameter d is determined by the drag coefficient c_d of the drogue moving in water at a low Reynolds number and typically given by

$$d=\frac{1}{2}c_d\rho A$$

where A denotes the frontal aerial surface of the drogue. In the above equations, we consider $V_b(t)$ as a control variable to control the depth of the drogue using measurements of the depth $x(t)$.

The condition of natural buoyancy at any depth $c\geq 0$ is determined by the initial conditions $x(0)=x_0$ and $\dot{x}(0)=0$ for the differential equation given in (1) and yields the desired (initial) volume

$$V_b(0)=V_0=\frac{m}{\rho}-V \tag{2}$$

for natural buoyancy. Ideally, the mass m and the volume V of the drogue should be chosen to allow for a

Water density does depend on salidity and depth, but these second order effects are neglected here large range in $V_b(t) > 0$ for buoyancy control. However, even small changes in $V_b(t)$ suffice in generating motion of the drogue due to no-friction free body movement of the drogue floating in water.

3.2 Actuator Dynamics

Using a pneumatic mechanism to add or bleed air to effectively increase $V_b(t)$ induces additional (actuator) dynamics that needs to be taken into account in order to design a control algorithm. The displaced volume $V_b(t)$ under the membrane is a function of both the air flow, the stiffness of the membrane and the external pressure surrounding the membrane. Considering a flow $\phi(t)$ used to change the volume $V_b(t)$ under the membrane, with the ideal gas equation (Fox et al., 2004) we see that the product of the membrane pressure $P_b(t)$ and volume $V_b(t)$ can be described by

$$P_b(t)V_b(t) = \left(n_0 + \int_{\tau=0}^{t} \phi(\tau)d\tau\right) RT(t) \qquad (3)$$

where n_0 indicates the number of gas molecules for natural buoyancy at $t = 0$, R is the Boltzman constant and $T(t)$ is the gas temperature which we assume to be known (measurable) for now. Finally, we assume a linear increase of the membrane pressure $P_b(t)$ due to changes in the volume $V_b(t)$ according to

$$P_b(t) = k_m V_b(t) + P_x(t),\ P_x(t) = k_x x(t) + P_0 \qquad (4)$$

where k_m is the (linear) membrane membrane stiffness and $P_x(t)$ models the effect of the external pressure as the sum of the atmospheric pressure P_0 and the product of the depth $x(t)$ and the pressure depth constant k_x given by

$$k_x = \rho g$$

The last equation allows us to compute the unknown n_0 in (3) at time $t = 0$ for which we have natural buoyancy and $\phi(0) = 0$. With the desired initial displacement volume V_0 under the membrane given in (2), we see that at an initial depth $x(0) = x_0 > 0$

$$P_b(0) = k_m V_0 + k_x x_0 + P_0,\ \ V_0 = \frac{m}{\rho} - V$$

allowing us to compute n_0 as

$$n_0 = \frac{k_m}{RT_0} V_0^2 + \frac{k_x x_0 + P_0}{RT_0} V_0,\ \ V_0 = \frac{m}{\rho} - V$$

indicating quadratic dependency on the initial displaced volume V_0 under the membrane due to the linear membrane stiffness k_m and linear dependency due to influence of the depth dependent pressure.

To complete the analysis, we still need an equation that describes the flow $\phi(t)$. To anticipate the switching control proposed in this paper, the flow $\phi(t)$ is modulated by three different switching states: *inflate*, *deflate* or *none*. The input u can switch between the values

$$u = \begin{cases} 1 & \text{inflate} \\ -1 & \text{deflate} \\ 0 & \text{none} \end{cases}$$

can be used to distinguish between three different switching states. Using the Bernoulli equation (Fox et al., 2004) and assuming a steady incompressible flow, each state has a different flow $\phi(t)$ that is modeled via a proportional relationship with the square root of the pressure difference:

$$\phi(t) = \begin{cases} k_v^i \sqrt{P_{CO_2} - P_b(t)} & u = 1 \\ -k_v^d \sqrt{P_b(t) - P_x(t)} & u = -1 \\ 0 & u = 0 \end{cases}$$

where P_{CO_2} denotes the (constant) output pressure of the CO_2 pressure regulator and k_v^i, k_v^d denote the valve constants of respectively the add and bleed valves. If both valves are the same, then $k_v = k_v^i = k_v^d$. With the pressure relationship given in (4) we see that the gas flow $\phi(t)$ can now be modeled as

$$\phi(t) = \begin{cases} k_v^i \sqrt{P_{CO_2} - k_x x(t) - k_m V_b(t) - P_0} \\ -k_v^d \sqrt{k_m V_b(t)} \\ 0 \end{cases} \qquad (5)$$

respectively for $u = 1$, $u = -1$ and $u = 0$. As a result, membrane inflation and deflation will occur at different flow rates. Membrane inflation is determined mainly by the CO_2 pressure regulator, but pressure build up due to depth and membrane stiffness (volume) has a negative influence. For deflation it can be seen that only the membrane stiffness contributes to a desired pressure difference and is independent of depth.

3.3 Combined Dynamic Model

Combing the different equations and eliminating intermediate variables leads to a dynamic switching system described by a set of coupled non-linear and non-stiff ordinary differential equations (Hairer et al., 1993) in depth $x(t)$, volume $V_b(t)$ and gas flow $\phi(t)$ given by

$$\ddot{x}(t) = -\text{sign}(\dot{x}(t))\frac{d}{m}\dot{x}^2(t) + \frac{\rho g}{m}(V_0 - V_b(t))$$

$$k_m V_b(t)^2 + [k_x x(t) + P_0]V_b(t) =$$

$$\left(n_0 + \int_{\tau=0}^{t} \phi(\tau)d\tau\right) RT(t)$$

$$\phi(t) = \begin{cases} k_v^i \sqrt{P_{CO_2} - k_x x(t) - k_m V_b(t) - P_0} & u = 1 \\ -k_v^d \sqrt{k_m V_b(t)} & u = -1 \\ 0 & u = 0 \end{cases}$$

where a summary of the meaning of the physical parameters is given in Table 2 in the Appendix of this paper. For analysis purposes, the above dynamical model is written in a short hand notation

$$\dot{z}(t) = f_u(z(t)) \tag{6}$$

where $z(t) = [x(t)\ \dot{x}(t)\ \phi(t)]^T$ combines the state variables and $f_u(z)$ for $u \in \{-1, 0, 1\}$ denotes the different dynamic behavior of the system as a function of the switching signal u. Simulations of the dynamics can be done along with the initial conditions

$$z(0) = \begin{bmatrix} x(0) \\ \dot{x}(0) \\ \phi(0) \end{bmatrix} = \begin{bmatrix} x_0 \\ 0 \\ 0 \end{bmatrix}$$

and the resulting initial number of gas molecules n_0 and volume given V_0 by

$$\begin{aligned} n_0 &= \frac{k_m}{RT_0} V_0^2 + \frac{k_x x_0 + P_0}{RT_0} V_0 \\ V_0 &= \frac{m}{\rho} - V \end{aligned} \tag{7}$$

It can be observed from the above equations that the membrane gas temperature T or density parameter ρ can be considered as a time or depth varying parameters that influences the dynamics of the system. The switching signal $u \in \{-1, 0, 1\}$ is the control input signal available to provide depth tracking and stabilization for the drogue.

4 EQUILIBRIUM AND STABILITY ANALYSIS

Equilibrium conditions for the drogue can be studied by setting the flow rate $\phi(t) = 0$ and the switching signal $u = 0$, reducing the differential equations to

$$\begin{gathered} \ddot{x}(t) = -\text{sign}(\dot{x}(t)) \frac{d}{m} \dot{x}^2(t) + \frac{\rho g}{m} (V_0 - V_b(t)) \\ k_m V_b(t)^2 + [k_x x(t) + P_0] V_b(t) = n_0 R T(t) \end{gathered} \tag{8}$$

Due to the assumptions of an ideal gas and linear membrane stiffness k_m, the volume $V_b(t)$ can be solved explicitely from the resulting quadratic equation as a function of depth $x(t)$ and temperature $T(t)$. This yields a single solution under the constraint $V_b(t) > 0$ given by

$$\begin{aligned} V_b(t) = & -\frac{1}{2k_m}[k_x x(t) + P_0] + \\ & \frac{1}{2k_m}\sqrt{[k_x x(t) + P_0]^2 + 4k_m n_0 R T(t)} \end{aligned} \tag{9}$$

With $T(t) = T_0$ and both $x(t),\ V_b(t) > 0$ and n_0 given in (7) it can be seen that the solution

$$\begin{aligned} x(t) &= x_0 > 0 \\ \dot{x}(t) &= 0 \\ V_b(t) &= V_0 = \frac{m}{\rho} - V > 0 \end{aligned}$$

is a stationary point of the equation in (8) as $V_b(t)$ satisfies

$$\begin{gathered} -\frac{1}{2k_m}[k_x x_0 + P_0] + \\ \frac{1}{2k_m}\sqrt{[k_x x_0 + P_0]^2 + 4k_m^2 V_0^2 + 4k_m(k_x x_0 + P_0)V_0} = \\ -\frac{1}{2k_m}[k_x x_0 + P_0] + \\ \frac{1}{2k_m}\sqrt{(2k_m V_0 + [k_x x_0 + P_0])^2} = V_0 \end{gathered}$$

known as neutral buoyancy of the drogue at depth x_0. However, the stationary point is an unstable equilibrium as any perturbation on $x(t) = x_0$ causes an unbounded $x(t)$, physically indicating that the drogue will either surface or sink without control. Such dynamic behavior is due to the compressibility of the membrane and can be used favorably to surface without little or no control energy.

The instability of the equilibrium can be shown by considering a perturbation on the depth $x(t) > 0$ at time $t = t_p$ given by $\ddot{x}(t_p) = 0$ and $x(t_p) = x_0 + \varepsilon,\ \varepsilon > 0$. Due to this small increase in depth,

$$\begin{aligned} V_b(t_p) &= -\frac{1}{2k_m}[k_x(x_0 + \varepsilon) + P_0] + \\ & \quad \frac{1}{2k_m}\sqrt{[k_x(x_0 + \varepsilon) + P_0]^2 + 4k_m n_0 R T_0} \\ &= -\frac{1}{2k_m}[k_x x_0 + P_0] - \frac{1}{2k_m} k_x \varepsilon + \delta \end{aligned}$$

where δ is given by the expression

$$\frac{1}{2k_m}\sqrt{[k_x x_0 + P_0]^2 + 4k_m n_0 R T_0 + k_x^2 \varepsilon^2 + 2k_x(k_x x_0 + P_0)\varepsilon}$$

With $4k_m n_0 R T_0 > 0$, the strict inequality

$$\delta < \sqrt{\left(\sqrt{[k_x x_0 + P_0]^2 + 4k_m n_0 R T_0} + k_x \varepsilon\right)^2}$$

where

$$\begin{gathered} \sqrt{\left(\sqrt{[k_x x_0 + P_0]^2 + 4k_m n_0 R T_0} + k_x \varepsilon\right)^2} = \\ = \sqrt{[k_x x_0 + P_0]^2 + 4k_m n_0 R T_0} + k_x \varepsilon \end{gathered}$$

indicates that

$$\begin{aligned} V_b(t_p) < & -\frac{1}{2k_m}[k_x x_0 + P_0] + \\ & \frac{1}{2k_m}\sqrt{[k_x x_0 + P_0]^2 + 4k_m n_0 R T_0} = V_0 \end{aligned}$$

making the displaced volume $V_b(t_p)$ under the membrane smaller than V_0 due to $x(t_p) = x_0 + \varepsilon$. With $(V_0 - V_b(t_p)) > 0$ and $\ddot{x}(t_p) = 0$, the differential equation for $x(t)$ in (8) indicates $\dot{x}(t_p) > 0$ causing a further increase of the depth $x(t)$ for $t > t_p$. This indicates the increase of the depth $x(t)$ (sinking) and the instability of the neutral buoyancy operating point of the drogue at $x(t) = x_0$. A similar argument can be given for a perturbation $x(t_p) = x_0 - \varepsilon$ that will result in a decrease in the displaced volume $V_b(t_p)$ under the membrane, causing the drogue to surface. The stability analysis indicates that neutral buoyancy is obtained *only* at one specific (initial) depth x_0 such that $V_b(t)$ given in (9) satisfies $V_b(t) = V_0 = m/\rho - V$ to provide a buoyancy force that balances the gravitational force.

5 DEPTH SWITCHING CONTROL

Although the compressibility of the membrane can be used favorably to surface or sink the drogue without little or no control energy, a contraction or stabilization algorithm towards a target depth reference $r(t)$ must be implemented via feedback control. Due to the switching behavior of the control signal $u \in \{-1, 0, 1\}$ we can only expect to obtain depth tracking and neutral buoyancy within a user specified tolerance level α around the depth reference $r(t)$. Our objective is to design a switching control algorithm that guarantees attaining the depth region $|x(t) - r(t)| < \alpha$ from an arbitrary initial depth condition x_0.

In this paper this is achieved by simply modulating the switching signal $u \in \{-1, 0, 1\}$ on the basis of feedback information of the depth $x(t)$ and the depth velocity $\dot{x}(t)$. With a target depth $r(t) > 0$ and a γ with $0 < \alpha < \gamma$, the depth measurements are classified in three different regions that are pairwise disjoint

$$\begin{aligned} \mathcal{R}_1 &= \{x \mid \gamma < |r(t) - x|\} \\ \mathcal{R}_2 &= \{x \mid \alpha < |r(t) - x| \leq \gamma\} \\ \mathcal{R}_3 &= \{x \mid |r(t) - x| \leq \alpha\} \end{aligned} \tag{10}$$

respectively denoted by the attraction region $\mathcal{R}_1$, the stabilization region $\mathcal{R}_2$ and the tolerance region $\mathcal{R}_3$. We will first summarize the proposed switching control algorithm. Subsequently we provide a proposition that shows how contraction to a specific depth region can be obtained.

Algorithm 1. *Let $\mathcal{R}_j,\ j = 1, 2, 3$ be the depth region defined in (10) where $0 < \alpha < \gamma$ and consider the maximum drogue velocities β_j for each region $\mathcal{R}_j$ where $\beta_3 \leq \beta_2 \leq \beta_1$ and let*

$$\sigma = sign(r(t) - x(t)) \in \{-1, 1\}$$

Default, the switching signal $u = 0$ but will be either $u = 1$ (add air to increase the volume $V_b(t)$) or $u = -1$ (bleed air to reduce $V_b(t)$) according to the following rules:

1. *If $x(t) \in \mathcal{R}_1$, consider the two situations:*
   a. *If $|\dot{x}(t)| \leq \beta_1$ then $u = -\sigma$ (increase depth velocity)*
   b. *If $\sigma \cdot \dot{x}(t) < 0$ then $u = -\sigma$ (velocity in wrong direction, increase/decrease depth velocity directly)*
2. *If $x(t) \in \mathcal{R}_2$, consider the two situations:*
   a. *If $|\dot{x}(t)| > \beta_2$ and $\sigma \cdot \dot{x}(t) > 0$ then $u = \sigma$ (decrease velocity)*
   b. *If $\sigma \cdot \dot{x}(t) < 0$ then $u = -\sigma$ (velocity in wrong direction, increase/decrease depth velocity directly)*
3. *If $x(t) \in \mathcal{R}_3$ and $|\dot{x}(t)| > \beta_3$ then $u = sign\{\dot{x}(t)\}$ (chatter input signal to achieve velocity bound)*

As can be seen from the above algorithm, each region $\mathcal{R}_j,\ j = 1, 2, 3$ has specific bounds β_j on the depth velocity $\dot{x}(t)$ that determines the modulating of the switching signal $u \in \{-1, 0, 1\}$. The attraction region $\mathcal{R}_1$ is used to give the drogue the right velocity with a maximum of β_1 to attract it in the stabilization region $\mathcal{R}_2$. When the drogue enters the region $\mathcal{R}_2$, the velocity is reduced to β_2 to avoid abrupt entering and exiting of $\mathcal{R}_2$ and providing an entering velocity of β_2 for the tolerance region $\mathcal{R}_3$. Obviously, the choice of β_2 is closely related to the size of γ. As soon as the drogue enter the tolerance region $\mathcal{R}_3$ the control can be turned off by choosing $\beta_3 = \beta_2$ as the entering velocity will be β_2. Choosing $\beta_3 < \beta_2$ allows an additional reduction of drogue velocity within the tolerance region at the price of a chattering control signal.

The proposed switching control in Algorithm 1 is motivated by the need to dampen the (unstable) rigid body motion of the drogue and maintain neutral buoyancy. The proposed algorithm is basically a Proportional and Derivative (PD) control architecture with specific level sets for position $x(t)$ and velocity $\dot{x}(t)$ measurements. It should be observed that feedback information on $x(t)$ and $\dot{x}(t)$ is required to implement the proposed control algorithm. However, $\dot{x}(t)$ can be approximated by discrete-time sampling and filtering of the depth $x(t)$, allowing a discrete-time implementation of the switching control algorithm based on depth measurements only.

6 HYBRID STABILITY ANALYSIS

For the analysis of the proposed switching control algorithm, the framework of hybrid systems analysis (Van der Schaft and Schumacher, 2000; Liberzon, 2003; Goebel et al., 2009) can be used to prove stability properties of the resulting control system. Based on the switching control law proposed in Algorithm 1, the behavior of the drogue is modeled as a hybrid system with three different modes corresponding to *inflation* for $u = 1$, *deflation* for $u = -1$, and a *neutral* mode for $u = 0$.

Each mode has its active region and these regions share switching surfaces with each other. The transition between modes occurs when the system trajectory crosses these switching surfaces. In this paper, a multiple Lyapunov functions based parameter dependent switching strategy is used to investigate the stability of the proposed switching control algorithm for the buoyancy control of the drogue.

In order to establish an active region for each actuator dynamic mode, three Lyapunov functions V_u, $u \in \{-1,0,1\}$ must be defined. The three quadratic Lyapunov functions V_u are defined as follows:

$$\begin{aligned} V_1 &= \frac{1}{2}(-(x-r)-10\dot{x}+\phi)^2 && \text{inflation} \\ V_{-1} &= \frac{1}{2}(-(x-r)-10\dot{x}+\phi)^2 && \text{deflation} \\ V_0 &= \frac{1}{2}(x-r)^2+\frac{1}{20}\dot{x}^2 && \text{neutral} \end{aligned} \tag{11}$$

Subsequently, the stability analysis is based on a theorem in (Branicky, 1998) on Lyapunov stability of switched and hybrid systems and summarized in the following result for the switched hybrid system considered in this paper.

Proposition 1. *Consider the system $\dot{z}(t) = f_u(z(t))$ in (6) and let V_u be the Lyapunov functions given in (11) for $u \in \{-1,0,1\}$. Let $L_{f_u}V_u$ be the Lie derivative of V_u along the vector field spanned by the subsystem $f_u(\cdot)$. The origin of the system is asymptotically stable if the following two conditions are satisfied*

$$\begin{aligned} &(a) \quad L_{f_u}V_u < 0 \textit{ for each } u \in \{-1,0,1\} \\ &(b) \quad V_u \textit{ is nonincreasing along switching} \\ &\qquad \textit{times on the uth system } \dot{z}(t) = f_u(z(t)) \end{aligned} \tag{12}$$

In this paper we use numerical analysis to verify the two conditions (12) in Proposition 1. It can be verified that each Lyapunov function V_u for $u \in \{-1,0,1\}$ in (11) is positive definite over its active region characterized by $L_{f_u}V_u < 0$. To characterize (12) over the three dimensional state vector $z = [x\ \dot{x}\ \phi]^T$ Lyapunov analysis is conducted by varying ϕ in the admissible range. This can be done as ϕ is not used directly in the switching control of Algorithm 1. Numerical evaluation of (12) for different values of ϕ is done using a non-stiff differential equation solver (Cooper, 2004) and implemented via `ode45` in Matlab.

For the inflation mode, $L_{f_1}V_1$ for $\phi = 0$ is plotted in Figure 2 and the active region for which (12) holds is separated by a dashed line, whereas the arrows indicate a decreasing value of V_1. If ϕ is varied, the whole solution surface moves through the ϕ-axis (perpendicular to $x - \dot{x}$ plane) resulting in the shift of the active region along the $\dot{x}$-axis. The same analysis can be done for other modes. A similar plot is created for the deflation mode where $L_{f_{-1}}V_{-1}$ for $\phi = 0$ is plotted in Figure 3 and the neutral mode where $L_{f_0}V_0$ for $\phi = 0$ is plotted in Figure 4.

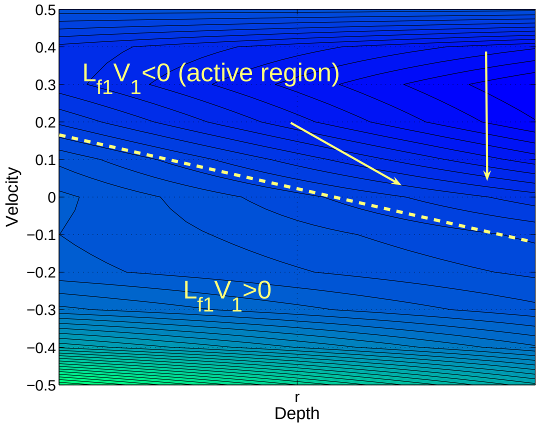


Figure 2: The active region of the inflation mode. The arrows indicate the allowable direction of solution trajectory. In inflation mode, the solution trajectory always follows this allowable direction.

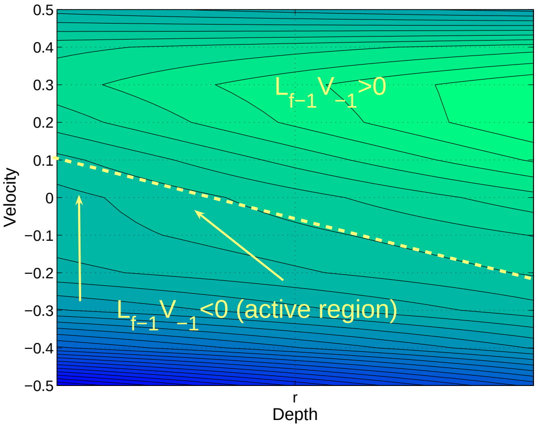


Figure 3: The active region of the deflation mode. The arrows indicate the allowable direction of solution trajectory. In deflation mode, the solution trajectory always follows this allowable direction.

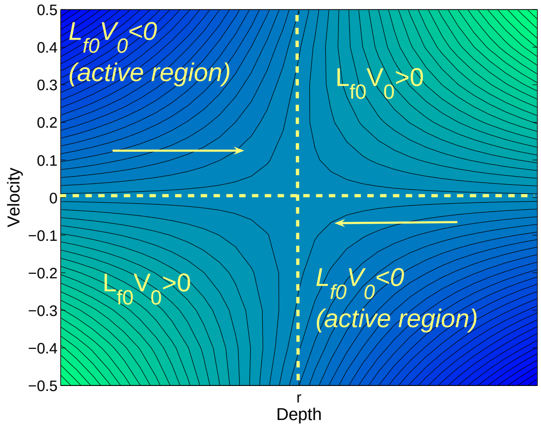


Figure 4: The active region of the neutral mode. The arrows indicate the allowable direction of solution trajectory. In neutral mode, the solution trajectory always follows this allowable direction.

Table 1: Numeric parameters of dynamic drogue model for non-linear simulation of switching control algorithm.

symbol	value	units
V	10^{-3}	m^3
m	1.5	kg
d	30	$\mathrm{N/m^2/s^2}$
ρ	$1.03\cdot 10^3$	$\mathrm{kg/m^3}$
g	9.81	$\mathrm{m/s^2}$
k_m	$5\cdot 10^8$	$\mathrm{Pa/m^3}$ or $\mathrm{N/m^5}$
T_0	288	K
R	8.314472	$(\mathrm{m^3Pa})/(\mathrm{K\ mol})$
k_a	10^{-4}	$\mathrm{mol}/\sqrt{\mathrm{Pa}}$
k_b	10^{-4}	$\mathrm{mol}/\sqrt{\mathrm{Pa}}$
P_{CO_2}	10^6	Pa or $\mathrm{N/m^2}$
P_0	10^5	Pa or $\mathrm{N/m^2}$
x_0	10	m

Based on these figures, the existence of a stability region characterized by (12) can be verified. Therefore, the proposed hybrid control strategy using the active regions and switching surfaces stabilizes the drogue system.

7 SIMULATION RESULTS

To illustrate the performance of the switching control in Algorithm 1, the controlled dynamic behavior of a model of a buoyancy controlled submersible drogue is simulated. The model is based on the set of coupled non-linear differential equations given in (6) and based on the numerical parameters listed in Table 1. A summary of the meaning of the physical parameters is given in Table 2 in the Appendix of this paper.

Using the switching control summarized in Algorithm 1 with the regions of (10) based on a depth tolerance $\alpha = 0.5$m and a stabilization region specified by $\gamma = 2\alpha = 1$m and a maximum velocity $\beta_1 = 0.3$m/s in region $\mathcal{R}_1$ and a maximum velocity $\beta_2 = \beta_3 = 0.1$m/s in region $\mathcal{R}_2$ and $\mathcal{R}_3$, the performance of the switching control algorithm can be evaluated by the simulation results depicted in Figure 5. The results were computed by numerically solving the non-stiff differential equations of the ODE's of the model of the drogue using an implementation of the explicit Runge-Kutta (4,5) pair implemented in `ode45` in Matlab (Shampine and Reichelt, 1997).

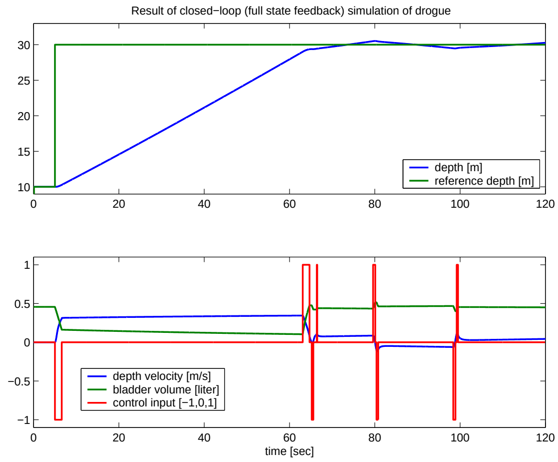


Figure 5: Simulation results of switching control Algorithm 1 with $\alpha = 0.5$m, $\gamma = 1$m, $\beta_1 = 0.3$m/s and $\beta_2 = \beta_3 = 0.1$m/s when the target depth references $r(t)$ changes from 10m to 30m at $t = 5$sec. The numerical values for the drogue model used during this simulation are listed in Table 1.

During the simulation the target depth $r(t)$ of $10m$, for which the drogue is initially neutrally buoyant, is changed to $30m$ at $t = 5$sec. It can be seen from the simulation results in Figure 5 that the drogue stays within approximately $\pm\alpha$m of the a constant target depth, whereas a change to a different target depth is accomplished with a maximum speed of $\beta_1 = 0.3$m/s and only requires a few switches of the control signal u. During steady state operation only very short actuation of either the ”add” ($u = 1$) or ”bleed” ($u = -1$) valves are used to maintain depth within the specified tolerance region α and stabilization region γ. The simulation confirms the hybrid stabilization results.

8 EXPERIMENTAL RESULTS

In an initial testing phase, the proposed control algorithm was implemented on the pneumatic membrane controlled cylindrical shaped drogue depicted in Figure 1(a). The drogue design uses an internally stored standard compressed 16 gram CO_2 cartridge with a regulator assembly to change the displaced volume under the latex or neopropene membrane via two small form factor Numatics series solenoid valves. One valve is used for inflating the membrane, the other valve is to bleed the CO_2 from the membrane.

The electrical components are powered by three 3.7Volt, 2700mAH Li-Ion batteries. A model 85 Ultra Stable Pressure Sensor is used for depth measurements measured by a 10bit AD converter on a Microsystems PIC18F4620 microprocessor.

The embedded control of the pneumatically controlled drogue only uses depth measurements $x(t)$ sampled at 10Hz and depth velocity estimates $v(t)$ are obtained via

$$v(t) = \frac{x_f(t) - x_f(t)}{\Delta_t}, \; x_f(t) = F(q)x(t)$$

where $F(q)$ is a second order Butterworth filter with a normalized cut-off frequency of 0.1 (1Hz). Both the depth measurements $x(t)$ and the control signal $u(t)$ as a function of the discrete-time t were saved for validation purposes and the results of the experimental work is summarized in Figure 6.

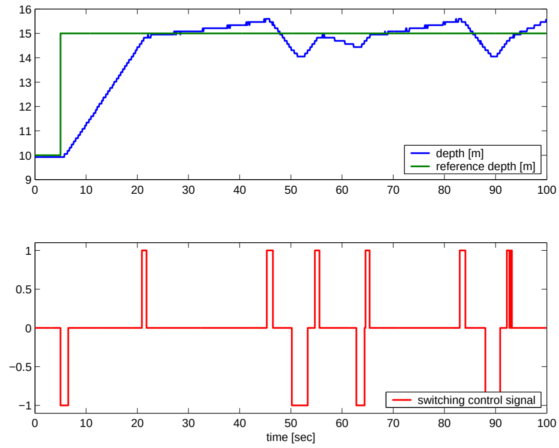


Figure 6: Preliminary experimental results of switching control Algorithm 1 with $\alpha = 0.5m$, $\gamma = 1$, $\beta_1 = 0.3$m/s and $\beta_2 = \beta_3 = 0.2$m/s applied to the pneumatic membrane controlled cylindrical shaped drogue depicted in Figure 1(a). The target depth references $r(t)$ changes from 10m to 15m at $t = 5$sec.

The experimental results confirm the stability of the control algorithm for the pneumatic membrane cylindrical shaped drogue even for the discrete-time implementation of the algorithm. The slightly larger values of β_2 and β_3, chosen due to the noise levels on the estimated velocity $v(t)$, cause larger velocity swings in the stabilization and tolerance region. Moreover the quantization effects of the depth measurements based on a 10bit AD converter also cause resolution limiations on the velocity estimate. It can be seen that a larger value for β_2 requires more switching for stabilization. Tuning of the controller parameters α, γ, β_1 and β_2 can be used to further improve the controller performance.

9 CONCLUSIONS

The dynamic properties of a submersible drogue for which buoyancy control is implemented by a flexible membrane can be described by a set of coupled nonlinear and non-stiff ordinary differential equations. It is shown that the compressibility of the membrane leads to an dynamically unstable dynamical system in terms of the depth, which can be used favorably to surface or sink the drogue without little or no control energy.

The instability does require a contraction or stabilization algorithm to maintain a target depth reference. In this paper it is shown that a simple pneumatic on/off switching control algorithm in which compressed CO_2 is either added or extracted from the membrane actuator on the basis of three different and pairwise disjoint depth regions can be used to stabilize the depth positioning of the drogue.

The switching control algorithm leads to a hybrid dynamical system, for which stability analysis results are summarized in the paper. Numerical evaluation of the stability conditions reveal that the proposed on/off switching control algorithm leads to a stabilized buoyancy-controlled drogue. Both simulation and experimental studies indicate stability properties and depth tracking performance within a specified tolerance levels.

REFERENCES

Bhatta, P., Fiorelli, E., Lekien, F., Leonard, N. E., Paley, D., Zhang, F., Bachmayer, R., Davis, R. E., Fratantoni, D. M., and Sepulchre, R. (2005). Coordination of an underwater glider fleet for adaptive ocean sampling. In *International Workshop on Underwater Robotics, Int. Advanced Robotics Programmed (IARP)*, Genoa, Italy.

Branicky, M. (1998). Multiple Lyapunov functions and other analysis tools for switched and hybrid systems.

IEEE Transactions on Automatic Control, 43:475–482.

Brennen, C. (1982). A review of added mass and fluid inertial forces. Technical Report CR82.010, Naval Civil Engineering Laboratory.

Colgan, C. (2006). Underwater laser shows. *Explorations, Scripps Institution of Oceanograhpy*, 12:20–27.

Cooper, J. (2004). *An Introduction to Ordinary Differential Equations*. Cambridge University Press.

Davis, R. (1991). Observing the general circulation with floats. *Deep-Sea Research*, 38:531–571.

Fiorelli, E., Bhatta, P., Leonard, N. E., and Shulman, I. (2003). Adaptive sampling using feedback control of an autonomous underwater glider fleet. In *Proceedings 13th International Symposium on Unmanned Untethered Submersible Technology*, Durham, NH.

Fox, R., McDonald, A., and Pritchard, P. (2004). *Introduction to Fluid Mechanics*. John Wiley & Sons Inc., Hoboken, NJ, U.S.A.

Goebel, R., Sanfelice, R. G., and Teel, A. R. (2009). Hybrid dynamical systems. *IEEE Control Systems Magazine*, 29:283.

Gould, J. (2004). Argo profiling floats bring new era of in situ ocean observations. *Earth and Ocean Sciences*, 85(11).

Gould, W. J. (2003). WOCE and TOGA - the foundations of the global ocean observing system. *Oceanography, Special Issue on Ocean Observations*, 16(4):24–30.

Hairer, E., Nørsett, S. P., and Wanner, G. (1993). *Solving Ordinary Differential Equations I: Nonstiff Problems*. Springer Verlag, Berlin.

Helfrich, K. and White, B. (2007). Rapid gravitational adjustment of a horizontal shear layer. In *American Physical Society, 60th Annual Meeting of the Divison of Fluid Dynamics*.

Kunzig, R. (1996). A thousand diving robots. *Dis. Mag.*, 17:60–63.

Liberzon, D. (2003). *Switching in Systems and Control*. Systems & Control: Foundations and Applications series. Birkhauser, Boston.

Ly, L. N. and Luong, P. (1997). A mathematical coastal ocean circulation system with breaking waves and numerical grid generation. *Applied Mathematical Modelling*, 21:633–641.

Paley, D., Zhang, F., and Leonard, N. E. (2008). Cooperative control for ocean sampling: The glider coordinated control system. *IEEE Transactions on Control Systems Technology*, 16:735–744.

Perry, M. and Rudnick, D. (2003). Observing the ocean with autonomous and lagrangian platforms and sensors (ALPS): The role of ALPS in sustained ocean observing systems. *Oceanography*, 4:31–36.

Schofield, O. and Tivey, M. (2004). Building a window to the sea: Ocean research interactive observatory networks (ORION). *Oceanography*, pages 113–120.

Shampine, L. F. and Reichelt, M. W. (1997). The Matlab ODE suite. *SIAM Journal on Scientific Computing*, 18:1–22.

Van der Schaft, A. and Schumacher, H. (2000). *An Introduction to Hybrid Dynamical Systems*. Lecture Notes in Control and Information Sciences 251, Springer-Verlag.

APPENDIX

Table 2 summarizes the symbols use in the derivation of the dynamical model of the drogue in Section 3.3.

Table 2: Summary of symbolic parameters of dynamic drogue model.

symbol	meaning
V	volume of drogue
m	mass of drogue
d	drag parameter in water
ρ	density of sea water
g	gravitational constant
k_m	linear membrane stiffness
T_0	temperature of water
R	gas constant
k_a	‘add air’ valve constant
k_b	‘bleed air’ valve constant
P_{CO_2}	regulated CO_2 pressure
P_0	atmospheric pressure
x_0	depth for neutral buoyancy

VISUAL-BASED DETECTION AND TRACKING OF DYNAMIC OBSTACLES FROM A MOBILE ROBOT

Dora Luz Almanza-Ojeda, Michel Devy and Ariane Herbulot
CNRS, LAAS, 7 avenue du Colonel Roche, F-31077 Toulouse, France
Université de Toulouse, UPS, INSA, INP, ISAE, LAAS-CNRS, F-31077 Toulouse, France
{dlalmanz, michel, herbulot}@laas.fr

Keywords: Moving obstacles, Detection, Tracking, Clustering, Monocular vision.

Abstract: This paper presents a methodology for detecting and tracking moving objects during mobile robot navigation in unknown environments using only visual information. An initial set of interest points is detected and then tracked by the Kanade-Lucas tracker (KLT). Along few images, point positions and velocities are accumulated and a spatio-temporal analysis, based on the a contrario theory, is performed for the clustering process of these points. All dynamical sets of points found by the clustering are directly initialized and tracked as moving objects using Kalman Filter. At each image, a probability map saves temporally the previous interesting point positions with a certain probability value. New features will be added in the most likely zones based on this probability map. The process detection-clustering-tracking is executed in an iterative way to guarantee the detection of new moving objects or to incrementally enlarge already detected objects until their real limits. Experimental results on real dynamic images acquired during robot outdoor and indoor navigation task are presented. Furthermore, rigid and non rigid moving object tracking results are compared and discussed.

1 INTRODUCTION

A key function required for autonomous robot navigation, must cope with the detection of objects close to the robot trajectory, and the estimation of their states. This function has been studied by the robotic and the ITS (Intelligent Transportation Systems) communities, from different sensory data such as laser, radar, vision, among others. For driver assistance, many contributions concern laser-based obstacle detection and tracking (Vu and Aycard, 2009). In order to track obstacles, it is required to estimate from the same sensory data, the egomotion of the vehicle: a global probabilistic framework known as SLAMMOT (Simultaneous Localization and Mapping with Mobile Object Tracking) has been proposed by Wang et al. (Wang et al., 2007). But in spite of numerous contributions, this function still remains a challenge when it is based only on vision. So this paper proposes a strategy to detect static points, and moreover to detect and cluster the moving ones in order to track mobile objects: it is the first step towards the full integration of a Visual SLAMMOT approach. It is proposed to reach this objective, using only a monocamera system.

There are various methods for feature detection and tracking using a single mobile camera. A. J. Davison (Davison, 2003) has proposed a spatio-temporal approach, in order to detect and track 2D points from an image sequence and then to reconstruct corresponding 3D points used to locate the camera. The Harris detector allows to initialize points to be searched in successive images from an active strategy, i.e. selecting the best points to be tracked. Each point is represented by its appearance, i.e. a $n \times n$ template around its first position. The system state at instant time t, is given by the 3D position and the speed of every tracked point, as well as the position and the speed of the camera. From this state, every point position is predicted in the image acquired at time $t+1$, forming an elliptic zone where the point must be found if the global state is consistent. Then, each interest point is searched in its predicted zone by a similarity measurement based on correlation score using its template. A clever strategy must be used to update the point template in order to better handle luminance variations and partial object occlusions.

A method widely used for robotics applications is based on the optical flow. Dense optical flow is a hard computing procedure that makes it not so interesting for real time applications. However, if optical flow is only extracted for interest features, i.e. for a very small part of total image points, a tracking procedure of many objects characterized by some points, could be applied in real time. Such a sparse solution has been proposed by Shi-Tomasi (Shi and Tomasi, 1994) and it is commonly used in computer vision because of its simplicity and its low computational cost. Our own method is based also on this method as a valid and confirmed procedure, that can be applied in a real time context during navigation. Among others, let us cite the work presented in (Lookingbill et al., 2007) where authors have used the optical flow field to leverage the difference in appearance between objects at close range and the same objects at more distant locations. This information allows them to interpret monocular video streams during off-road autonomous navigation and to propose an adaptive road following in unstructured environments. This method has been evaluated for the navigation of an intelligent vehicle in a desertic terrain.

However, once interest points and optical flow are extracted and tracked from an image sequence, it is so important to distinguish which of those tracked points represent moving objects. Clustering techniques are the first basic solution to this question, but unfortunately most of them require initial information about the scene as the number of clusters to find. The success of these methods highly depends on these parameters. T. Veit et. al (Veit et al., 2007) coped with the same issue for the analysis of short video sequences. They validate a clustering algorithm based on the *a contrario* method (Desolneux et al., 2008) which does not need parameter tuning or initial scene information for finding clusters of mobile features. This approach has been also used in (Poon et al., 2009) to detect moving objects in short sequences; additionally, authors obtain 3D components of feature points to better detect the correspondence between points and moving objects on which they have been extracted. This work presents experimental results on real images, acquired from fixed cameras, so that essential issues of autonomous navigation are not considered.

This paper proposes moving object detection and tracking on a robot navigation context based on KLT tracker and the a contrario clustering. Then, the resulted clusters are initialized as moving objects and tracked by a Kalman Filter. At each iteration, image locations are ponderated by a bi-variate pdf function when a point is tracked. This value represents the probability that the location contains a dynamical point, being 0 the most interesting location where to find an interesting point.

Next section explains the proposed strategy. KLT procedure used to detect and track interest points is briefly described in section 3. Section 4 describes the main concepts of the *a contrario* theory. The global detection-clustering-tracking approach and the use of probability map in a long sequence of images are presented in section 5. Experimental results on real images are presented and discussed in section 6. Finally last section concludes and explains future works.

2 OVERALL DETECT, CLUSTER AND TRACK STRATEGY

In this work only visual information in grey-scale acquired from a mobile robot is used for scene analysis. General block diagram in Figure 1 describes the procedure carried out to detect and track multiple moving objects.

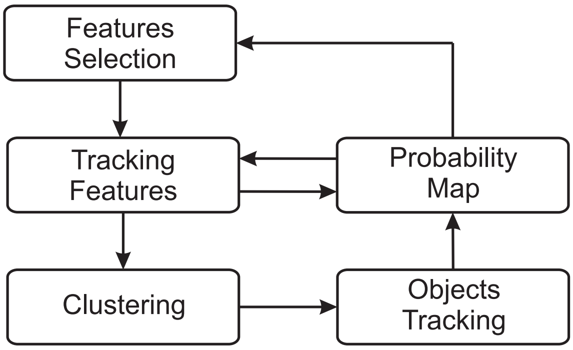


Figure 1: Algorithm to detect and track multiple objects.

N initial feature points are selected in the input image(t) using the Shi-Tomasi approach. Feature locations in next image are searched by the KLT tracker, based on correlation and optimization processes. Let us suppose first that no object is tracked; so the process loops on N_{im} successive images executing only two tasks, feature tracking and update of the probability map. We will call *time of trail*, this set of N_{im} processed images, i.e. the number of images used to accumulate positions and apparent velocities of tracked features. In order to select new features, we seek into the probability map the most interesting zones in the image for searching new points, i.e. points which have the higher probability to be on mobile objects. KLT process is executed continuously in this way, while the robot navigates in order to provide new visual information at each time of trail. Details of the KLT theory are explained in next section.

At the end of each time of trail, only moving KLT features are selected for being grouped by the a contrario clustering method. Here, moving KLT features are characterized by a velocity vector higher than a threshold set to 1 pixel. Resulted clusters represent moving objects in the scene; every one is directly initialized as a moving object and then tracked with a position and an apparent velocity estimated by a Kalman Filter. The Object tracking task exploits the KLT tracker: it verifies the consistency between the point tracking in the current image and the Object characteristics. At every iteration, this object could be merged with another detected object based on similar velocity and close position. Implementation details of both clustering and merging process are presented in section 4. Finally, object current positions are stored in the probability map with the highest probability of occupation, so this zone of the image is not interesting to look for new features.

3 FEATURE SELECTION AND TRACKING

Optical flow procedure used in this work is based on the initial technique proposed in (Lucas and Kanade, 1981) and on the well-known *Select Good Features to Track* algorithm (Shi and Tomasi, 1994). This technique is largely used in the robotics community because it proposes to match the more salient points, minimizing the processing time.

Detection of Moving Points. N distinctive feature points are initially extracted from image t_0 by the analysis of spatial image gradient in two orthogonal directions. Locations of these N points in next image are obtained by maximizing a correlation measurement over a small window. Iterative process is accelerated by constructing a pyramid with scaled versions of the input image. Furthermore, rotation, scaling and shearing are applied on each correlation window by optimizing a linear spatial transformation parameters during iterative process. Once displacement vectors are obtained for all initial features, apparent velocities are estimated based on displacement vectors.

This KLT optical flow method is sparse because only few points are initially selected to describe image content. This sparse method is used in order to save processing time, because some other essential tasks must be performed. Figure 2 (frames 54 to 58 extracted from the sequence shown in second row of Figure 5) depicts feature points detected and tracked during a time of trail from $N = 150$ initial selected points. From these accumulated locations, blue points represent points with long optical flow displacements. For these points the assumption that they belong to a moving object has a high probability.

Perception on Moving Objects. When moving objects are detected, the number of dynamic points is subtracted from the N points tracked permanently by KLT. Thus in following iterations, KLT will select less than N new points. This strategy allows to track only a fix number of N points between KLT and the Object tracking process. This rigorous control on the number of points is important in our methodology because long image sequences will be evaluated and the performance directly depends on the number of processed points.

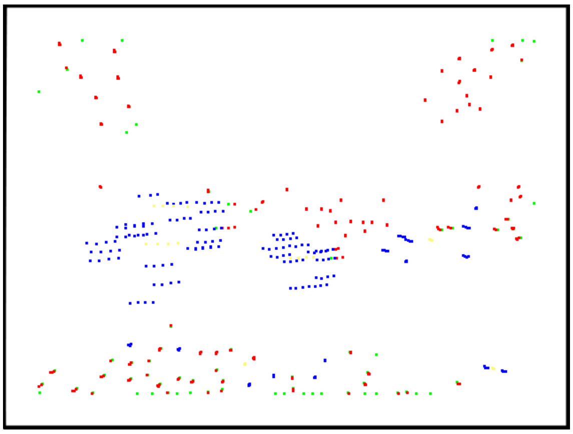


Figure 2: Accumulated Optical flow in 5 successive images from the sequence shown in the second row of Figure 5.

Figure 3 shows N feature points with locations tracked by KLT process at three different times of trail (8, 18 and 30). After 18 images and without any prior knowledge of the environment, we perceive consistent points motions, caused by the camera movement. By analyzing optical flow extracted from images acquired from a moving camera in dynamic environments, we deduce that larger is the time of trail, better will be the perception of moving objects, but also of the camera motion. In order to minimize the egomotion impact, we propose to reduce this time to 5 images which covers up egomotion and also, points out independent movements. This can be verified in Figure 2; a constant movement in the cloud of points can be seen in the middle of the image; egomotion is less remarkable. Furthermore, the best advantage of this choice $N_{im} = 5$ is the reduction of the waiting time before performing the a contrario clustering task.

4 A CONTRARIO METHOD AND MERGING PROCESS

Visual perception is a complex function that requires the presence of ”salient” or ”evident” patterns to iden-

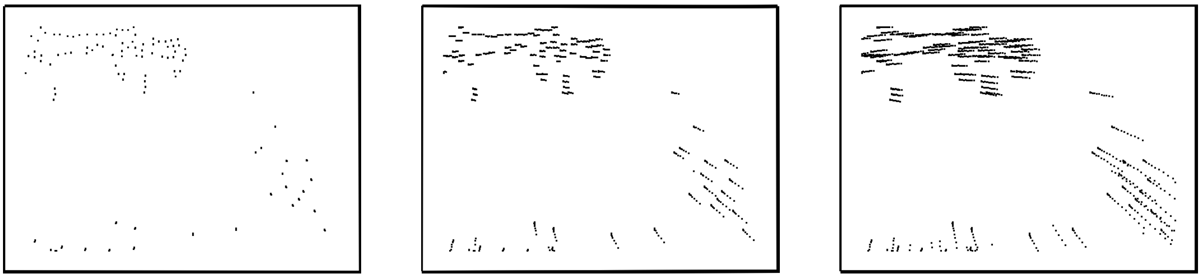


Figure 3: From $N = 150$ selected points in the first image, tracking results on a sequence acquired when the robot turns left: accumulated Optical flow along 8 (left image), 18 (middle) and 30 (right) images.

tify something that ”breaks” the continuous motion due to the camera egomotion. This ”salient pattern” corresponds to ”meaningful event” detected by the a contrario method (Cao et al., 2007). Basic concepts of the a contrario clustering inspired by the Gestalt theory, are exposed in (Desolneux et al., 2003) and deeply in (Desolneux et al., 2008). In general, Gestalt theory establishes that groups could be formed based on one or several common characteristics of their elements. In accord to this statement, an a contrario clustering technique (proposed by Veit, et. al. (Veit et al., 2007)) identifies one group as meaningful if all their elements show a different distribution than an established background random model. Contrary to most of clustering techniques, neither initial number of clusters is required nor parameter has to be tuned. In our context, i.e. unknown environment, approximated egomotion..., these characteristics are very favorable.

4.1 Evaluation of a Background Model

We use the background model proposed in (Veit et al., 2007) which establishes a random organization of the observations. Hence, background model elements are independent identically distributed (iid) and follow a distribution p. The iid nature of random model components propose an organization with not coherent motion present.

Next, given an input vector $V(x,y,v,\theta)$ in R^4, first objective is to evaluate which elements in V show a particular distribution contrary to the established distribution p of the background model (that explains ”a contrario” name). To avoid element by element evaluation, first, a binary tree with V elements is constructed using a single linkage method. Each node in the tree represents a candidate group G that will be evaluate in a set of given regions designed by $\mathcal{H}$. This set of regions is formed by different size of hyper-rectangles that will be used to test the distribution of several data groups. Each region H is centered at each element X in the group until finding the region H_X that contains all the group and at the same time makes minimal the probability of the background model distribution. Different size of hyper-rectangles are used in function of data range, in our experiments we use 20 different sizes by dimension. Final measure of meaningfulness (called Number of False Alarms NFA in referenced work) is given by eq 1.

$$NFA(G) = N^2 \cdot \left|\mathcal{H}\right| \min_{\substack{X\in G,\\ H\in\mathcal{H},\\ G\subset H_X}} B(N-1, n-1, p(H_X)) \tag{1}$$

In this equation N represents the number of elements in vector V, $\left|\mathcal{H}\right|$ is the carnality of regions and n is the elements in group test G. The term which appears in the minimum function is the accumulated binomial law, this represents the probability that at least n points including X are inside the region test centered in X (H_X). Distribution p consists of four independent distributions, one for each dimension data. Point positions and velocity orientation follow a uniform distribution because object moving position and direction are arbitrary. In other hand, velocity magnitude distribution is obtained directly of the empirically histogram of the observed data. So that, joint distribution p will be the product of this four distributions. A group G is said to be meaningful if $NFA(G) \leq 1$.

Furthermore two sibling meaningful groups in the binary tree could belong to the same moving object, then a second evaluation for all the meaningful groups is calculated by Eq. 2.To obtain this new measure, we reuse region group information (dimensions and probability) and just a new region that contains both test groups G_1 and G_2 is calculated. New terms are $N' = N-2$, number of elements in G_1 and G_2, respectively $n_2' = n_1 - 1$ and $n_2' = n_2 - 1$, and term $\mathcal{T}$ which represents the accumulated trinomial law.

$$NFA_G(G_1,G_2) = N^4 \cdot \left|\mathcal{H}\right|^2 \mathcal{T}\left(N', n_1', n_2', p_1, p_2\right) \tag{2}$$

Both mesures 1 and 2 represent the significance of groups in binary tree. Final clusters are found by exploring all the binary tree and comparing if it is more significant to have two moving objects G_1 and G_2 or to fusion it in a group G. Mathematically, $NFA(G) < NFA_G(G_1, G_2)$ where $G_1 \cup G_2 \subset G$.

4.2 Merging Groups

This function is executed only if moving objects have already been detected. O is a set of M objects that will contain all candidate objects for merging evaluation. That is, $O = O_T \cup O_C$ where O_T consists of $(1, 2, ..., k)$ moving objects tracked by Kalman filter, and O_C consist of $(1, 2, ..., l)$ new moving clusters, interpreted either as new moving objects, or part of existing ones . For each object in O, the velocity vector is modeled by the mean of their velocity components in X and Y, respectively represented by μ_{vX} and μ_{vY} . We use these models to evaluate eq.3 that let establish a decision constraint for merging.

$$\min_{\substack{i,j \in M,\\ i \neq j,\\ O_i, O_j \subset O}} \left(\begin{bmatrix} s(\mu_{vX}(O_i), \mu_{vX}(O_j)) \\ s(\mu_{vY}(O_i), \mu_{vY}(O_j)) \end{bmatrix} \right) < \begin{bmatrix} d_{vX} \\ d_{vY} \end{bmatrix} \tag{3}$$

We evaluate the similarity measure s which performs the subtraction among velocity models for each object in O. Parameters d_{vX} and d_{vY} are constant values set to one pixel in accord with the previous established threshold for detecting moving features in KLT process. This value is chosen in order to conserve the best trade off between the threshold of moving points in KLT module and the expected bias among object velocities in the scene. This evaluation is carried out in a linked way, where merged groups are removed from O and added as a new object at the end of the list with, obviously, a new corresponding velocity model. This strategy allows the merging of the same objects previously detected with a more enriched model. Finally, even when both O_T and O_C object velocities are not resulted from the same process, they could be compared because both are based on pixel displacement in the scene (their optical flow).

5 MOVING OBJECTS TRACKED BY A KALMAN FILTER

Groups found by clustering technique are composed of point locations at different processing time. Hence, to confirm that moving object is still in the scene, only points present in last processed image are taken to initialize each cluster as a moving object in O_C. Therefore only the points located on the current image will be used to model the moving object.

To track moving objects in O_T along the image sequence, Kalman filter prediction evaluates a constant velocity model. To initialize this model a vector state is defined for each moving object detected. Vector state consists of the barycenter of object in X and Y and its velocities are set to μ_{vX} and μ_{vY} values, respectively. We assign the estimated position and velocity calculated by Kalman filter to baricenter of the object. This estimated position is used as the center of a window that will be extracted in the next image in order to search the object points. We called this window the zone of object and its size is a function of previous object limits and a security margin. Once object region is extracted in next image, we carried out a correlation process to find new object location.

5.1 Model Object Detection

The concept of object developed here covers the management of several points which have been evaluated as a part of a moving object. In this work, the task of tracking consists in following each object element with its particular appearance and the only relationship among them is their velocity and position. Thus, model object initialization consists in extracting a window patch (a template) around each point in the image where object was detected. The same number of templates are extracted around the estimated feature location in the next image. Appearance of initial templates in the current image is updated by an affine model. Feature points could be removed from the model if one or more of the following cases happen:

- Feature location is not found by correlation
- Location found is not inside the bounding box
- Displacement and velocity of the points found are not inside of the normally distribution of their respective mean data.

5.2 Occupancy Map

In our algorithm, it is important to add new points that complete the model of the detected moving object and at the same time detect new incoming objects. In a context of unknown environment, feature points should be initially chosen in all the image without highlighting some locations. To overcome this uniform point selection along robot navigation process, a space-time occupancy map is constructed by cells centered on tracked points indicating favorable locations for new feature detection. This map develops the idea of an occupation grid in the sense of higher probabilities represent the locations in the image that are no important to seek for a new point (like occupied locations). So, mainly locations in the image with lower probability values in the map will be first used to look for new points.

First probability map locations are initialized to $p_0 = 0.5$ value that represents the initial fair selection of feature points, that is, $p(u,v,t=0) = p_0$. When an interesting point is detected in (u,v) at $t > 0$, it becomes the center of an occupied cell and its corresponding value in the map is given by eq. 4 with $\alpha_t = 0$. A cell becomes empty in function of interest point displacements at every iteration : if a tracked point leaves the cell it was using at time $t-1$ its probability is updated and then this cell is labeled as empty in the current image t. The new cell in which tracked point is located at time t becomes now an occupied cell. The Figure 4 depicts simulated results of this map construction along five successive images for a tracked point. A square cell of size 11*x*11 centered on (u,v) point location is used. Cell value is always given by a 2D Gaussian pdf in combination with the rules established by eq. 4 and 5. The inverse effect of function $\alpha(u,v)$ applied to previous point positions is clearly seen in this image. With this probability map we enhance the assumption of new points of incoming object will appear behind its current detected point positions.

$$p(u,v,t) = \sum_{u,v \in Cell} (\alpha(u,v)_t + \mathcal{G}(\mu_u,\mu_v,\sigma_u,\sigma_v)) \quad (4)$$

where $\alpha(u,v)$ function describes the previous probability in the cell according to the previous state of location (u,v), that is:

$$\alpha(u,v)_t = \begin{cases} p_m - p(u,v)_{t-1} & \text{if } Cell_{u,v,t-1} \text{ occupied} \\ p_0 & \text{if } Cell_{u,v,t-1} \text{ empty} \\ 0 & \text{if } (u,v,)_{t-1} \text{ fair} \end{cases} \quad (5)$$

where p_m is the maximal probability value, set to 1 in our procedure. In order to avoid that this map stores for long time the same probability values in some locations, the map is reset each 2 times of trail process, except for the current object tracking locations.

6 EXPERIMENTAL RESULTS AND DISCUSSIONS

Proposed algorithms have been implemented in C, C++ and TCL and included as module into a framework for developping algorithms in robotics. Robot navigation was performed in indoor and outdoor context with a camera mounted on the robot (640 × 480 at 10Hz). The number N of tracked points by KLT is set to 150. At this first stage of the experiments, any information about the odometry of the robot is available, therefore, we carefully control robot speed navigation.

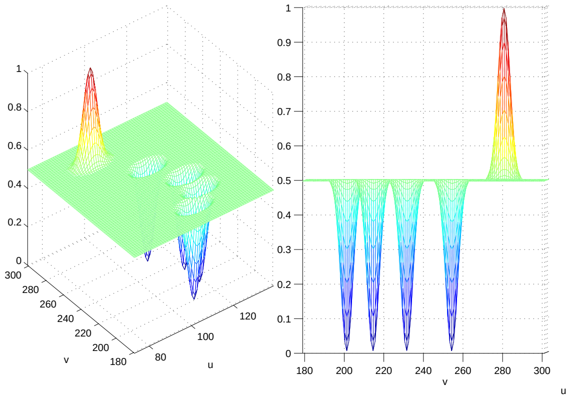
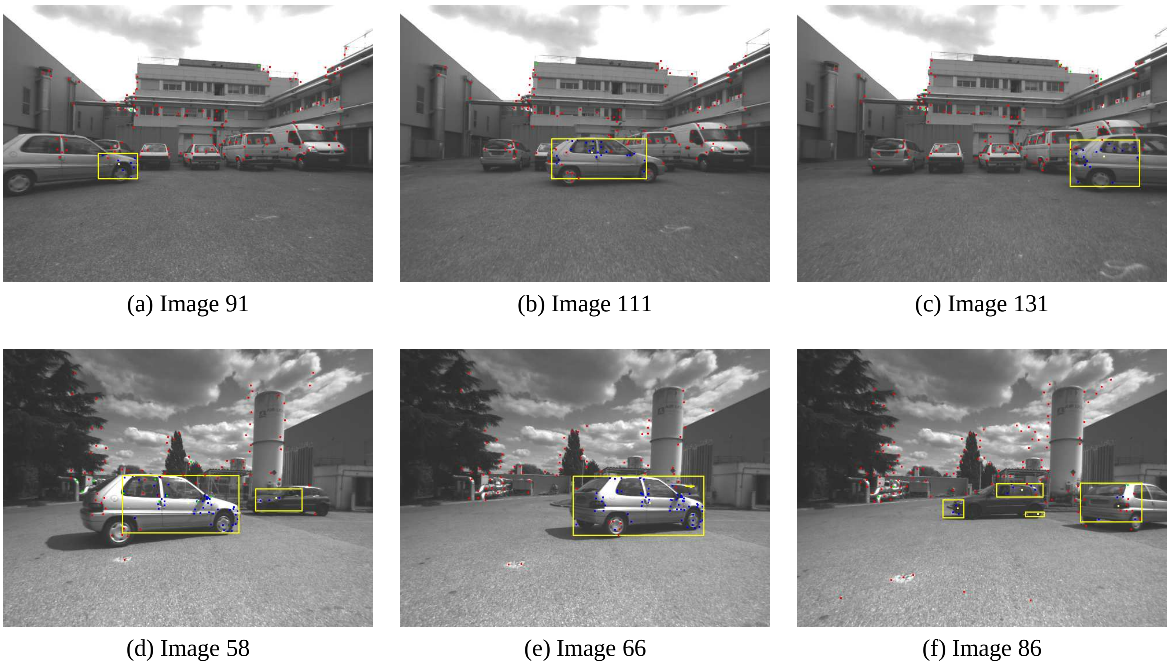


Figure 4: Simulated results of probability map for a tracked point in 5 successive images. Inverse shapes represent the probability in the four previous images, higher shape depicts current point location.

First test was carried out on a sequence of 300 images. The first row in Figure 5 shows three images selected from this sequence, at the time of a moving object appears in the field of vision of the robot. Image 5a shows the bounding box of moving object that enters from the left side of the image, all detected points inside of rectangle are used to initialize object model. After 20 processed images bounding box of initial object is enlarged (see image 5b). Even when in image 5c half of the moving object is out of the scene, the object is still tracked by the Kalman Filter. This is a specially difficult sequence because some images show drastic changes due to vibrations. So most of the points enters to the clustering process, but their movement is classified as random. The second row on Figure 5 shows the algorithm performance with object occlusions. In this case the car which is closer to the robot is well detected and tracked. Some problems to perfectly detect the second car after the occlusion occur and image 5d shows that the algorithm divides it in 3 different moving objects.

6.1 Non Rigid Moving Objects Tracking

An extension of tests to indoor environments is carried out for evaluating the development of our algorithm to track non rigid moving objects. Indoor environment induces more stable illumination conditions and normally a less charged background where non rigid objects could be better handle. The Figure 6


(a) Image 91 (b) Image 111 (c) Image 131

(d) Image 58 (e) Image 66 (f) Image 86

Figure 5: Experimental results in outdoor environment. First row shows moving object detection and tracking: (a) initial detection, then total detection in image (b) by merging function, finally tracking just the latest part of the car in (c). Second row : moving object occlusion is successfully handle by our proposed method.

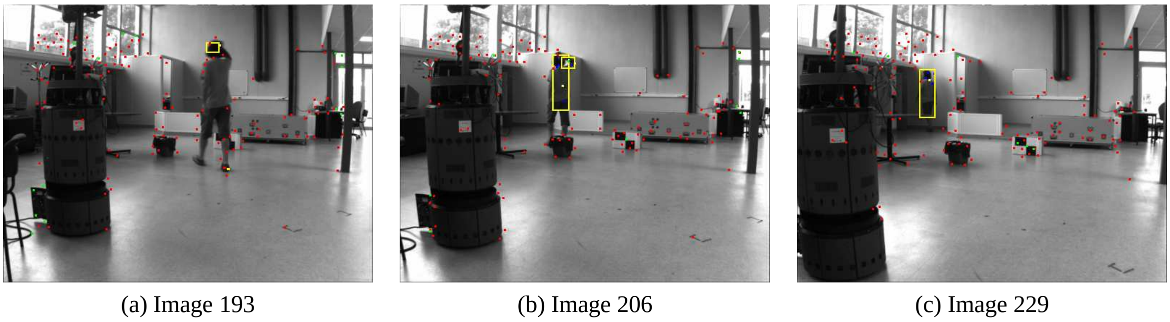

(a) Image 193 (b) Image 206 (c) Image 229

Figure 6: Indoor experimental results show the non-rigid objects detection and tracking. A good but incomplete detection is done for a person who crosses the room image (a). Then a more enhanced detection is done in images (b) and (c) however random natural turns during person walking perturb the successive detection of object points.

shows a person that crosses the field of view of the mobile robot during its navigation in an indoor context. As long as the person moves away of the robot, almost a complete detection is achieved. However, with a considerable delay, initially only the head of the person is detected. The leg movements are large and in a forward-backward direction and it disturbs the detection of a non random movement. To overcome this restriction in our algorithm, a strategy that involves the use of 3D camera and several point positions given by a SLAM process is being developed. So, an additional but fast analysis comparison between the depths of moving points and detected moving object will provide enough information to associate these points to moving object. The 2D dimensional context of our proposed method is preserved by transforming 3D points into 2D points before entering to clustering process and at the end taking into account their third coordinate position only for achieving a faster and more complete non rigid detection.

7 CONCLUSIONS

Experimental results show that even with few images, it is possible to detect a rigid (and partially non-rigid) moving object by a spatio-temporal analysis of features. Object model is enlarged thanks to prior knowledge managed by the proposed probability map. This map is successfully used during the active search of feature points because it mainly highlights zones that certainly contain new moving interesting points. Our tests are performed off line on a recorded sequence; however, the global algorithm works fast and could process images at 10Hz. The clustering method is the highest time consuming in the global process; for that reason, the number of trails to be grouped by the clustering method, should be no more to 150 points. Thus, the trade-off between image size and that number of points guarantees the highest performance in overall strategy.

It has been assumed that all pixels whose displacements are less than one pixel could be considered as noise or as points displaced by little vibrations of the camera. However the most sensible part of our algorithm resides in robot motion. Under not controlled conditions of velocity, most significant displacements are concentrated in both left and right image sides, mainly caused by egomotion. A general strategy to avoid egomotion detection and non rigid moving objects is being integrated based on monocamera SLAM approach. An interchange of 3D and 2D points information between SLAM and our MOT process will be continuously carried out giving a cooperative sense to our new proposed strategy. That is, detected static points will be sent to SLAM, these points are candidates to be included as a new landmark in the stochastic map used to update camera pose estimation. Then, this camera pose will be received by our MOT process to estimate the camera motion and calculate real detected point displacements.

ACKNOWLEDGEMENTS

This work has been performed in the context of the RINAVEC project funded by ANR, the french *Association Nationale de la Recherche*. It has been supported by the scholarship 183739 of the Consejo Nacional de Ciencia y Tecnología (CONACYT), the Secretaría de Educación Pública and by the mexican government.

REFERENCES

Cao, F., Delon, J., Desolneux, A., Musé, P., and Sur, F. (2007). A unified framework for detecting groups and application to shape recognition. *Journal of Mathematical Imaging and Vision*, 27(2):91–119.

Davison, A. (2003). Real-time simultaneous localisation and mapping with a single camera. In *Int. Conf. on Computer Vision*, pages 1403–1410.

Desolneux, A., Moisan, L., and Morel, J.-M. (2003). A grouping principle and four applications. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(4):508–513.

Desolneux, A., Moisan, L., and Morel, J.-M. (2008). *From Gestalt Theory to Image Analysis A Probabilistic Approach*, volume 34. Springer Berlin / Heidelberg.

Lookingbill, A., Lieb, D., and Thrun, S. (2007). *Autonomous Navigation in Dynamic Environments*, volume 35 of *Springer Tracts in Advanced Robotics*, pages 29–44. Springer Berlin / Heidelberg.

Lucas, B. D. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision (darpa). In *Proc. 1981 DARPA Image Understanding Workshop*, pages 121–130.

Poon, H. S., Mai, F., Hung, Y. S., and Chesi, G. (2009). Robust detection and tracking of multiple moving objects with 3d featu res by an uncalibrated monocular camera. In *Proc. 4th Int. Conf. on Com puter Vision/Computer Graphics CollaborationTechniques*, pages 140–149, Berlin, Heidelberg. Springer-Verlag.

Shi, J. and Tomasi, C. (1994). Good features to track. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition, 1994.*, pages 593–600.

Veit, T., Cao, F., and Bouthemy, P. (2007). Space-time a contrario clustering for detecting coherent motion. In *IEEE Int. Conf. on Robotics and Automation, ICRA'07*, pages 33–39, Roma, Italy.

Vu, T. V. and Aycard, O. (2009). Laser-based detection and tracking moving objects using data-driven markov chain monte carlo. In *IEEE Int. Conf. on Robotics Automation (ICRA), Kobe, Japan*.

Wang, C., Thorpe, C., Thrun, S., Hebert, M., and Durrant-Whyte, H. (2007). Simultaneous localization, mapping and moving object tracking. *Int. Journal of Robotics Research*.

SHORT PAPERS

AUTONOMOUS MANEUVERS OF A FARM VEHICLE WITH A TRAILED IMPLEMENT IN HEADLAND

Christophe Cariou, Roland Lenain, Michel Berducat
Cemagref, UR TSCF, 24 avenue des Landais, 63172 Aubière, France
{*christophe.cariou, roland.lenain, michel.berducat*}*@cemagref.fr*

Benoit Thuilot
Clermont Université, Université Blaise Pascal, LASMEA, BP 10448, 63000 Clermont-Ferrand, France
benoit.thuilot@lasmea.univ-bpclermont.fr

Keywords: Guidance system, Mobile robot, Path planning, Motion control, Agriculture.

Abstract: This paper addresses the problem of path generation and motion control for the autonomous maneuvers of a farm vehicle with a trailed implement in headland. A reverse turn planner is firstly investigated, based on primitives connected together. Then, both steering and speed control algorithms are considered. When the system is driving forward, the control algorithms are based on a kinematic model extended with additional sliding parameters and on model predictive control approaches. When the system is driving backward, two different steering controllers are proposed and compared. Real world experiments have been carried out with an experimental trailer hitched to a mobile robot. At the end of each row, the reverse turn is automatically generated to connect the next reference track, and the maneuvers are autonomously performed. Reported experiments demonstrate the capabilities of the proposed algorithms.

1 INTRODUCTION

For many years, researchers and manufacturers have widely pointed out the benefits of developing automatic guidance systems for agricultural vehicles, in particular to improve field efficiency while releasing human operator from monotonous and dangerous operations. Auto-steering systems are becoming common place (*e.g. Agco AutoGuide*, *Agrocom E-drive*, *Autofarm AutoSteer*, *Case IH AccuGuide*, *John-Deere AutoTrac*) and focus on accurately following parallel tracks in the field. However, more advanced functionalities are today required, in particular for headland driving. In fact, the operator must still manually perform maneuvers at the end of each row before reengaging the automatic guidance system on the next path to follow. In order to benefit of fully automated solutions, and therefore reduce the operator's workload (and even enable to consider driverless agricultural vehicles), the automation of the maneuvers in headland has to be studied with meticulous care.

Very few approaches have been proposed in that way, mainly based on loop turns (*e.g. John-Deere iTEC Pro*, see figure 1(a)). The drawback of such an approach is that it involves excessive headland width for turning on the adjacent track, all the more if a long trailer is used. It is thereby not adapted for small fields and far from optimal in term of productivity, headland being usually either low-yield field areas due to high soil compaction or wasted areas as they cannot be used for planting agricultural products.

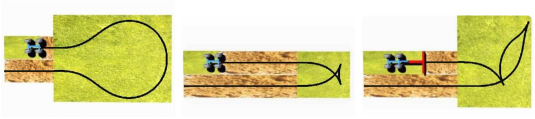

(a) Loop turn. (b) Fish-tail. (c) With trailer.

Figure 1: Different types of maneuver in headland.

Another solution is to perform reverse turns, i.e. maneuvers executed with stop points and a reverse motion as depicted in figures 1(b) and 1(c), leading to reduced headlands. However, although more in accordance with European agricultural practices, automation of such maneuvers in headland has rarely been considered in the literature. In fact, numerous approaches devoted to road applications have been proposed for the autonomous maneuvers of a vehicle, even with one or several trailers, see (Altafini C., 2001), (Lamiraux F., 1998), (Hermosillo J., 2003), but most of these control algorithms seem not well-adapted for an agricultural context. This paper proposes to address the automation of the maneuvers of a vehicle-trailer system, dedicated to an off-road context. This paper extends our previous work (Cariou et al., 2009) where accurate fish-tail maneuvers without trailer were autonomously performed. The general 1-trailer system is here considered, i.e. the trailer is hitched up at some distance from the middle point of the rear axle of the vehicle.

2 MOTION PLANNER

With regard to the path planning problem of agricultural machines in headland, generic optimal control algorithms are often investigated to find optimal point-to-point trajectories for a given cost function from a wide variety of configurations, see (Vougioukas et al., 2006). Primitive-based planning approaches are also widely used in the literature, relying either on clothoids, polynomial splines or cubic spirals to construct non-holonomic motions, see (Lau B., 2009). We have studied a similar approach well-adapted to agricultural maneuvers. It allows to rapidly obtain an efficient path planning solution for reverse turns of a vehicle-trailer system, based on elementary primitives (line segment, arc of circle) connected together with pieces of clothoid in order to ensure curvature continuity.

2.1 Arc of Clothoid

The curvature c of a clothoid varies linearly with respect to its curvilinear abscissa s ($c = g.s$), see figure 2. An arc of clothoid BP_1 admissible for the considered vehicle can then be defined in order to connect a line segment AB to a circle of radius R. To avoid the saturation of the steering actuator, R is chosen slightly upper than the minimum curvature radius of the vehicle. The proportionality coefficient g is computed according to the maximum vehicle front-wheel angular velocity ($\omega_a = 20°/s$) and of the vehicle linear velocity ($v_{ref} = 1.75m/s$) during the reverse turn, see (Cariou et al., 2009) for more details. In that way, continuous curvature trajectories admissible for the vehicle can easily be constructed.

2.2 Trajectory Generation Strategy

The aim is to connect two adjacent tracks AB and CD separated from a distance d, see figure 3(a). The proposed strategy considers the following motions.

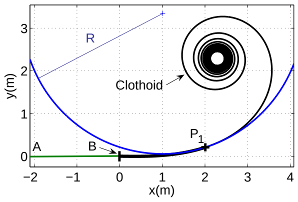
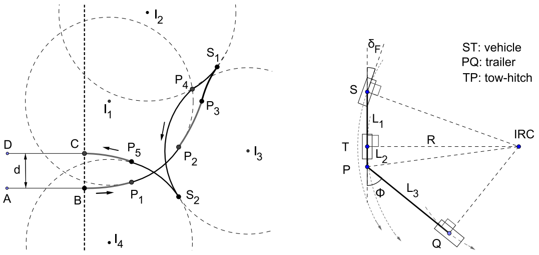


Figure 2: Clothoid, $g = 0.15$.



(a) Trajectories. (b) Rotation centers.

Figure 3: Path planning.

- The first movement from B to S_1 is composed of an arc of clothoid BP_1, to increase the curvature from $c = 0$ to $c = \frac{1}{R}$, an arc of circle P_1P_2 of center I_1 and of radius R, a second arc of clothoid P_2P_3 to decrease the curvature from $c = \frac{1}{R}$ to $c = 0$, and a part of a third arc of clothoid P_3S_1 required to align the trailer with the vehicle at the end of the movement. Aligning the vehicle-trailer system at the first stop point S_1 leads to a suitable configuration to plan the reverse motion. At S_1, the wheels are reorientated to change the vehicle instantaneous rotation center to I_2.
- The reverse movement is then built, composed firstly of an arc of circle S_1P_4 to increase the vehicle-trailer angle ϕ, see the notation in the bicycle representation figure 3(b). The point P_4 is determined in order that the vehicle-trailer system reaches the configuration shown in figure 3(b): the center of rotation of the trailer coincides with the center of rotation of the vehicle, when the vehicle front steering angle is set to a value $\delta_F^d = 20°$. Geometrical considerations show that the expected value ϕ_{ref} for the angle ϕ is:

$$\phi_{ref} = \pi - \arctan(\frac{R}{L_2}) - \arccos(\frac{L_3}{\sqrt{R^2 + L_2^2}}) \quad (1)$$

with: $R = L_1 / \tan \delta_F^d$

This is a stable configuration enabling a circular motion of radius R when pure rolling without sliding conditions are satisfied. It serves here as an objective configuration. At P_4, the wheels are reorientated to change the vehicle instantaneous rotation center from I_2 to I_3. Then, an arc of circle P_4S_2 of center I_3 and radius R is built.

- The third movement is composed of an arc of circle S_2P_5 of center I_4 and radius R, and an arc of clothoid P_5C to decrease the curvature from $c = \frac{1}{R}$ to $c = 0$. S_2 is the second stop point, defined as the intersection between the circles of center I_3 and I_4.

Figure 4 presents simulation results with two adjacent tracks separated from a distance $d = 2m$. At the first stop point S_1, the vehicle-trailer angle is $\phi = 0°$, i.e. the trailer and the vehicle are aligned, see the trajectories of points S, T and Q, respectively the centers of the vehicle front and rear axle and the center of the trailer axle. During the reverse motion, the vehicle-trailer angle reaches and maintains the expected configuration ϕ_{ref} (here $\phi_{ref} = 53°$).

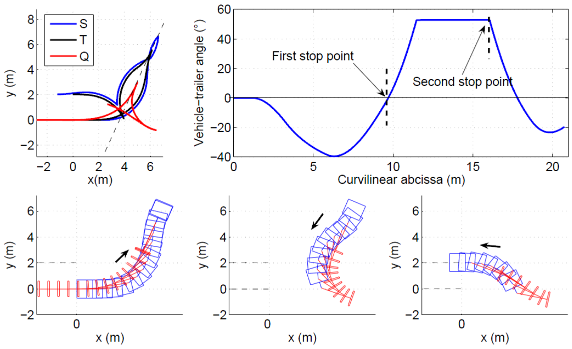


Figure 4: Path planning results.

3 CONTROL ALGORITHMS

3.1 Forward Motion

Accurate automatic guidance of mobile robots in an agricultural environment constitutes a challenging problem, mainly due to the low grip conditions usually met in such a context. In fact, as pointed out in (Wang D., 2006), if the control algorithms are designed from pure rolling without sliding assumptions, the accuracy of path tracking may be seriously damaged, especially in curves. Therefore, sliding has to be accounted in the control design to preserve the accuracy of path tracking, whatever the path to be followed and soil conditions.

3.1.1 Kinematic Model extended with Sliding Parameters

In the same way than in (Lenain et al., 2006a), two parameters homogeneous with sideslip angles in a dynamic model, are introduced to extend the classical kinematic model, see the bicycle representation of the vehicle in figure 5. These two angles, denoted β_F and β_R for the front and rear axle, represent the difference between the theoretical direction of the linear velocity vector at wheel centers, described by the wheel plane, and their actual direction. These angles are assumed to be entirely representative of sliding influence on vehicle dynamics. The notations used in this paper are listed below and depicted in figure 5.

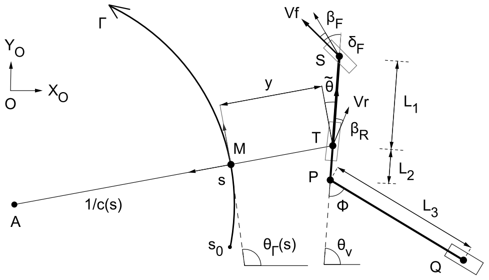


Figure 5: Path tracking parameters.

- S and T are the centers of the front and rear virtual wheel. T is the point to be controlled.
- θ_v is the orientation of vehicle centerline with respect to an absolute frame $[O, X_O, Y_O)$.
- δ_F is the front steering angle and constitutes the first control variable.
- V_r is the vehicle linear velocity at point T and constitutes the second control variable.
- β_F and β_R are the front and rear sideslip angles.
- M is the point on the reference path Γ to be followed, which is the closest to T.
- s is the curvilinear abscissa of point M along Γ.
- $c(s)$ is the curvature of the path Γ at point M.
- $\theta_\Gamma(s)$ is the orientation of the tangent to Γ at point M with respect to the absolute frame $[O, X_O, Y_O)$.
- $\tilde{\theta} = \theta_v - \theta_\Gamma$ is the vehicle angular deviation.
- y is the vehicle lateral deviation at point T.
- ϕ is the vehicle-trailer angle.
- $L_1 = 1.2m$ and $L_3 = 2.34m$ are respectively the vehicle and trailer wheelbases. $L_2 = 0.46m$ is the vehicle tow-hitch.

The equations of motion are derived with respect to the path Γ. It can be established, see (Lenain et al., 2006a), that:

$$\left\{\begin{array}{lcl} \dot{s} & = & V_r \frac{\cos(\tilde{\theta}-\beta_R)}{1-c(s)y} \\ \dot{y} & = & V_r \sin(\tilde{\theta}-\beta_R) \\ \dot{\tilde{\theta}} & = & V_r\left[\cos(\beta_R)\lambda_1 - \lambda_2\right] \\ \dot{\phi} & = & -\frac{V_r}{L_1L_3}\left[\tan\delta_F\left(L_2\cos\phi + L_3\right) + L_1\sin\phi\right] \end{array}\right. \tag{2}$$

with: $\lambda_1 = \frac{\tan(\delta_F - \beta_F) + \tan(\beta_R)}{L_1}$, $\lambda_2 = \frac{c(s)\cos(\tilde{\theta} - \beta_R)}{1 - c(s)y}$

Model (2) accurately describes the vehicle motion in presence of sliding as soon as the two additional parameters β_F and β_R are known. An observation algorithm has been developed to achieve indirect estimation, relying on the sole lateral and angular deviation measurements, see (Lenain et al., 2006b).

3.1.2 Control Law Design

The extended model (2) constitutes a relevant basis for mobile robot control design. As the vehicle-trailer system is well-known for being naturally exponentially stable when driving forward, the trailer is ignored in this case. In (Lenain et al., 2006a), the first three equations in model (2) have been converted in an exact way into linear equations, according to the state and control transformations:

$$\begin{array}{r} [s, y, \tilde{\theta}] \rightarrow [a_1, a_2, a_3] = [s, y, (1 - c(s)y)\tan(\tilde{\theta} + \beta_R)] \\ [V_r, \delta_F] \rightarrow [m_1, m_2] = [\frac{V_r \cos(\tilde{\theta} + \beta_R)}{1 - c(s)y}, \frac{da_3}{dt}] \end{array} \tag{3}$$

Finally, if derivatives are expressed with respect to the curvilinear abscissa, the following chained form is obtained:

$$\left\{ \begin{array}{lllll} a'_2 & = & \frac{da_2}{da_1} & = & a_3 \\ a'_3 & = & \frac{da_3}{da_1} & = & m_3 = \frac{m_2}{m_1} \end{array} \right. \tag{4}$$

Since chained form (4) is linear, a natural expression for the virtual control law m_3 is:

$$m_3 = -K_d a_3 - K_p a_2 \quad (K_p, K_d) \in \Re^{+2} \tag{5}$$

since it leads to:

$$a''_2 + K_d a'_2 + K_p a_2 = 0 \tag{6}$$

which implies that both a_2 and a_3 converge to zero, i.e. $y \rightarrow 0$ and $\tilde{\theta} \rightarrow \beta_R$. The inversion of control transformations provides the front steering control law:

$$\begin{array}{lll} \delta_F & = & \beta_F + \arctan\Big\{ - \tan(\beta_R) \\ & & + \frac{L_1}{\cos(\beta_R)} \big(\frac{c(s)\cos\tilde{\theta}_2}{\alpha} + \frac{A\cos^3\tilde{\theta}_2}{\alpha^2} \big) \Big\} \end{array} \tag{7}$$

$$\text{with:} \left\{ \begin{array}{lll} \tilde{\theta}_2 & = & \tilde{\theta} - \beta_R \\ \alpha & = & 1 - c(s)y \\ A & = & -K_p y - K_d \alpha \tan\tilde{\theta}_2 + c(s)\alpha\tan^2\tilde{\theta}_2 \end{array} \right.$$

In addition, as the actuation delays and vehicle inertia may lead to significant overshoots, especially at each beginning/end of curves, a predictive action must be added to the steering control in order to maintain accurate path tracking performances, see (Lenain et al., 2006b) for more details. A second control loop can be designed, dedicated to speed control. In (Cariou et al., 2009), a Model Predictive Control technique is used to anticipate speed variations and reject signifiant overshoots in longitudinal motion.

3.2 Backward Motion

It is well-known that steering backward a vehicle-trailer system has a tendency to jackknife, and requires special driving skill. In the literature, numerous path following approaches have addressed this problem when the trailer is hooked directly at the center of the rear axle of the vehicle. In contrast, the case of deported trailers has been rarely considered. In this paper, we propose to indirectly control the vehicle-trailer angle. Two control strategies are proposed.

3.2.1 Regulation of the Vehicle-trailer Angle

With the motion planner described in subsection 2.2, the previous steering control law (7) can be used during the backward motion until the vehicle-trailer system presents the expected angle ϕ_{ref}, corresponding to the configuration at point P_4 depicted in figure 3(b). Next, as the rest of the backward movement is quite short to reach the stop point S_2, the vehicle-trailer angle can be simply stabilized on ϕ_{ref}. More precisely, relying on the fourth equation in model (2), the error dynamic $\dot{\phi} = K_R(\phi_{ref} - \phi)$ ($K_R > 0$) can be imposed with the following front-wheel steering control law:

$$\delta_F = \arctan \frac{-L_1 \sin\phi - \frac{K_R L_1 L_3 (\phi_{ref} - \phi)}{V_r}}{L_2 \cos\phi + L_3} \tag{8}$$

3.2.2 Stabilization of the Trailer on the Planned Trajectory

Another solution is to control the center of the trailer axle on its respective planned trajectory, see the trajectory of Q during the backward motion in figure 4. For that, the trailer is first considered as an independent virtual vehicle, with a rear steering wheel located at the hitch point P and a fixed front-wheel located at point Q, see figure 6. The control objective can then be expressed as ensuring the convergence of this virtual vehicle (moving forward) to the reference path Γ. The first three equations in model (2) describe the motion of such a vehicle, except that δ_F has to be remplaced by $-\delta_c$, since the variations in the orientation of the vehicle are inverted when rear steering is considering instead of front steering. The state variables are now y_t and $\tilde{\theta} = \theta_t - \theta_\Gamma$, respectively the trailer lateral and angular deviation, see figure 6. A chained form transformation similar to the one presented in subsection 3.1 can then be applied, and the expected value for δ_c can be directly inferred from (7):

$$\delta_c = -\arctan\left\{L_3\left(\frac{c(s)\cos\tilde{\theta}}{\alpha_t} + \frac{A_t\cos^3\tilde{\theta}}{\alpha_t^2}\right)\right\} \tag{9}$$

$$\text{with:} \left\{\begin{array}{l} \alpha_t = 1 - c(s)y_t \\ A_t = -K_p y_t - K_d\,\alpha_t\tan\tilde{\theta} + c(s)\alpha_t\tan^2\tilde{\theta} \end{array}\right.$$

δ_c describes the desired direction of the linear velocity $\vec{v}$ at the hitch point. Then, the vehicle-trailer angle ϕ_{ref} ensuring that the center of rotation of the trailer coincides with the center of rotation of the vehicle can easily be inferred from δ_c via basic geometrical relations, see figure 6:

$$\phi_{ref} = \delta_c + \arcsin\frac{L_2\sin\delta_c}{L_3} \tag{10}$$

Finally, the proposed angle ϕ_{ref} can be controlled just as in subsection 3.2.1: reporting (10) into (8) provides the front steering control law for the vehicle.

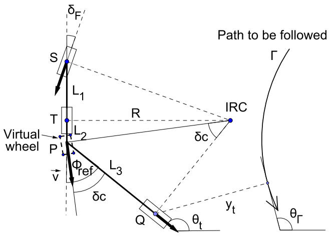


Figure 6: Trailer as a virtual vehicle.

4 EXPERIMENTAL RESULTS

In this section, the capabilities of the proposed control algorithms are investigated on an irregular natural terrain, using the experimental vehicle-trailer system depicted in figure 7. The vehicle is an all-terrain mobile robot whose weight is 650kg. The only exteroceptive sensor is an RTK-GPS receiver, whose antenna has been located straight up the center of the vehicle rear axle. It supplies an absolute position accurate to within 2cm, at a 10Hz sampling frequency. The vehicle-trailer angle is measured using a potentiometer. A gyrometer has also been added to obtain an accurate heading of the vehicle during the maneuvers.

In the forthcoming experimental test, the objective for the vehicle-trailer system is to follow autonomously two straight lines AB and CD separated from $2m$, see figure 8(a), and to execute autonomously the reverse turn using control law (8). The lateral deviation recorded at the center T of the rear wheels, according to the curvilinear abscissa, is reported in figure 8(b). At the beginning, the vehicle starts at about $25cm$ from the path to be followed. Then, it reaches the planned path and maintains an overall lateral error about $\pm 15cm$ during the maneuver. The main overshoots in the lateral deviation take place at the moment of a large deceleration and acceleration on a sliding ground (curvilinear abscissas $15m$ and $32m$). Despite such conditions, the lateral deviation remains inside $\pm 15cm$. This highlights the capabilities of the proposed algorithms, taking into account for sliding effects and actuator delays.

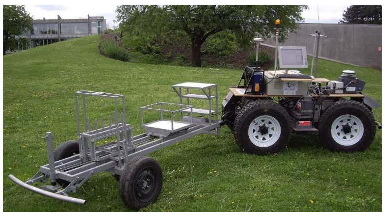
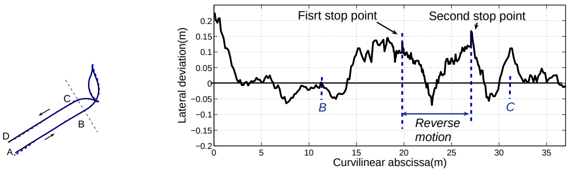


Figure 7: Experimental vehicle-trailer system.

The vehicle speed is reported at the top of figure 9. The speed reference $v_{ref} = 1.75m/s$ is correctly followed, and the speed variations are satisfactorily anticipated with the predictive approach. At the center of figure 9 is reported the vehicle front steering angle. It can be observed that its variations are quite smooth and that the wheels are satisfactorily reorientated to change the vehicle instantaneous rotation center at the stop points as well as at point P_4 (curvilinear abscissa $22m$). The vehicle-trailer angle w.r.t. the curvilinear abscissa is reported at the bottom of figure 9. In accordance with the simulations depicted in figure 4, this angle reaches $-40°$ during the first movement. The trailer and the vehicle are then aligned at the first stop point. During the reverse motion, this angle increases up to $\phi_{ref} = 53°$ (since here $\delta_F^d = 20°$). The angle is then correctly regulated to the value $\phi_{ref} = 53°$.



(a) Paths. (b) Lateral deviation at T (veh. rear axle).

Figure 8: Experimental results.

In the following test, the objective consists to validate the control law proposed in subsection 3.2.2 for the stabilization of the vehicle-trailer system on a planned trajectory. To this aim, a 80m-long reference path was first recorded during a preliminary run achieved in manual driving with the mobile robot moving forward. Then, the path is autonomously

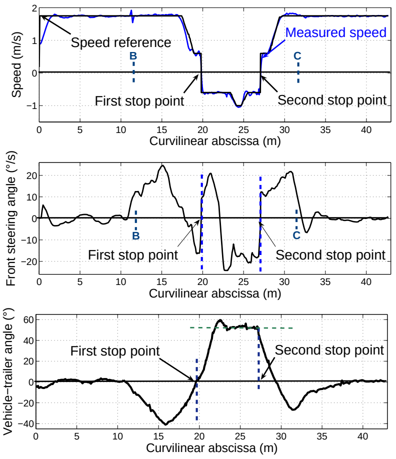


Figure 9: Speed, front steering angle, and ϕ.

followed backward at $0.5m/s$ with the vehicle-trailer system. At the beginning, the trailer starts with a lateral deviation of $1m$ from the path to be followed. Then, it reaches the planned path and maintains a satisfactory overall lateral error about $\pm 20cm$. Finally, the reverse turn maneuver depicted in figure 8(a) is performed with this control law. The lateral deviation, reported in figure 11, maintains an overall lateral error within $\pm 20cm$ during the maneuver.

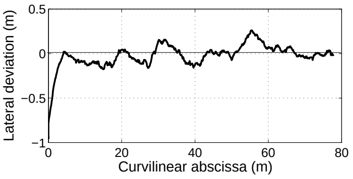


Figure 10: Reversing a vehicle-trailer system.

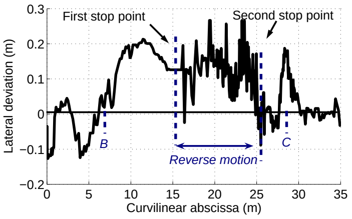


Figure 11: Lateral deviation at Q (trailer axle).

5 CONCLUSIONS

This paper addresses the problem of path generation and motion control for the autonomous maneuvers of a farm vehicle with a trailed implement in headland. A reverse turn planner has been first presented, based on primitives connected together with pieces of clothoid, in order to ensure curvature continuity. Next, control algorithms have been considered. When the system is driving forward, the control algorithms are based on a kinematic model extended with additional sliding parameters and on model predictive control approaches. When the system is driving backward, two different steering controllers have been presented. Promising results are reported with an off-road mobile robot pulling a trailer during reverse turn maneuvers. In spite of a fast speed and high steering variations, an overall tracking error within $\pm 20cm$ is obtained during the maneuver.

REFERENCES

Altafini C., Speranzon A., W. B. (2001). A feedback control scheme for reversing a truck and trailer vehicle. In *IEEE Transactions on robotics and automation, 17(6):915-922*.

Cariou, C., Lenain, R., Thuilot, B., and Martinet, P. (2009). Motion planner and lateral-longitudinal controllers for autonomous maneuvers of a farm vehicle in headland. In *IEEE Intelligent Robots and Systems, St Louis, USA*.

Hermosillo J., S. S. (2003). Feedback control of a bisteerable car using flatness: application to trajectory tracking. In *IEEE Int. American Control Conference, Denver, Co, USA, 4:3567-3572*.

Lamiraux F., L. J. (1998). A practical approach to feedback control for a mobile robot with trailer. In *IEEE Int. conf. on Robotics and Automation, Louvain, Belgique*.

Lau B., Sprunk C., B. W. (2009). Kinodynamic motion planning for mobile robots using splines. In *IEEE Intelligent Robots and Systems, St Louis, USA*.

Lenain, R., Thuilot, B., Cariou, C., and Martinet, P. (2006a). High accuracy path tracking for vehicles in presence of sliding: Application to farm vehicle automatic guidance for agricultural tasks. In *Autonomous Robots, 21(1):79-97*.

Lenain, R., Thuilot, B., Cariou, C., and Martinet, P. (2006b). Sideslip angles observer for vehicle guidance in sliding conditions: application to agricultural path tracking tasks. In *IEEE Int. conf. on Robotics and Automation, Orlando, USA*.

Vougioukas, S., Blackmore, S., Nielsen, J., and Fountas, S. (2006). A two-stage optimal motion planner for autonomous agricultural vehicles. In *Precision Agriculture, 7:361-377*.

Wang D., L. C. B. (2006). Modeling skidding and slipping in wheeled mobile robots: control design perspective. In *IEEE int. conf. on Intelligent Robots and Systems, Beijing, China*.

AUTOMATED 2D MEASURING OF INTERIORS USING A MOBILE PLATFORM

Alexander Fietz, Sebastian M. Jackisch, Benjamin A. Visel
Robert Bosch GmbH, Corporate Research, 71701 Schwieberdingen, Germany
{Alexander.Fietz, Sebastian.Jackisch, Benjamin.Visel}@de.bosch.com

Dieter Fritsch
Institute for Photogrammetry, Universitaet Stuttgart, 70174 Stuttgart, Germany
dieter.fritsch@ifp.uni-stuttgart.de

Keywords: Mobile robots, Surveying, Indoor environment, Exploration, Scan matching.

Abstract: This paper presents an approach for a fully automated measuring of a self-contained indoor scene in 2D using a mobile platform. A novel sensor is used, designed for the acquisition of precise floor plan scans, mounted on an ordinary mobile robot. The software framework is configured to optimize the accuracy and the completeness of the acquired data. Therefore an exploration strategy is presented that finds gaps in the data and determines the next best view in an explorative way. The scene is presented to the system as continuous line segments. Data management builds up the gathered point cloud in a homogeneous density, using the relevant and most accurate available information. Existing scan matching techniques are modified in a way to work robust and precise in the scenarios expected. The framework is tested in an exemplary indoor scene. Additionally, the scene is surveyed with a referencing system to build ground truth data and to enable an accuracy evaluation of the developed system.

1 INTRODUCTION

Architects, craftsmen but also further occupational groups engage in the constructional modification of interiors. Often the basis of their work is a detailed and accurate floor plan. The acquisition of the necessary geometrical information plus the following modeling still is a time-consuming procedure. Typically, the measuring instruments used therefore are tachymeters or just measuring tapes. Thus an automated floor plan creation is a topic that is worth investigating.

The intention of building a floor plan at the push of a button requires the utilization of a mobile robotic platform. This issue implicates a number of fundamental tasks that have to be overcome. The existence of a hardware framework to allow processing like the activation of sensors or the wheels is just a basic requirement. For an autonomous operation the system has to be able to locate itself in an unknown environment based on the available sensor data. Simultaneously one or even several maps have to be built up out of the collected data. The generation of a floor plan further used by the operator, but also a representation of the environment the robot can deal with. Normally these are associated with each other. The system has to know which parts of the environment still have to be surveyed and how it manages to do that. Therefore exploration strategies have to be applied that propose new measurement positions and determine paths to ensure a collision-free navigation.

The paper is organized as follows. Section 2 addresses the related and previous work, section 3 introduces the hardware framework of the used system. In sections 4&5 the scan matching technique and the exploration strategy are presented. Experiments and results are given in section 6, followed by section 7 that concludes the paper and refers to future work.

2 RELATED WORK

There has been a lot of research work in building two dimensional maps using robotic platforms. Thereby a broad variation of the presented approaches can be found, according to the already mentioned number of tasks a robotic system must accomplish. The adequate representation of the environment is one of the interesting subtopics. Depending on the systems major task an occupancy grid (Elfes, 1987), feature-based (Kuipers *et al.*, 1993) or topological (Choset & Burdick, 1996) maps can be the most appropriate choice. Occupancy grids have the nature of an easy handling, but need a lot of memory capacity. Thus their usage comes along with an increased computing time, especially in spacious applications. In comparison to that, feature-based and topological maps allow a rapid treatment, but depict just an abstraction of the scene.

Exploration strategies are based on these representation types. An early approach was given by Yamauchi in [1997]. He used a frontier-based approach that is based on an occupancy grid and accomplishes the exploration by classifying single grid cells. González-Baños & Latombe presented in [2002] a similar approach, that also observes the frontiers to the unknown areas, but directly evaluates the measured data points. Consecutive data points are connected to line segments and the obscurations inbetween are filled by modeling with geometrical primitives. The resulting line-like frontier models are built for every new scan and update the global model with a mergence. An alternative approach is given by Schmidt *et al.* in [2006]. The assumption, that the environment consists of simple rectangular geometries enables an abstraction of the collected data to geometrical primitives. Exploration is pushed by checking their topology and a sufficient observation of them. These strategies do not suit to the aspired application because they either primarily don't target the completeness of data or can't handle more complex indoor geometries. So an alternative method had to be found, that is presented in chapter 4.

Exploration applications require a simultaneous map building. Thereby, the well-known SLAM-problem, the bilateral dependancy of data registration and auto-localization, has to be considered to deal with the geometrically imprecise maps as well as noisy and ambigious sensor information. Various versions, how scan data can be matched to a map, were presented in the past years (Besl & McKay, 1992), (Lu und Milios, 1994), (Yaqub *et al.*, 2006). State of the art in global mapping is the usage of probalistic methods. For robot motion and data acquisition uncertainty models are generated at which Kalman or particle filters are applied. In (Thrun, 2002) a general review is given.

3 HARDWARE FRAMEWORK

The applied laser scanner is a prototype, designed for precise indoor measuring tasks. Within a measurement operation a point cloud of 3600 points can be recorded that yields a 360° view of the environment. The combination of multiple phase measurements allows distance measurements with a standard deviation of a few millimeters in a range of 7.5 meters. An integrated leveling unit adjusts the floor uneveness to make horizontal measurements possible. Signal analysis allows a prior filtering of erroneous data.
The sensor is mounted on a custom three-wheeled VolksBot platform provided by the Fraunhofer Institut (Fraunhofer, 2009). Data handling and sensor control is done with a notebook placed on the top of the platform. This set-up minimizes obscurations for the sensor. Used algorithms are implemented in Matlab (Figure 1).

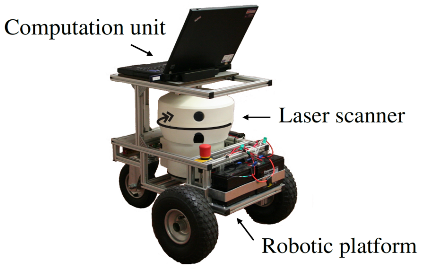


Figure 1: Hardware set-up.

4 GAP FILLING EXPLORATION

This section introduces an approach for the exploration of an indoor scene in order to provide data of the whole environment in sufficient density.

A: Representation of the Environment. The approach is based on the assumption that all occur-

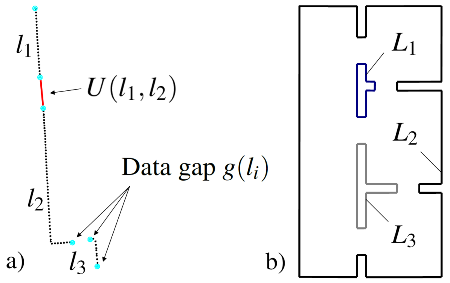


Figure 2: Representation of the environment: a) measured line fragments l_i, b) continuous line segments L_i.

ing surfaces can be described as continuous and self-contained line segments L_i (Figure 2b)). The environment is assumed as static. During the measuring procedure exploration is incomplete and actually searched segments L_i are just existent in fragments l_i that have to be connected (Figure 2a)). A segment ends, when no more neighboured points can be found within a certain distance d_{gap}. Since a phase measuring sensor is used, the layout of the measuring and exploration procedure has to be adapted accordingly. Data can be captured just from single selected points, while the mobile platform is resting. This causes a typical *next best view* problem.

For the definition of the area, accessible for the robot, an occupancy grid is used, composed of classified cell elements $o_{m,n}$ (Figure 5(a)). Spatial areas infused by the laser beams of the sensor can be classified as accessible, respectively as buffer area, if the affected cells are arranged close to obstacles or unexplored regions. This is necessary because the robot is regarded as a circular, moving item in the map. The same grid is used for path planning. If the next measurement position is determined, an A* Algorithm is used to find an unobstructed path consisting of passable grid cells to it.

The end points of l_i refer to gaps $g(l_i)_{1|2}$ in the logged data. Since these should be eliminated during exploration they are the basic information of the strategy. Existing approaches assume a knowledge of the assignment $U(l_i \leftrightarrow l_j)$ between single gap points building an unexplored area (González-Baños & Latombe, 2002). This is advantageous for the choice of explorative measuring positions, but leads to ambiguities if the scene consists of several line segments L_i. Hence, the presented approach tries to determine $U(l_i \leftrightarrow l_j)$, but isn't reliant on it.

B: Score Map. Potential measuring positions $p_{m,n}$ arise from the set of accessible grid cells $p \in M$ with $M = \{p_{m,n} \mid c_{m,n} \mathrel{\widehat{=}} \text{accessible}\}$. All candiate positions are evaluated due to their explorative suitability S_p as follows:

$$S_p = -f_1 \cdot k \cdot {D_{cp}}^2 - f_2 \sum_{i=1}^{k} {D_{pg}^{i}}^2 + f_3 \sum_{i=1}^{k} D_{gg}^{i} \qquad (1)$$

with k: number of visible gaps $\in p_{m,n}$

The candidate position with maximum value S_p is elected for next measurement position. To avoid unnecessary long distances between the single measurements, the euclidean distance D_{cp} between the current x_{cur} and the next position candidate x_{pos} is considered. Long distances lead to low evaluations. The same is true for the distance D_{pg} of x_{pos} to the visible data gaps (see section C). Since short distance measurements promise a better signal-to-noise ratio, those are prefered. Another influencing factor describes the importance of a data gap with regard to the progress of the exploration (see section D). Here, an assumption of gap assignments is utilized to determine the size of the unexplored area the gap could belong to. Partial influences are weighted individually with the factors f_1-f_3.

This results in a score map, that can be overlayed with the current occupancy grid. (Figure 5(c)) shows this exemplarily. Bright shaded grid cells indicate a good suitability for being the next measurement position.

In the course of the exploration uncloseable data gaps can occur, caused by unobservable surfaces or adverse geometrical situations. These must not hinder the exploration procedure. The system has to be able to recognize this circumstance to adapt its behavior appropriately. For that reason, data gaps are provided with counters. A gap, that couldn't be closed after a certain number of trials, is not tried to be observed any longer. That provides, that in repeatedly observed areas these data points are contained, which were captured from a shorter distance. Thus an inheritance of gaps must be intended for the case of a data exchange.

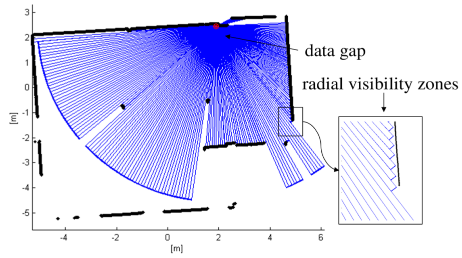


Figure 3: Visibility of gaps.

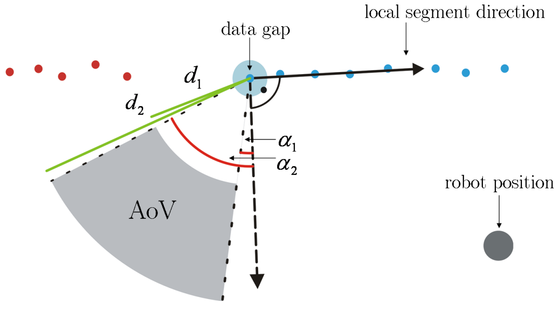


Figure 4: Area of visibility.

C: Area of Visibility. To ascertain the visibility of gaps, inverse visibility scans are performed. The environment of each gap is divided into radially extending zones. Data gaps are visible along each zone up to the point a line segment l_i intersects the zone or the sensors maximum range is reached (Figure 3).

Gap visibility is confined additionally. On the one hand, the point of view, from where gaps can be observed, should not have a too shallow angle of incidence regarding the surface normal, because of signal reflection. On the other hand measuring positions should be chosen, that imply a progress of global exploration. Therefore, a ring segment-shaped area of visibility $AoV = AoV(lsd, [\alpha_1, \alpha_2], [d_1, d_2])$ is definied, from where gaps can be observed (Figure 4). AoV is aligned to $\vec{n}$ the normal vector of lsd, the local segment direction of the gap . To avoid object observations on the back side, $\vec{n}$ points at the same side where points of l_i where measured from. To enable an explorative behavior, AoV is spanned in an area beyond $\vec{n}$ relative to the current robot position.

D: Assignment Estimation. Gap assignments $U(l_i \leftrightarrow l_j)$ are not used directly to deduce advantageous measuring positions. Nevertheless, it is tried to estimate potential connections. At least conclusions on the size of the unexplored region, the data gaps belong to, shall be retrieved to obtain an indicator for their importance with regard to the explorative benefit. So possible mismatches just lead to adverse ratings instead of fatal misbehaviour. The estimation is made using the hungarian method (Kuhn, 1955). Starting point is the filling up of the square cost matrix $C_{n\times n}$, that displays a complete bipartite graph. Assignment costs $c_{i,j}$ between the data gaps $g_{n,1|2}$ are given with their distances to each other. Distances of assigned gaps are integrated in equation 1 as D_{gg}.

$$c_{i,j} = \begin{cases} \mid g_i - g_j \mid, & if \;\; l(g_i) \neq l(g_j), \\ \infty & else. \end{cases} \tag{2}$$

So gaps $g_{n,1|2}$, belonging to the same line segment l_n, should not be assigned to each other. This assumption is valid as long as there are no almost completely explored line segments L_i, that could be closed. Then the probability of mismatches is increased. Figure 5(b) shows this exemplarily.

5 SCAN MATCHING

When the next measuring position is chosen the system has to go there. The scanner is just usable at standstill, so odometry is the only accessible data for motion control. In interiors its accuracy is sufficient for collision-free navigation, but not for a direct registration of collected data, in particular when highly precise models are to be derived. Hence a robust and precise scan matching is necessary for updating the current system position and to assure exact mapping. Registration of two-dimensional point clouds can be described as a rigid body transformation on the basis of 2 translative components T_x, T_y and a rotatory component α. So a point P'' can be transformed with:

$$\begin{pmatrix} P'_x \\ P'_y \end{pmatrix} = \begin{pmatrix} cos(\alpha) & -sin(\alpha) \\ sin(\alpha) & cos(\alpha) \end{pmatrix} \cdot \begin{pmatrix} P''_x \\ P''_y \end{pmatrix} + \begin{pmatrix} T_x \\ T_y \end{pmatrix} \tag{3}$$

To estimate the unknown transformation parameters $\Delta_{\hat{x}}$ an adjustment according to the *Gauß-Markov* model is proceeded:

$$\Delta_{\hat{x}} = (A^T GA)^{-1} A^T G \Delta_{per} = \begin{pmatrix} \Delta_\alpha \\ \Delta_{T_x} \\ \Delta_{T_y} \end{pmatrix} \tag{4}$$

A detailed description can be found in (Niemeier, 2002). Functional relationships f_n between scans have to be derived. For that perpendicular distances Δ_{per} between data points P''_n of a new scan D_{new} and their foot points F'_n on the straight line segments s, detected in the l_i of the existing data D_{exs}, are used. s are derived like in (Nguyen et *al.*, 2005). To derive f_n, Δ_{per} is expressed using equ.3:

$$\Delta_{per,n} = \mid P''_n - F'_n \mid \tag{5}$$

Matching should take place in a manner that $\sum \Delta^2_{per}$ is minimized. Since f_n are not linear functions, approximate values have to be found and the adjustment has to be executed iteratively.
For $A_{n,3}$, f_n have to be differentiated with respect to the unknown parameters x:

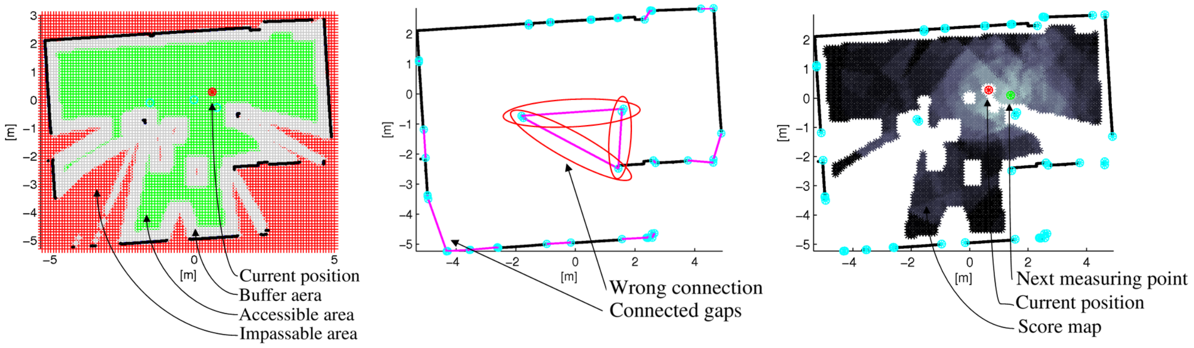
$$A_{n,3} = \begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ \vdots & \vdots & \vdots \\ a_{n,1} & a_{n,2} & a_{n,3} \end{pmatrix} \quad with: a_{n,u} = (\frac{\delta f_n}{\delta x_u}) \tag{6}$$

To give longer s a higher priority, since they are less affected by outliers and signal noise, every f_n is weighted on the trace of the weight matrix G, depending on their own length and the longest occurring straight line segment:

$$G_{(n,n)} = \frac{\mid s_n \mid}{\mid s_{max} \mid} \tag{7}$$

Two essential thresholds affect the matching. First, d_{line} for the definition of s, designating the maximum permitted distance of points to the best fit straight line. Second, d_{area} for the area on both sides of s, where points of D_{new} can be found for Δ_{per} (Figure 6). Points further afield are not considered.

One major difficulty in scan matching is the



(a) Occupancy grid.

(b) Gap assignment.

(c) Score map.

Figure 5: Score map determination.

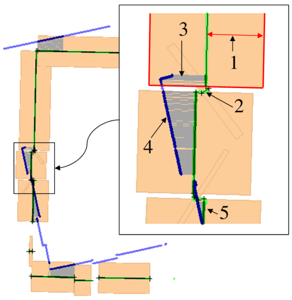


Figure 6: Scan matching with iterative stronger threshold restriction: 1) d_{area}, 2) straight line segment s, 3) perpendicular disances Δ_{per}, 4) new scan data D_{new}, 5) existing scan data D_{exs}.

choice of adequate thresholds. While weak thresholds entail a lower accuracy, a more strictly selection is at the expense of robustness. This led to the implementation of an iteratively exerted matching, where d_{line} & d_{area} are restricted increasingly by halving. This is iterated as long as there can be found enough line segments in D_{exs} with a certain minimum length and a sufficient variation in their orientation. The latter is necessary to avoid a degree of freedom in any spatial direction. In this way it is achieved that just straightest lines and the points in their closest neighbourhood are used for matching.

6 EXPERIMENTAL RESULTS

The system setup was tested in a interior scene of ca.150 m^2, consisting of several rooms and cleared out (Figure 7). Based on 93 measurements a point cloud of around 15k points was drawn up. Thereby the system covered a distance of just over 160 meters. Parameters and thresholds according to Table 1 were used.

Table 1: Parameter settings for test environment exploration.

Parameter	Value	Description
d_{area} (initial)	5cm	distance to s, where points for Δ_{per} are considered
d_{line} (initial)	2cm	maximum distance of points to the best fit straight line
f_1	1.3	weighting of the distances to $p_{m,n}$
f_2	1	weighting of the distances to $g(l_i)_{1\vert 2}$
f_3	0.6	weighting of gap assignment assumption
d_{gap}	2cm	minimal data gap distance $\widehat{=}$ model density
s_{grid}	0.1m	edge length of grid cells $o_{m,n}$
$v_{counter}$	3	maximum number of trials to eliminate a gap
α_1	5	angular intervall for area of visibility
α_2	60	angular intervall for area of visibility
d_1	0.3m	spacing intervall for area of visibility
d_2	7.5m	spacing intervall for area of visibility

In principal, exploration of all rooms was successful. Deficits can be observed with regard to the efficiency. One issue are seesaw changes between zones interesting for exploration, e.g. measuring positions 10, 11 & 12. Also notably is the high amount of measurements on the left side of the scene. This was caused by finely structured heatings, that were installed there. These led to many obscurations in the scanner data and thus to data gaps the system tries to close. To be able to conclude about the systems accuracy, ground truth data were recorded by generating a model of the same scene with a tachymeter[1]. Same points were tapped manually from the model the mobile system generated. Both models were overlayed to ascertain difference vectors. Coordinate differences possess a standard deviation of $s_0 = 5.2mm$ with maximal deviations of $1.2cm$ (Figure 7f)). As one can see, the alignment of vectors shows no significant systematic influence of errors. d_{area} & d_{line} in Table 1 and in the example of Figure5 differ. As Figure5 is for illustrative purpose, thresholds are set stricter in practise. A rough pre-matching step is made then to overlay the data already relatively accurate for higher robustness against errors. Therefore the algorithm of (Winkel-

[1] Leica TPS400 accuracy: -distance: $2mm + 2ppm$ -angle: $5''$

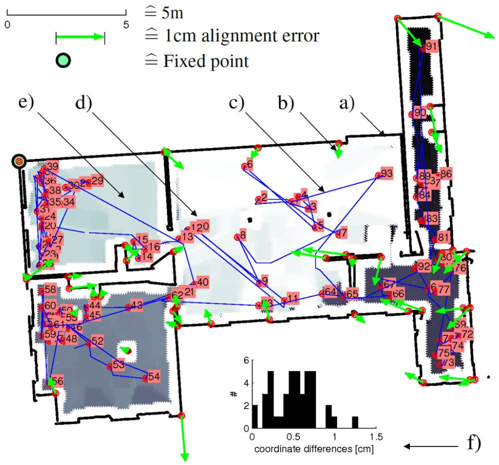


Figure 7: Test environment: a) model data, b) difference vectors to tachymeter model, c) path, d) succession of measuring positions, e) background, shaded dependend on the point of time, the area was declared as passable, f) histogram of differences to tachymeter model.

bach *et al.*, 2004) was adapted to the present conditions.

7 CONCLUSIONS

This work introduces an approach for automated exploration and mapping of interiors. Experiments showed that the system can deliver floor plan data of high accuracy. To our best knowledge no previous work on robotic 2D mapping could prove a comparable accuracy. This is attributable to the sensor, but also to the applied matching method. Mapping and localization is done via scan matching solely, in constrast to many current approaches. So spacious environments cause difficulties for the approach, since e.g. loop closing scenarios (Stachniss *et al.*, 2004) are not considered here. But the approach is demonstrably powerful as long as the interiors size is limited and there are enough geometrical landmarks existent to deliver constraints to the matching, what both is the case for small office environments. The exploration strategy presented allows a complete observation of the scene, but its efficiency is improvable. So future work will engage in implementation of higher-level strategies for exploration planning and control.

REFERENCES

Besl, P.J. and McKay, N.D. (1992). A method for registration of 3D shapes. *IEEE Transaction on Pattern Analyisis and Machine Intelligence 14, Nr. 2.* p. 239 – 256.

Choset, H. and Burdick, J. (1996). Sensor based motion planning: The hierarchical generalized voronoi graph. Ph.D. Thesis California Insitute of Technology, Passadena.

Elfes, A.(1987). Sonar-based real world mapping and navigation. *IEEE J. Robotics and Automation*, RA-3(3):249 – 265.

Fraunhofer IAIS Projekt: VolksBot. Domain: *http://www.volksbot.de/* (29.09.09).

González-Baños, H.H. and Latombe, J.C. (2002). Navigation strategies for exploring indoor environments. *Int. J. of Robotics Research.* 21(10–11):829–848.

Kuhn, H. W. (1955). The hungarian method for the assignment problem. *Naval Research Logistic Quarterly*, 2 83–97.

Kuipers, B., Froom, R., Lee, W.K. and Pierce, D. (1993). The semantic hierarchy in robot learning. In J. Connell and S. Mahadevan, editors, *Robot Learning*, Kluwer Academic Publishers, Boston, MA.

Lu, Feng; Milios, Evangelos (1994). Robot pose estimation in unknown environments by matching 2d range scans. *IEEE Computer Vision and Pattern Recognition Conference (CVPR).*

Niemeier, W. (2002). Ausgleichungsrechnung. *de Gruyter*, Berlin - New York, ISBN 3-11-014080-2.

Nguyen V., Martinelli A., Tomatis N. and Siegwart, R. (2005). A comparison of line extraction algorithms using 2D laser rangefinder for indoor mobile robotics. *Proc. Conference on Intelligent Robots and Systems, Aug 2005, Edmonton, Canada* pp. 1929–1934.

Schmidt, D., Luksch, T., Wettach, J. and Berns, K. (2006). Autonomous behavior-based exploration of office environments. *3rd International Conference on Informatics in Control, Automation and Robotics - ICINCO, Setubal, Portugal, August 1-5, 2006*, pp. 235-240.

Stachniss, C.,Haehnel, D. and Burgard, W. (2004). Exploration with active loop-closing for FastSLAM. In:*Proceeding of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems.* Sendai, Japan, pp.1505–1510.

Thrun, S. (2002). Robotic Mapping: A survey. Technical Report CMU-CS-02-111, School of Computer Science, Carnegie Mellon University, Pittsburg

Winkelbach, S., Rilk, M., Schoenfelder, C. and Wahl, F.M. (2004). Fast random sample matching of 3d fragments. *Pattern Recognition, 26th DAGM Symposium. Volume 3175 of Lecture Notes in Computer Science.*, Springer p. 129-136.

Yamauchi, B. (1997). Frontier-based approach for autonomous exploration. In: *Proceedings of the IEEE International Symposium on Computational Intelligence, Robotics and Automation*, pp. 146-151.

Yaqub, T., Tordon, M.J. and Katupitiya, J. (2006). Line segment based scan matching for concurrent mapping and localization of a mobile robot. In: *Proc. 9th International Conference on Control, Automation, Robotics and Vision*, Singapore, pp.1-6.

AUTOMATIC CALIBRATION OF A MOTION CAPTURE SYSTEM BASED ON INERTIAL SENSORS FOR TELE-MANIPULATION

Jörg Hoffmann
Institut für Informatik II, University of Bonn, Germany
hoffmann@cs.uni-bonn.de

Bernd Brüggemann
Fraunhofer Institute for Communication, Information Processing and Ergonomics (FKIE), Wachtberg, Germany
bernd.brueggemann@fkie.fraunhofer.de

Björn Krüger
Institut für Informatik II, University of Bonn, Germany
kruegerb@cs.uni-bonn.de

Keywords: Telerobotics and teleoperation, Motion capturing, Automatic calibration, Inertial sensors.

Abstract: We introduce an intuitive way of controlling a manipulator. This control should fulfill several constraints like low room consuming and operable in a short time. With this constraints in mind we chose an inertial sensor based motion capturing system. Such systems have to be adapted to the user. We present a method to adapt the system by a fast calibration while it is been used. The calibration does not require specified given motions. The soundness of the system is shown in synthetic and real experiments.

1 INTRODUCTION

Interactive manipulation of their environment is an essential task for robots. To provide a good tool for manipulation in most cases robots are equipped with a robot arm. Controlling such a manipulator is a challenging task, especially if it should be remotely controlled. The operator usually has an impression about the environment by video only. Based on this few information he has to perform complex motions with a manipulator. Controlling each joint separately demands long previous training and high concentration. The operator has to plan a sequence of motions for each joint in advance. The difficulty of these operations is raised by the fact that it may be necessary to re-adjust the joints during the motions.

As part of danger defence and disaster response exploring a potential dangerous area without exposing persons becomes more and more important. On one hand, threats by terrorist groups can cause great damage with relatively small effort. On the other hand, disasters in industrial areas may release dangerous and not easily detectable chemicals (TICs). If such an emergency occurs, information about what has happened and how the situation is developing is crucial for an effective response. Here robots can help to acquire desired information without exposing personnel to unknown risks. Such robots can be equipped with several sensors to identify different threats. But often it is just as important to manipulate the environment. This may occur e.g. if a suitcase has to be opened, or a sample has to be taken for analysis.

Nowadays most robots, which are meant to be used within a disaster area are at least partly remote controlled. So the operator has no direct line of sight to the robot and the area the robot is working in. Being reliant on only few sensors (like a camera) complex manipulations have to be executed. Often such tasks are time critical and far away from any available infrastructure. Therefore the needed control method for the robot arm has to fulfill several constraints: Intuitively usable, can be employed quickly, space saving and easy to transport, and pose only small restrictions to the mobility of the user

In this paper we present a method to steer the manipulator directly by the operator's movements without the need of an exoskeleton. We use five inertial sensors to capture the motions of the operator. The most important part in our system is the auto-calibration which makes an exact sensor placement unnecessary. In fact we only need to know roughly where the sensors are positioned.

The remainder of the paper is organized as follows: In section 2 related works are presented and different ways of remotely controlled manipulators are shown. In section 3 the problem and its constraints are described while section 4 shows the preprocessing of the sensor signals. The automatic calibration for our inertial sensor based motion capturing system is presented in section 5, followed by the robot arm control mechanism in section 6. Section 7 presents results of the calibration as well as its effect on the direct manipulation controls. The paper closes with our conclusions.

2 RELATED WORK

2.1 Robot Arm Control

Industrial robots are the prime example of the market's demand of manipulators. But also in mobile robotics the control of manipulators is essential. Here, in contrast to the industrial robots, the movement is not predefined and has to be adapted to the situation and the task. So, in the course of time several different methods for controlling a manipulator were developed. Roughly those methods can be divided into remotely controlled and autonomous movements.

Remotely controlled drives have the advantages to fall back on the decision of the operator. Although there is no need of autonomous decisions the system has to provide the operator with full control and information needed for the motion task. Within the remotely controlled manipulators, methods can be distinguished by their input devices, e.g. joysticks (some mentioned in (Laycock and Day, 2003)). From standard joysticks to those with force feedback, the boundaries to master-slave control mechanisms are fluid.

The idea to couple a control mechanism directly to the manipulator is rather old (e.g. (Goertz, 1954)). Such master-slave approaches have some advantages like being very accurate and the operator is always aware of the motions the manipulator will perform. Additionally, master-slave controls are able to provide the operator with haptic feedback. So there can be more resistance within the master device if the manipulator will be near to an obstacle or it is possible to feel the structure of a surface (e.g. (Tachi et al., 1990; Yokokohji and Yoshikawa, 1992)).

In-between master-slave devices and joysticks, there are exoskeletons. Such devices are worn by the operator. Exoskeletons are available for just a limb (e.g. an arm) or as whole body device (Bergamasco et al., 1994). Its advantages are a very precise reconstruction of the human motion and, compared to the master-slave approach, less space is needed. But due to the stiff construction, those exoskeletons prevent the operator from performing movements in a natural way.

Motion capturing is another approach to control a manipulator by reconstructing human movements. (e.g. (Miller et al., 2004; Pollard et al., 2002)). Approaches range from camera based motion capturing (with active or passive markers) to motion capturing with inertial sensors.

2.2 Motion Capturing with Inertial Sensors

Many techniques for capturing human motions have been developed in the last decades. Most of these techniques suffer from several disadvantages: they are not portable, need complex calibrations, or have disturbing exoskeletons. A good overview over the most important techniques is given in (Welch and Foxlin, 2002).

In the last years inertial sensor systems have become more popular for many applications. They are used for action recognition in games (Slyper and Hodgins, 2008), and for detection of typical or similar patterns in medical applications (Sabatini et al., 2005; Chang, 2006). A lot of work has been done on partial motion capturing, where only special parts of the body, as upper limbs for example, are considered. In this area applications like home rehabilitation systems (Zhou et al., 2006; Zhou and Hu, 2007; Tao et al., 2007) or robot controllers (Miller et al., 2004) have been developed. For tracking applications several systems based on fusion of inertial sensors and a variety of other systems have been introduced (Foxlin, 2005). For full body motion capturing a portabel system based on inertial sensors combined with an acoustic system was designed (Vlasic et al., 2007). Since Nintendo introduced the Wiimote in 2006 a lot of applications were developed (Schou and Gardner, 2007; Lee, 2008; Shiratori and Hodgins, 2008), while low cost sensors are available for mass market applications.

All these applications have to face similar technical difficulties. The data of inertial sensors are noisy and may contain a drifting. This problem is tackled using some basic techniques in all applications. Kalman filters are widely used for an optimal estimation based on data of multiple sensors (Vlasic et al., 2007; Zhou and Hu, 2007; Tao et al., 2007; Shiratori and Hodgins, 2008), especially when different

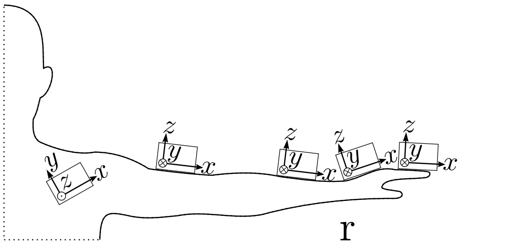


Figure 1: Positioning of the five inertial sensors on the right arm. Each joint is assigned to one sensor. The sensor on the shoulder is used as fix point.

types of sensors are combined. A Monte Carlo optimization can be used as basis for the estimation of positions and orientations (Zhou et al., 2006). For recognition of typical patterns in acceleration data classical signal processing techniques, like phase detection (Sabatini et al., 2005) or frequence analysis by Haar wavelet transformations (Slyper and Hodgins, 2008) have been employed. To estimate the direction of gravity several heuristics have been developed (Miller et al., 2004; Foxlin, 2005).

3 PROBLEM DESCRIPTION

To directly control the robot manipulator, the movement of the whole human arm has to be reconstructed. Therefore, we used four inertial sensors. One for each part of the arm: Upper arm, lower arm, hand, and fingers. A fifth sensor is needed to be positioned on the shoulder as a fix point (see Figure 1). The sensors provide their orientation but no information about their position in space. If the setup of the human arm and each sensor positioning on that arm is known, then it is possible to compute the arm configuration:

Let G_i be the missing position for the i-th joint, q_i the orientation measured by sensor i and s_i the length of the segment i (from joint i to $i+1$). $R_{q_i}(v)$ is the rotation of vector v around q_i. Then the desired positions of each joint are:

$$G_0 = [0,0,0]^T \tag{1}$$

$$G_i = G_{i-1} + R_{q_i}([s_i,0,0]^T) \tag{2}$$

Those equations needs the sensors to be oriented in direction of the bone it is connected to. Unfortunately the positioning of the sensors in reality is not as accurate as it is needed. So each sensor has a bias to the desired positioning. Figure 2 shows such a situation. Assuming the positioning is accurate, the orientations of the sensors imply a bended arm. When taken the faulty positioning α and β into account it can be recognized that the original arm configuration corresponds to a stretched arm. If the sensors are placed carefully on the human arm, the result will be alright but not very precise. But for a trustworthy and accurate result it is important to know the difference between the bone orientation and its corresponding sensor.

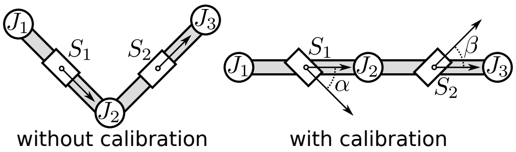


Figure 2: Imprecise positioning of sensors will cause false reconstruction of movements. Assuming the sensors are placed carefully the orientation implies a bended arm (left). When knowing that the sensors are placed faulty by an angle α and β the arm has to be reconstructed as streched (right).

When the system is calibrated, the reconstructed motions of the operator can be used to control the manipulator. As the used manipulator (and many robot arms currently available) has not the same morphologies as the human arm, it is not possible to map the reconstructed joint angles directly to the robot. So another mapping has to be found. We used a simple mapping by directly using the position and orientation of the operator's fingers. These informations are easy to compute with the calibrated arm model. As the finger's position is related to the shoulder, it can be used to provide a target for the manipulator's tool center point (TCP). This means that there will be no direct correlation between the pose of the human arm and the manipulator but a correlation between the position of the fingers and the TCP.

4 SIGNAL PREPROCESSING

We use two different types of sensor data provided by the inertial sensor system. First, we get orientation data which are used to perform forward kinematics on our arm model. Second, we obtain acceleration data which are used for the calibration. While the orientation data is reliable over long time periods the acceleration data is more noisy and is afflicted by drifting. In this section we show how we proceed with these artifacts in our sensor data.

4.1 Removing Noise

We have to remove noise on both sensor signals: The orientation data and the acceleration data. Both data types have a high frequency noise. To reduce this noise, we used a standard binomial low pass filter. As the velocity signal is a function in the position space, a filter with a window $w = 20$ data points is chosen.

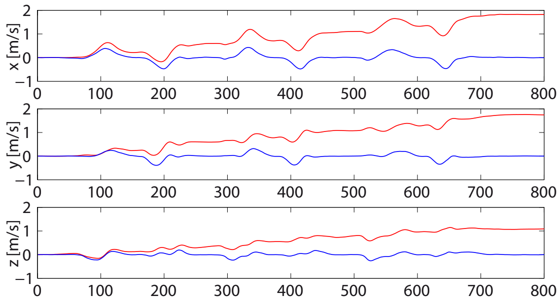


Figure 3: Velocity data for the three axis. Without zero velocity updates the data show a clear drift (red), with zero velocity updates the drift is visibly reduced (blue).

Since the orientations are represented by quaternions, we employ a smoothing filter to the orientation data like it is described by (Lee and Shin, 2002) in order to achieve reliable data.

4.2 Removing Drifting

The acceleration measurements include the gravitation. These so-called local accelerations have to be converted to global acceleration, i.e. accelerations without gravitation. When using the sensor orientation to remove acceleration caused by gravity, small errors in the orientation accumulate to large drifts in the rotational acceleration.

We can efficiently remove drifting by applying zero velocity updates (ZUPT) introduced by (Grejner-Brzezinska et al., 2001). Regardless whether we know the exact positions and orientations of our sensors: If the orientations do not change neither will the positions. We computed the velocity of each joint of our arm model. If these velocities are slower than a threshold (we use $0.05\frac{\mathrm{m}}{\mathrm{s}}$) we presume that the arm is not moving, i.e. its accelerations are zero. The measured non zero accelerations at these zero velocity points describe how the data have drifted. We use these information to correct the acceleration between two zero velocity points by changing the accelerations linearly. Some results of this method are given in Figure 3. Without zero velocity updates the data show a clear drift away from the zero line, with zero velocity updates the drift is visibly reduced. It shows that the sensor drifting can be removed clearly. Note that this technique is an offline strategy. However, this is no drawback, since we need the acceleration data only for the calibration of the sensors. For reconstruction we only use the orientation data, which do not have any drifting at all.

5 SENSOR CALIBRATION

The goal of our calibration is to detect the sensors' positions and orientations on the operator's arm automatically. Figure 2 shows the importance of this step, since a wrong bias may lead to faulty reconstructions of the arm motions. During our calibration step we use a model of the human arm. This model consists of four by spherical joints connected segments corresponding to: Upper arm, lower arm, hand, and fingers. We assume that the first joint of the model connects the upper arm to a world coordinate frame. This model contains also simulated sensors on each segment. Based on the models movement simulated data can be acquired. The calibration goes on as follows:

1. Recording of calibration motion
2. Signal preprocessing (see section 4)
3. Apply measured orientations to the model (move the model)
4. Derive simulated data from model
5. Compare simulated data with measured data
6. Change parameters of the arm model to minimize the error. Continue with step 3.

Step 3 to 6 are repeated until the resulting error is small enough. So the calibration can be expressed as a optimization problem, where the distance between the measured data and the data derived from the model is minimized:

$$\min_{\vec{t},\vec{q},\vec{s}} \mathcal{D}\left[D_{\text{sensors}}, D_{\text{model}}(O_{\text{sensors}}, \vec{t}, \vec{q}, \vec{s})\right], \tag{3}$$

where:

- $\mathcal{D}$ is some distance function
- D_{sensors} are data computed from the measured accelerations of the sensors
- D_{model} are data computed with the arm model which depends on:
  - O_{sensors} are measured orientation data
  - $\vec{t}$ position of each sensor on its corresponding segment
  - $\vec{q}$ orientation of each sensor with respect to its corresponding segment
  - $\vec{s}$ length of each segment

5.1 Optimization Problem

The optimization crucially depends on two different parameters. First the distance function $\mathcal{D}$. This function defines the similarity of two data sets D. As we have to compare data of different quality (i.e. position, orientation and length) different distance functions have to be considered. The actually used function $\mathcal{D}$ may consist of one or any combination of different distance functions. The adequate combination for our problem will be determined in the results. The three chosen basic distance functions are:

$$\mathcal{D}_{\mathrm{E}} = \sqrt{(D_{\mathrm{sensors}} - D_{\mathrm{model}})^2} \tag{4}$$

$$\mathcal{D}_{\cos} = \arccos\left(\left\langle \frac{D_{\mathrm{sensors}}}{||D_{\mathrm{sensors}}||}, \frac{D_{\mathrm{model}}}{||D_{\mathrm{model}}||} \right\rangle\right) \tag{5}$$

$$\mathcal{D}_{\mathrm{dir}} = ||D_{\mathrm{sensor}} - D_{\mathrm{model}}|| \tag{6}$$

The second important parameter of the optimization problem is the kind of data D we use. For the optimization we only use measured acceleration data and quantities deduced from the acceleration. Hence the usable features are:

- aG: local acceleration (measured data from sensors)
- a: global acceleration (without gravitation)
- v: velocity
- p: position

Although all the deduced features are computed from the local acceleration, there is a difference in quality. As mentioned above the global acceleration is computed using the measured orientation. Therefore, methods like zero velocity updates are applicable. The velocity data v are obtained from integrating a and are smoother which could result in a more stable optimization. By additional integration we obtain position data p, which leads to a further generalization. So D may be any combination of those features.

The goal is to get D_{sensors}, the measured data, and D_{model}, the results from the model with respect to the chosen distance function as similar as possible. The movements of the model are produced by orientation data O_{sensors}. The arm model is configured by the segments length, position and the orientation of each sensor. Depending on this configuration the forward kinematics of the arm induced by O_{sensors} results in a specific movement. This movement generates the data D_{model}. This constitutes a non-linear optimization problem which we solved by using the Levenberg-Marquardt algorithm.

5.2 Calibration Motion

The result of the calibration step depends on the underlying movement. There is no need for specific given motion that has to be performed. However, the calibration motion should cover movements in any degree of freedom, if possible. As the calibration movement is used to determine the arm parameters, a wide range of different motions will result in a well defined solution. Also it is expected that the optimization process will converge faster. In general to provide better results the calibration motion should include fast movements, pauses (for zero velocity updates) and movements in any possible direction. Note that the main part of the movements should be faster than the ZUPT threshold, otherwise the calibration might be unreliable. Experiments show that calibration movements of less than one minute are sufficient.

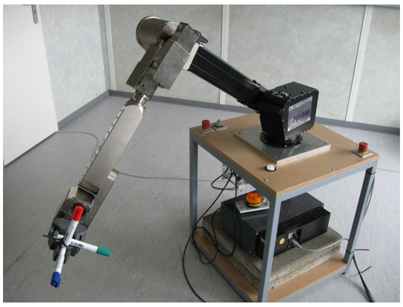


Figure 4: The used manipulator. Originally it is mounted on an EOD robot.

6 ROBOT ARM CONTROL

The manipulator used in our experiments is the robot arm normally mounted on the teleMAX robot of the manufacturer Telerob. The configuration is similar to a human arm, with two differences. First, it is equipped with a telescope joint. Second and more important, it is rotated by 90 degrees so that the shoulder does not turn vertically but horizontally (see Figure 4). The robot arm is equipped with a so called TCP control. With its help the tip of the gripper can be moved along and rotate around all axis. Every movement command is related to the point of origin in the shoulder joint. Using the inertial sensor based reconstruction of the human arm, the finger position can be determined and send to the manipulator. The goal of the presented control system is to give the operator a feeling for the movements of the manipulator. Therefore, the relation of human actions and robot reactions must be visible to the operator. If a movement is performed by the operator the manipulator has to react and imitate this movement. When designing the controler, one has to be kept in mind that the used manipulator is not able to perform movements as fast as humans.

To let the manipulator perform a similar trajectory as the operator, a list of end effector positions could be implemented. While the human performs the movement such a list is filled and the manipulator will move to the given end effector positions as fast as possible. We decided not to use such a technique because of the speed gap between operator and manipulator. Experiments showed that in our setup there is no feeling of being directly coupled to the manipulator. The manipulator will be more and more behind the operator's movements. Therefore, we decided to implement a direct transmission of the current operator's arm position. This provides the operator with a feeling of being directly connected to the manipulator and, in contrast to the list, it is possible to interrupt a not desired movement. But to define a trajectory the TCP has to follow, the operator has to move in the same speed as the robot does. Otherwise only target points are given, which the robot tries to reach on a linear way.

Fully stretched the manipulator has a range of about 1.50 m, more than a human arm. To achieve the full length of the manipulator, there must be a scaling function between the human motion and the manipulator movement. In experiments with different users, seemed to scaling the up-down direction (the z-axis) provided an unnatural feeling. Scaling in x- and y-axis seemed to be no problem to adapt to. So a scaling for x- and y-axis is used to allow the operator to use the whole operation range of the manipulator. We decided not to use any scaling for the z-axis.

7 RESULTS

7.1 Calibration of Synthetic Examples

To test the calibration and the used optimizer, we first used synthetic data. The synthetic data were produced by the arm model. Here for, the model is moved and the simulated sensors provide us with acceleration and orientation data. As stated in Section 5 the calibration mechanism has to find the correct positioning of each of the four sensors and the length of the segment corresponding.

To check whether the optimizer can find the correct configuration we performed several experiments. On the one hand, we tested situations where only one of the parameters is erroneous. On the other hand we have configurations where all three parameters were unknown. For example we tested the calibration of the system with sensors erroneous in orientation. So the model was build up with the sensors in orientation not equal to the segment's orientation. Movements were simulated and the sensor data computed. These sensor data were now used by the optimizer. It knows where the sensors are and how long each segment is. It does not know in which orientation the sensors are mounted. The experiment shows that the optimizer was able to determine the sensors' orientations. The described experiment was repeated with each parameter being erroneous and also with all three parameters erroneous.

	$\mathcal{D}_{\text{e}}, \mathcal{D}_{\text{dir}}$			$\mathcal{D}_{\text{e}}, \mathcal{D}_{\text{cos}}$			$\mathcal{D}_{\text{dir}}, \mathcal{D}_{\text{cos}}$			$\mathcal{D}_{\text{e}}, \mathcal{D}_{\text{dir}}, \mathcal{D}_{\text{cos}}$		
	E(tr)	E(ori)	E(len)	E(tr)	E(ori)	E(len)	E(tr)	E(ori)	E(len)	E(tr)	E(ori)	E(len)
a, aG	0	0	0.01	0	0	0.02	0	0	0	0	0	0.01
a, v	0.29	0.12	0.11	0.11	0.08	0.07	0.25	0.07	0.07	0.23	0.09	0.08
aG, v	0.42	0	0.07	0.42	0	0.07	0.43	0	0.09	0.42	0	0.08
a, aG, v	0	0	0.02	0	0	0	0.01	0	0	0.41	0	0.05
a, aG, v, p	0	0	0.01	0.43	0	0.07	0.04	0	0.02	0	0	0.02

Figure 5: Some example results for calibration of synthetic data with all three parameters faulty. E(tr) is the error of the positioning of the sensors, E(ori) the error of the orientation, E(len) the error of the segment length.

As the optimizer itself is also parameterized, the output depends on the chosen combination of distance functions and the features chosen (see Section 5). To check which combination of distance functions and features is adequate to solve the problem, we tested every possible combination on the synthetic data. In figure 5 some results are given for the experiments with all three parameters erroneous.

There are some combinations which are not suitable to use but it shows that there are several different combinations of metrics and features which enable the optimizer to find the original arm configuration. This shows that the method is able to find the correct configuration in general. The next step is to see whether the system is also stable with real data.

7.2 Calibration of Real Sensor Data

To test how the calibration step is effective to the motion reconstruction on real data a ground truth is necessary. We used an optical motion capturing system to compare the real movement of the operator with the uncalibrated and calibrated reconstruction from our system. In Figure 6 two example movements are shown. The red crosses show the finger positions (used as target position for the TCP of the manipulator) of the uncalibrated system. The green circles show the calibrated reconstruction. The blue dots represent the data from the optical system. In the left picture a wiping motion is shown. The right digram shows a gripping motion. In both examples the calibrated system performs more precise with respect to the ground truth.

To show the effect of calibration to the motion reconstruction we used a poorly positioned system to reconstruct a stretched arm. In Figure 7 the wrong

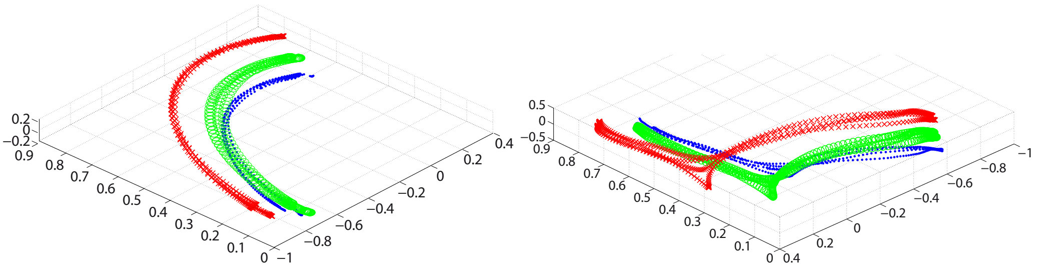


Figure 6: Comparison between an optical MoCap system as ground truth and our inertial sensor based system. The red crosses: finger positions of the uncalibrated system, green circles: calibrated reconstruction, blue dots: optical system.

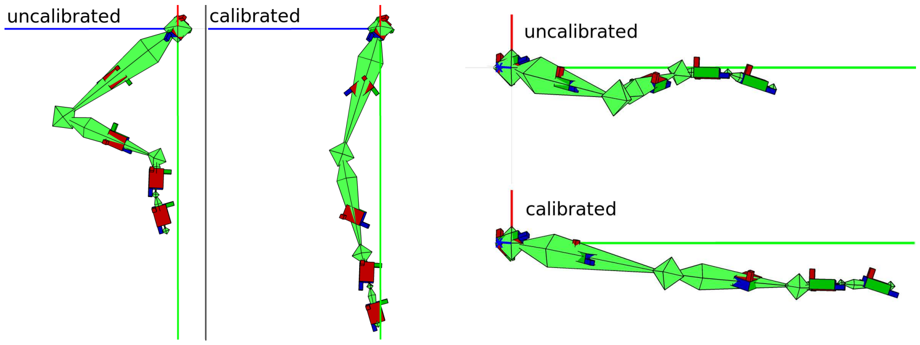


Figure 7: Arm model before and after calibration. The underlying pose is a stretched arm. As the sensors on upper and lower arm were not well orientated, the uncalibrated model returns a bended arm. After calibration the model shows are pose much more alike the operator's actual arm configuration., a stretched arm. Left picture shows view from above, right picture from the side.

reconstruction can be seen. This flexed position happens due to the fact that the model expects the sensors to be aligned to the orientations of the segments. After the calibration step the model is very close to the real pose, but with a small error in the bended elbow.

7.3 Control

To show that the presented method in general is adequate to control the manipulator we had different persons to perform several tasks. Those tasks were:

- Perform an infinity symbol
- Push a small log from the corner of a table
- Grip the log, (whereas the opening of the gripper is only slightly wider than the logs diameter)

Half of the users had never used this control method before. They had to perform each of the tasks twice. Once without any training, and once after some training. The results can be seen in Table 1 and 2.

Here you can see that, even totally untrained, most users were able to use the manipulator in the desired way. In fact the untrained test persons have a lower failure rate as the trained people. This may result from the ambitions of the trained people to be faster. This seems to make them more careless.

Table 1: Performance of untrained users with the control system.

task	successful	failure	average time [s]
infinity	8	0	37
Pushing	8	0	49
Gripping	7	1	52

Table 2: Performance of fairly trained users with the control system.

task	successful	failure	average time [s]
infinity	14	0	34
Pushing	13	1	31
Gripping	12	2	35

Often expressed criticism was the gap between the speed of the manipulator and the human arm. Most users needed some time to adapt to the much slower manipulator. So we expect better results when using a more agile manipulator.

8 CONCLUSIONS

In this paper we presented a control method for a robot arm. The main requirements to fulfill were an intuitive control, simple usage, and little space consuming. Our system based on inertial sensors to be worn on the operator's arm. With the help of an automatic calibration function, the exact placement of these sensors can be found, which makes an exact placement obsolete. Hence, equipping the system is rather simple and fast. The system is able to reconstruct the motion of the operator and therefore, send them to the manipulator. This results in a direct motion control where the user steers the TCP of the manipulator with his own movements. First experiments showed that even untrained persons can use the control system to fulfill certain tasks.

However, the system showed some points to improve, especially in its intuitive use. As it took a lot of time to get used to the fact, that the operator only steers the TCP but not the morphologies of the arm, a next step in development will be to map the reconstructed joint angles of the human arm to the joints of the manipulator to achive a similar manipulator configuration compared to the operator's pose. Also a continuous calibration could be useful. The calibration as it is designed showed no reason not to be used parallel to the steering task. This enables the system to react to changes e.g. a loose sensor or changes in the magnetic field. Here experiments are necessary to show if the system stays stable.

REFERENCES

Bergamasco, M., Allotta, B., Bosio, L., Ferretti, L., Parrini, G., Prisco, G., Salsedo, F., and Sartini, G. (1994). An arm exoskeleton system for teleoperation and virtual environments applications. In *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, pages 1449–1454 vol.2.

Chang, M. M. Y. (2006). Motion segmentation using inertial sensors. In *VRCIA '06: Proceedings of the 2006 ACM international conference on Virtual reality continuum and its applications*, pages 395–399, Hong Kong, China. ACM.

Foxlin, E. (2005). Pedestrian tracking with shoe-mounted inertial sensors. *IEEE Comput. Graph. Appl.*, 25(6):38–46.

Goertz, R. (1954). Mechanical master-slave manipulator. *Nucleonics*, 12(11):45–46.

Grejner-Brzezinska, D. A., Yi, Y., and Toth, C. K. (2001). Bridging gps gaps in urban canyons: Benefits of zupt. *Navigation Journal*, 48(4):217–225.

Laycock, S. and Day, A. (2003). Recent developments and applications of haptic devices. *Computer Graphics Forum*, 22(2):117–132(16).

Lee, J. and Shin, S. Y. (2002). General construction of time-domain filters for orientation data. *IEEE Transactions on Visualizatiuon and Computer Graphics*, 8(2):119–128.

Lee, J. C. (2008). Hacking the Nintendo Wii Remote. *IEEE Pervasive Computing*, 7(3):39–45.

Miller, N., Jenkins, O. C., Kallmann, M., and Matarić, M. J. (2004). Motion capture from inertial sensing for untethered humanoid teleoperation. In *Proceedings of the IEEE-RAS International Conference on Humanoid Robotics (Humanoids)*, Santa Monica, CA.

Pollard, N. S., Hodgins, J. K., Riley, M. J., and Atkeson, C. G. (2002). Adapting human motion for the control of a humanoid robot. In *Proceedings of International Conference on Robotics and Automation*, pages 1390–1397.

Sabatini, A., Martelloni, C., Scapellato, S., and Cavallo (2005). Assessment of walking features from foot inertial sensing. *Biomedical Engineering, IEEE Transactions on*, 52(3):486–494.

Schou, T. and Gardner, H. J. (2007). A wii remote, a game engine, five sensor bars and a virtual reality theatre. In *OZCHI '07: Proceedings of the 19th Australasian conference on Computer-Human Interaction*, pages 231–234, New York, NY, USA. ACM.

Shiratori, T. and Hodgins, J. K. (2008). Accelerometer-based user interfaces for the control of a physically simulated character. In *SIGGRAPH Asia '08: ACM SIGGRAPH Asia 2008 papers*, pages 1–9, Singapore. ACM.

Slyper, R. and Hodgins, J. K. (2008). Action capture with accelerometers. In *Proceedings of the 2008 ACM/Eurographics Symposium on Computer Animation*.

Tachi, S., Arai, H., and Maeda, T. (1990). Tele-existence master-slave system for remote manipulation. In *Intelligent Robots and Systems '90. 'Towards a New Frontier of Applications', Proceedings. IROS '90. IEEE International Workshop on*, pages 343–348 vol.1.

Tao, Y., Hu, H., and Zhou, H. (2007). Integration of vision and inertial sensors for 3D arm motion tracking in home-based rehabilitation. *The International Journal of Robotics Research*, 26(6):607–624.

Vlasic, D., Adelsberger, R., Vannucci, G., Barnwell, J., Gross, M., Matusik, W., and Popović, J. (2007). Practical motion capture in everyday surroundings. In *SIGGRAPH '07: ACM SIGGRAPH 2007 papers*, page 35, New York, NY, USA. ACM Press.

Welch, G. and Foxlin, E. (2002). Motion tracking: No silver bullet, but a respectable arsenal. *IEEE Comput. Graph. Appl.*, 22(6):24–38.

Yokokohji, Y. and Yoshikawa, T. (1992). Bilateral control of master-slave manipulators for ideal kinesthetic coupling-formulation and experiment. In *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on*, pages 849–858 vol.1.

Zhou, H. and Hu, H. (2007). Upper limb motion estimation from inertial measurements. *International Journal of Information Technology*, 13(1).

Zhou, H., Hu, H., and Tao, Y. (2006). Inertial measurements of upper limb motion. *Medical and Biological Engineering and Computing*, 44(6):479–487.

LEGS DETECTION USING A LASER RANGE FINDER FOR HUMAN ROBOT INTERACTION

Flávio Garcia Pereira, Raquel Frizera Vassallo and Evandro Ottoni Teatini Salles
Department of Electrical Engineering, Federal University of Espírito Santo
Av. Fernando Ferrari, 514, Goiabeiras, Vitória-ES, CEP 29075-910, Brazil
{flaviog, raquel, evandro}@ele.ufes.br

Keywords: Human-robot interaction, Legs detection, Laser scanner.

Abstract: The ability to detect humans is an important skill for service robots, especially if these robots are employed in an environment where human presence is constant, for instance a service robot which works as a receptionist in the hall of a hotel. The principal aim of the proposed method is to estimate the human position using data provided by a Laser Range Finder (LRF). The method utilizes two Finite State Machines (FSMs) to detect some leg patterns and, after that, it computes the probability of being a pair of legs for each detected pattern. In order to validate the proposed method some experiments were performed and are shown.

1 INTRODUCTION

In recent years, great effort has been done in order to improve Human-Robot Interaction (HRI) research field. Researchers such as Bellotto and Hu (Bellotto and Hu, 2009) claim that the studies of HRI are currently some of the most fascinating research field in mobile robotics, and Bekey (Bekey, 2005) emphasizes the idea that cooperation and interaction among men and robots are the big challenges of the next years. Because that, the robots which will interact with humans must have the skill to detect people. This ability will enable robots to understand better and anticipate human intentions and actions (Arras et al., 2007). The main purpose of our work is to develop a laser-based human detection in order to allow a mobile robot to interact with people.

There are many researches concerning people detection using laser scanners. The work done by (Carballo et al., 2009) introduces a method for people detection around a mobile robot using two layers of laser scanners, thus two sets of features for each person are detected. Based on these features and a previous knowledge about human body shape, the human detections is performed. In (Fod et al., 2002), the authors present a technique to track moving objects in a workspace covered by multiple lasers. The method to detect people shown in (Arras et al., 2007), uses a supervised learning technique to create a classifier that facilitates such detection. The classifier is trained using AdaBoost method. A way to detect line and circles from laser data in an indoor environment is introduced by (Xavier et al., 2005). The authors still perform leg detection by considering it as a circle with some particularities like the diameter of the circle. The approach to track multiple moving objects shown in (Schulz et al., 2001) uses laser data and combines particle filters with existing approaches to multi-target tracking. The system uses leg detection and occupancy grids to detect people. Topp and Henrik (Topp and Christensen, 2005) introduces the "Human Augmented Mapping" which represents an integration of automated map generation with learning of environmental concepts. They propose a method similar to the one presented in (Schulz et al., 2001) with the difference that their method allows handling people standing still, which is useful for interaction.

The authors of (Cui et al., 2005) present a system that employs multiple laser scanner and one camera to track multiple persons. They track people through a meanshift method and laser tracking and fuse these two information using a Bayesian formulation. The work presented in (Müller et al., 2007) implements a fusion of laser, sonar and vision data to find and track people by a mobile shopping assistance robot. A system to track people in real time in uncontrolled environments is presented in (Scheutz et al., 2004). This system combines leg detection based on laser data and face detection implemented in the specialized hardware *cellular neural network* (CNN) universal machine. Reference (Bellotto and Hu, 2009) presents a multisensor data fusion techniques for tracking people by a mobile robot using a laser scanner and one monocular camera. They extract the features from a laser scanner and look for some leg patterns. Vision is used for face detection and the human tracking is performed by fusing the two different sensor data. Luo *et al.* (Luo et al., 2007) describe a method to find and track a human using a monocular camera, which is responsible for finding the human face, and a laser scanner assembled on their robot, whose function is to find human body and arms. The data sensor is fused by statistical independence. In (Kleinehagenbrock et al., 2002) a hybrid method for integrating laser range and vision data is presented. They use the laser data to detect human legs and colored images to find skin color and face. These information are fused to better perform the human tracking.

The approach we present in this paper is focused in detecting human legs using a laser scanner and to determine their position. If the legs detection is positive, the robot starts interacting with the human. Our method to detect legs is similar to the approach developed in (Bellotto and Hu, 2009). Their system finds human legs after identifying some legs patterns that correspond to legs apart, forward straddle and legs together. In our case, only two patterns are considered legs apart and legs together. In order to find these two patterns we implemented a Finite State Machine (FSM) and for each pattern found we calculate the probability of being a pair of legs. The advantages of the proposed method are the low computational cost (the implemented method performs each detection in approximately 35*ms*), the simplicity (it only uses two FSMs and a probability function to classify a pair of legs) and the low quantity of parameters that need to be estimated (the distance between the legs and the difference between their widths).

The remaining of this paper is organized as follows. Section 2 presents the method to find human legs using a laser scanner. Section 3 illustrates the results of some performed experiments and, at last, the conclusions and future work are presented in Section 4.

2 LEGS DETECTION

The method to detect legs that we are going to present extracts features from a laser scanner and, as the method shown in (Bellotto and Hu, 2009), identifies patterns relative to the legs posture. These patterns correspond to the following situations: legs apart (LA) and two legs together (LT). The structure of the legs detection algorithm can be seen in Figure 1.

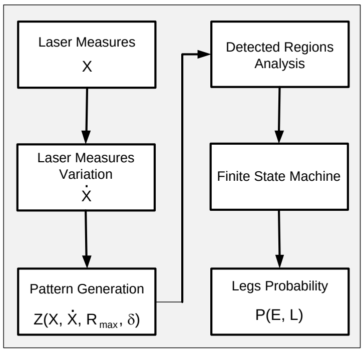
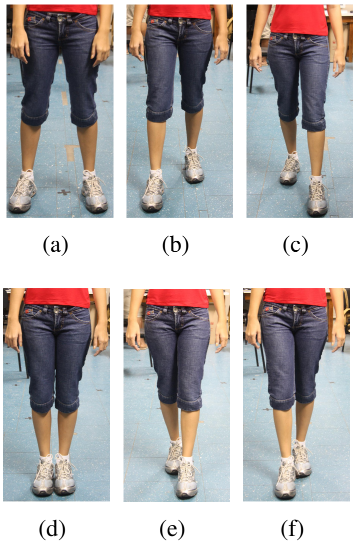


Figure 1: Structure of the proposed algorithm.

In order to determine the most common leg position when a person stops and talks to another, some people were observed. Figure 2 illustrates these situations.


(a) (b) (c)

(d) (e) (f)

Figure 2: Legs position. Legs apart (a)-(c). Legs together (d)-(f).

2.1 Transitions Array

The distance measures provided by the laser scanner are stored in an array $X = [x_1, x_2, \ldots, x_N]$ where x_i is each distance measure captured and N is the total number of readings. After that, an array with the difference between two consecutive measures ($\dot{X} = [\dot{x}_1, \dot{x}_2, \ldots, \dot{x}_{N-1}]$) is calculated as $\dot{x}_i = x_{i+1} - x_i$, with $i = 1, 2, \ldots, N-1$. Then, the array $Z = [z_1, z_2, \ldots, z_{N-1}]$, which stores the transitions related with each measure in x_i, is built. Z is created based on X, $\dot{X}$, R_{max} and δ, where R_{max} is the maximum distance we are considering for the measures done by the laser scanner (2*m* in this case) and δ is a distance threshold. We define five different transitions:

- Transition 0: $|\dot{x}_i| < \delta$ and $x_i = R_{max}$;

- Transition 1: $|\dot{x}_i| > \delta$ and $\dot{x}_i < 0$;
- Transition 2: $|\dot{x}_i| < \delta$ and $x_i \neq R_{max}$;
- Transition 3: $|\dot{x}_i| > \delta$, $\dot{x}_i > 0$ and $x_i \neq R_{max}$;
- Transition 4: $|\dot{x}_i| > \delta$, $\dot{x}_i > 0$ and $x_i = R_{max}$.

Figure 3 illustrates these five transitions. The dashed line indicates the maximum measure, i. e., $2m$.

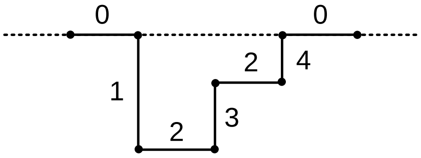


Figure 3: Defined transitions.

2.2 The Finite State Machines

After generating the array Z, it is performed a search for leg candidates (LCs). These candidates are the regions in Z that starts with a Transition 1 preceded by a Transition 0 and finish with a Transition 4. Each can be pre-classified either as single leg or as pair of legs together candidate as follows. A LC is pre-classified as a single leg if the distance between the extremities is in the range $(5cm, 15cm)$. If this distance is bigger than $15cm$ and smaller than $32cm$ the LC is pre-classified as a pair of legs together. Otherwise the LC is discarded.

Once this pre-classification is accomplished, the LCs are verified by two FSMs. One of them looks for the LA pattern and the other for the LT. Each state of these machines receives as input a value between 0 and 4, which represents the transitions already mentioned. A LA pattern is defined as a sequence 012401240 (Figure 4(a)-(c)), where the numbers represent the respective transitions. The LT pattern can assume three different sequences: 0121240, 01240 and 0123240 (Figure 4(d)-(f)).

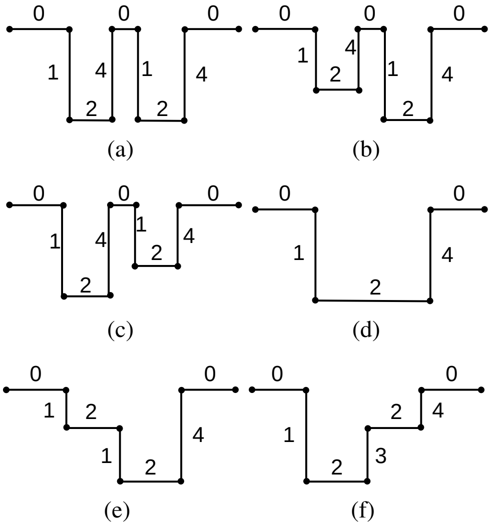
The patterns shown in Figure 4 correspond to the legs position that appear in Figure 2.

Figures 5 (a) and (b) show, respectively, the FSMs for detecting the LA and LT patterns. To simplify, the inputs that take the FSMs to an invalid state are not drawn in the schematics shown in those figures.

Notice that the numbers close to the arrows that link the states are in the form input/output, where the inputs are the values of the transitions in the vector Z and the output can be either 0 (the pattern was not identified) or 1 (recognized pattern).

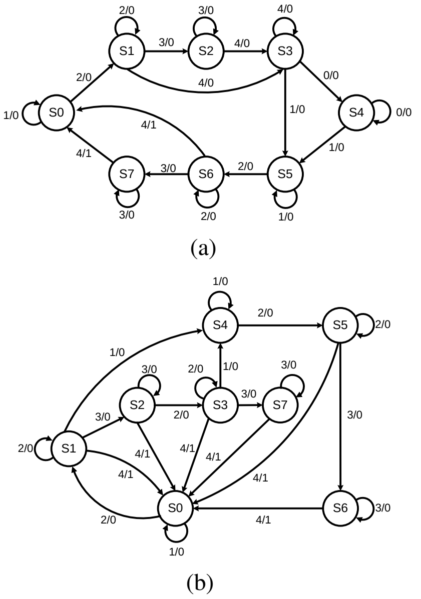
2.3 Legs Probability

Once the detected patterns are classified by the FSMs, some characteristics of the pair of legs are extracted in order to determine a probability of being legs, such as the distance between the extremities (E) and the difference between the width of each detected leg (L). This probability was introduced to avoid situations where, for example, there are two people near the robot, one of them at the left extremity and the other at the right extremity and, moreover, only one leg of each one is detected. In this situation, the FSMs will classify these legs as a pair of separate legs, but since they are far from each other, they have low probability of being a pair of legs which belongs to the same person. Moreover, if a person stands still in front of



(a) (b) (c) (d) (e) (f)

Figure 4: Leg patterns. Legs apart (a)-(c). Legs together (d)-(f).



(a)

(b)

Figure 5: Finite State Machines. Legs apart (a) and legs together (b).

the robot with legs apart and using a walking stick, it could be interpreted as a leg. However, as the walking stick is thinner than a leg, it will not happen due to the probability of being legs. This probability is calculated as,

$$P(E,L) = \left[1 - \tanh\left((\kappa L)^3\right)\right] \exp\left(-\frac{(E-\bar{E})^2}{2\sigma^2}\right), \quad (1)$$

where L is the difference between the width of each leg, E is the distance of the exterior extremity of two consecutive legs and κ is a positive constant. The average distance of human legs when they stop in front of the laser and the standard deviation are given by $\bar{E}$ and σ, respectively. Figure 6 shows the probability of a pattern to be a pair of legs apart according to variables E and L. For the situation where a pattern is classified as a pair of apart legs by the FSMs, the parameters of Eq. 1 are $\bar{E} = 0.3037$m, $\sigma = 0.028$m, otherwise the values are $\bar{E} = 0.25$m, $\sigma = 0.0268$m. The adopted range for the variable E is $[0.15m,\ 0.45m]$ and for the variable L is $[0,\ 0.1m]$. The value of κ does not change no matter the situation and it is $\kappa = 15$. The graphic of being a pair of legs together is similar to the legs apart and is not shown.

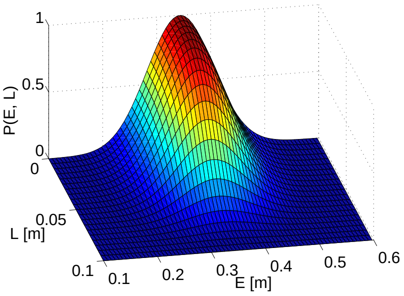


Figure 6: Probability of being a pair of legs apart.

By observing the graphic presented in Figure 6, it can be seen that the probability assumes its maximum value when L is zero and E is equal to $\bar{E}$.

3 EXPERIMENTAL RESULTS

The experiments were performed in an indoor environment using a mobile robot Pioneer 3-AT from ActivMedia, equipped with a laser scanner Sick LMS200. Even though the laser scanner provides distance measures from 0° (right side of the robot) to 180° (left side of the robot), the experiments were performed using the measures from 60° to 120°, because the aim is to detect people who are interested in interacting with the robot. So, people who stop outside the mentioned region, are not considered interested in interacting.

The system was developed in C++ and runs in a PC with MS Windows installed, a Core 2 Duo processor 2.1GHz and 4GB RAM. This PC is capable to execute around 25 loops per second, however we fixed the execution time in 10 loops per second. The robot used to perform experiments is shown in Figure 7.

Figure 7: Robot used to perform the experiments.

In order to show the reliability of the proposed method, some experiments were performed. The robot was positioned in a free area and, while the leg detection algorithm was running, some people were asked to stop in front of the robot in the same manner they would stop when they want to talk to another person and, sometimes, the experiment was performed with more than one person. it were performed 152 detections and the algorithm was able to classify correctly 88.16% of the cases. Figure 8 shows a person stopped in front of the robot and Figure 9 brings the detection rates obtained during this experiment.

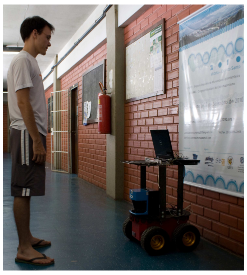


Figure 8: A person stopped in front of the robot.

Before showing the experimental results, it is important to mention that during the tests nobody stopped in front of the robot with the legs together. Due to this, in order to perform a complete set of experimental results, some people were asked to stand in front of the robot with their legs together. Following subsections illustrate some experimental results of the proposed leg detection algorithm.

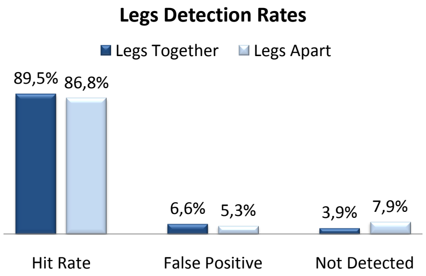


Figure 9: Detection Rates.

3.1 Experiment 01

In this experiment two people were standing still in front of the robot with the legs apart. Figure 10 shows the distances measures captured by the laser scanner and the obtained result for the leg detection algorithm. The circles represent the detected pair of legs by the FSMs, and the numbers represent the probability of being legs of each detected pair of legs.

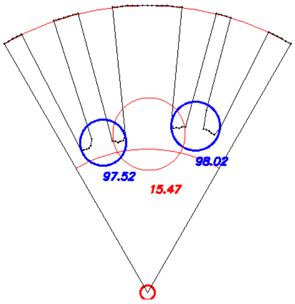


Figure 10: Experiment 01 - obtained results.

Notice that in Figure 10 there are three circles, i. e., the FSMs detected three possible pair of legs apart. However, the second pair of legs detected by the FSMs has a low probability (15.47%) and it is not considered a pair of legs. Moreover the other two pairs have probabilities greater than 97% (see Figure 10) and are classified as a pair of legs. However, if the people were closer one from another, the algorithm would detect three pair of legs instead of two. It can be solved with a face detection algorithm.

3.2 Experiment 02

In this experiment three people stood in front of the robot and one set of measures was captured by the laser. Two of them have the legs apart and the third one has the legs together. The person who is in the middle is the one with the legs together. Figure 11 shows the distance measures captured by the laser scanner and the pattern identification done by the FSMs represented by the circles.

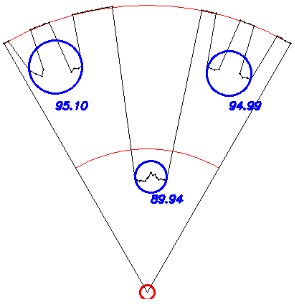


Figure 11: Experiment 02 - obtained results.

The probabilities of being a pair of legs, calculated after the legs pattern identification, are also shown in Figure 11. Notice that all the patterns have a high probability of being a pair of legs, which means that the leg detection algorithm classified correctly the people's legs.

3.3 Experiment 03

As the Experiment 1, this experiment shows the legs detection result for two people stopped in front of the robot. However, here one person has the legs apart and the other one has the legs together. This experiment has the particularity that these two people did not stop in front of the robot with the legs parallel. In this case, as can be seen in Figure 12, one of the legs is closer to the robot.

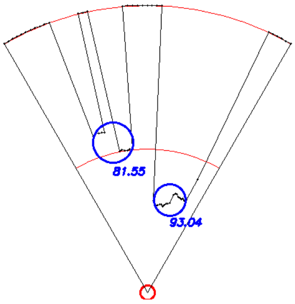


Figure 12: Experiment 03 - obtained results.

For this experiment, the pattern detection by the FSMs and the probabilities of being a pair of legs are shown in Figure 12. Although the legs are not parallel, the algorithm was able do classify correctly the patterns detected by the FSMs.

4 CONCLUSIONS AND FUTURE WORK

We proposed in this paper a method to find human legs using a LRF. The method utilizes the distance measures provided by the laser scanner and look for some legs patterns using two FSMs and, after that, calculates the probability of each detected pattern being a pair of legs.

Some experiments were presented to show the performance of the proposed method. It was demonstrated that the method can detect human legs with accuracy, but since we used only laser sensor information, some false positives can be detected. In order to reduce this false positives and solve occlusion cases, our future work is concerned in introducing a face detection and, thus, performing human-robot interaction.

REFERENCES

Arras, K., Mozos, O., and Burgard, W. (2007). Using boosted features for the detection of people in 2d range data. In *2007 IEEE International Conference on Robotics and Automation (ICRA 2007)*, pages 3402–3407.

Bekey, G. A. (2005). *Autonomous Robots : From Biological Inspiration to Implementation and Control (Intelligent Robotics and Autonomous Agents)*. The MIT Press.

Bellotto, N. and Hu, H. (2009). Multisensor-based human detection and tracking for mobile service robots. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 39(1):167–181.

Carballo, A., Ohya, A., and Yuta, S. (2009). Multiple people detection from a mobile robot using a double layered laser range finder. In *IEEE ICRA 2009 Workshop on People Detection and Tracking*, Kobe, Japan.

Cui, J., Zha, H., Zhao, H., and Shibasaki, R. (2005). Tracking multiple people using laser and vision. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2005. (IROS 2005).*, pages 2116–2121.

Fod, A., Howard, A., and Mataric, M. (2002). A laser-based people tracker. In *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, volume 3, pages 3024–3029.

Kleinehagenbrock, M., Lang, S., Fritsch, J., Lomker, F., Fink, G., and Sagerer, G. (2002). Person tracking with a mobile robot based on multi-modal anchoring. In *Robot and Human Interactive Communication, 2002. Proceedings. 11th IEEE International Workshop on*, pages 423–429.

Luo, R., Chen, Y., Liao, C., and Tsai, A. (2007). Mobile robot based human detection and tracking using range and intensity data fusion. In *Advanced Robotics and Its Social Impacts, 2007. ARSO 2007. IEEE Workshop on*, pages 1–6.

Müller, S., Erik, S., Scheidig, A., Böhme, H.-J., and Gross, H.-M. (2007). Are you still following me? In *Proceedings of the 3rd European Conference on Mobile*, pages 211–216, Freiburg.

Scheutz, M., McRaven, J., and Cserey, G. (2004). Fast, reliable, adaptive, bimodal people tracking for indoor environments. In *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 2, pages 1347–1352 vol.2.

Schulz, D., Burgard, W., Fox, D., and Cremers, A. (2001). Tracking multiple moving targets with a mobile robot using particle filters and statistical data association. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 2, pages 1665–1670 vol.2.

Topp, E. and Christensen, H. (2005). Tracking for following and passing persons. In *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 2321–2327.

Xavier, J., Pacheco, M., Castro, D., Ruano, A., and Nunes, U. (2005). Fast line, arc/circle and leg detection from laser scan data in a player driver. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 3930–3935.

FORMATION CONTROL BETWEEN A HUMAN AND A MOBILE ROBOT BASED ON STEREO VISION

Flávio Garcia Pereira, Marino Frank Cypriano and Raquel Frizera Vassallo
Department of Electrical Engineering, Federal University of Espírito Santo
Av. Fernando Ferrari, 514, Goiabeiras, Vitória-ES, CEP 29075-910, Brazil
{flaviog, marino, raquel}@ele.ufes.br

Keywords: Formation control, Stereo vision.

Abstract: This paper presents a reactive nonlinear formation control between a human and a mobile robot based on stereo vision. The human, who is considered the leader of the formation, is positioned behind the robot and moves around the environment while the robot executes movements according to human movements. The robot has a stereo vision system mounted on it and the images acquired by this system are used to detect the human face and estimate its 3D coordinates. These coordinates are used to determine the relative pose between the human and robot, which is sent to the controller in order to allow the robot to achieve and keep a desired formation with the human. Experimental results were performed with a human and a mobile robot, the leader and follower respectively, to show the performance of the control system. In this paper a load was put on a mobile robot in which the nonlinear controller was applied, allowing the transportation of this load from a initial point to a desired position. In order to verify the proposed method some experiments were performed and shown in this paper.

1 INTRODUCTION

Navigating and following are the two basic functionalities of mobile robots. Both of them are widely studied topics. Many strategies have been developed for several kinds of environments either with a single agent systems or multi-agent systems (Bicho and Monteiro, 2003; Das et al., 2002; Egerstedt and Hu, 2001; Egerstedt et al., 2001; Jadbabaie et al., 2002; Vidal et al., 2003).

In most of these studies the robots are used to navigate in order to reach a specific goal point in a given environment while avoiding obstacles (Althaus et al., 2004). However, due to limitations in sensing, most times, robots are not able to operate in complex environments, even if they are provided with accurate sensors like a 3D range finder. Therefore, we believe that a human-robot navigation system is a good solution for this kind of problem, since human can lead the robot in a cooperative task.

The motivation for the work presented in this paper comes from situations where robots are used to help people in an object transportation task where the person does not load the object, only guides the robot through the environment.

A new issue for navigation emerges from interaction tasks, which concerns the control of the platform (Althaus et al., 2004). The main idea is that the robot's behavior must be smooth and similar to the human's movements.

In human-robot interaction the actions of the robot depend on the presence of humans and their wish to interact.

In (Bowling and Olson, 2009) a human-robot team is composed by two omnidirectional robots and one human. The authors consider the case in wich a human turns abruptly a corner and the robots may keep the human in their line of sight. The strategy is accomplished by taking the center of mass of the formation and analyzing its rotation.

The work proposed in this paper consists in coordinating the movements of a mobile robot by a human in order to transport a load from a desired point to another one in a safe way. In this system, the human, which is considered the formation leader, is responsible for guiding the robot through the environment by making the load transportation from an initial point to a goal position and orientation. The robot (follower) must move according to the human maitaining the formation. This type of cooperative work needs the robot to acquire enough information about the human position to achieve the goal. This is reached through a vision-based nonlinear controller which estimates the linear and angular velocities that the robot must execute to follow the human. This application is useful because as the load is on the robot, the human does not need to carry the object with it. The human function is to guide the robot through the environment.

Unlike the conventional leader-follower cooperation, where commonly the formation leader is positioned in front of the follower(s), the formation control proposed in this work takes into consideration the fact that the robot navigates in front of the human, which is considered the leader of the formation. As the formation control is based on vision, a stereo vision system is mounted on top of the robot pointing backwards, which allows the robot to navigate "looking" at the human face, instead of "looking" at the back of the human. Another interesting point of this formation is the fact that, since the robot goes in front of the human (leader), the personn can observe the robot behavior during the cooperation task and, in addition, can notice the presence of obstacles and guide the robot in order to avoid them. This is something valuable because we focused on the cooperation problem and still did not include an obstacle avoidance algorithm to the robot behavior. That is one of our issues for future work.

This paper is organized as follows. The technique for estimating the face pose is briefly discussed in Section 2 and the formation controller as its stability analysis are shown in Section 3. Some experimental results are given in Section 4 and, finally, the conclusions and future work are commented in Section 5.

2 HUMAN DETECTION

The nonlinear controller that will be presented in Section 3 needs the distance from the human to the robot (d) and the angle between the vector which links the human position with the center of the robot (θ). In order to estimate these two variables, it is used a calibrated stereo vision system attached to the robot. The human pose is estimated by detecting the face in both images captured by the vision system and, afterwards, recovering its 3D coordinates.

The face detection is performed with the algorithm proposed in (Viola and Jones, 2001), which utilizes several images to train a detection algorithm based on Adaboost. After detecting the face in both images, an algorithm to detect some facial characteristics is applied and the coordinates of these points in the image plane are determined. Knowing the positions of these points in both images, their 3D coordinates can be recovered. These facial features are, for instance, the eyes, nose and mouth. Figures 1 (a) and 1 (b) show the images captured by the stereo vision system and the detected facial features.

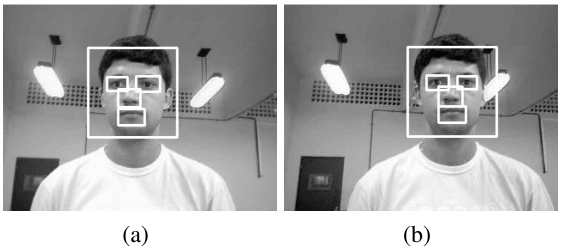


Figure 1: Images captured by the stereo vision system.

The face coordinates is considered as being the average of the facial features coordinates, i. e.,

$$[X,Y,Z] = \frac{1}{N}\sum_{i=1}^{N}[x_i,y_i,z_i], \tag{1}$$

where N is the number of detected features and $[x_i,y_i,z_i]$, are the 3D coordinates of each detected facial feature in the vision system reference frame attached to the robot. Figure 2 illustrates the robot and the stereo vision system reference frames.

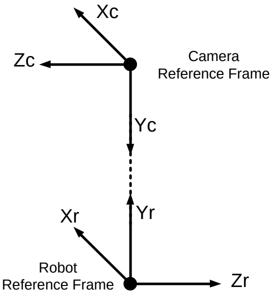


Figure 2: Robot and vision system reference frames.

The relation between these coordinate frames is given by a rotation of 180° around the x-axis and a translation of $1.2m$ in the y-axis. Thus,

$$R_f = T_y(1.2)R_x(180)C_f \tag{2}$$

where R_f and C_f are respectively the coordinates in the robot reference frame and in the camera reference frame.

After estimating the 3D coordinates of the face, the variables d and θ are calculated on the vision system's XZ-plane (ground plane). Figure 3 shows the human and the vision system positions in the XZ-plane.

The values of d and θ, that define the distance and the angle between the human and the robot, are deter-

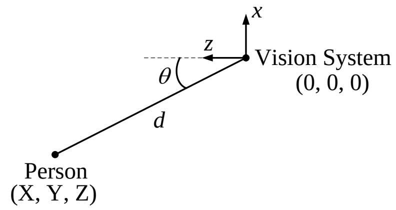


Figure 3: Human and vision system position in the XZ-plane.

mined as

$$\begin{cases} d = \sqrt{X^2+Z^2} \\ \theta = \operatorname{atan}\left(\dfrac{X}{Z}\right), \end{cases} \tag{3}$$

where X and Z are, respectively the x and z coordinates of the face. Notice that, however d and θ where calculated in the camera frame, they are valid in the robot frame, due to the aligment between the two frames.

3 CONTROLLER

The aim of the proposed controller is to keep the robot with a desired distance and orientation to the human partner in order to accomplish a specific task. That is done by using the information obtained from the pair of images provided by the stereo vision system assembled on the robot.

Figure 4 illustrates a mobile robot with a stereo vision system attached to it and one person behind it. Considering this case, the kinematic equations that describe the robot movement with respect to the person are given by

$$\begin{cases} \dot{d} = v_h - v_r\cos(\theta) \\ \dot{\theta} = \omega + \dfrac{v_r\sin(\theta)}{d} + \dfrac{v_h\sin(\tilde{\theta})}{d}, \end{cases} \tag{4}$$

where v_r and ω are, respectively, the robot linear and angular velocities, v_h is the human velocity, d is the distance between the person and the vision system while θ is the angle of the human face according to the stereo vision system frame and $\tilde{\theta}$ is the angular error.

The values of d and θ are obtained from the 3D coordinates of the middle of the face after 3D reconstruction. The way to obtain these values was explained in Section 2.

After determining d and θ, the distance and orientation errors ($\tilde{d}$, $\tilde{\theta}$) are calculated as

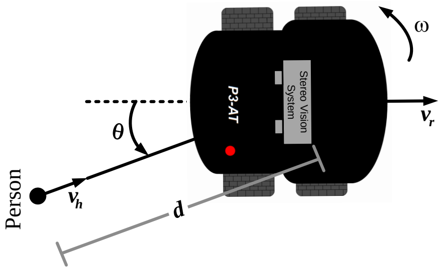


Figure 4: Relative position between the robot and the person.

$$\begin{cases} \tilde{d} = d_d - d \\ \tilde{\theta} = \theta_d - \theta, \end{cases} \tag{5}$$

where d_d and θ_d are respectively the desired distance and orientation that the robot has to keep from the human.

For the presented system, it is proposed a nonlinear controller in order to accomplish the objective of control, i. e, to keep the robot following the human with specified distance and orientation. For this controller, the linear and angular velocities are

$$\begin{cases} v_r = \dfrac{1}{\cos(\theta)}\left(v_h - k_d\tanh(\tilde{d})\right) \\ \omega = k_\theta\tanh(\tilde{\theta}) - v_r\dfrac{\sin(\theta)}{d} - v_h\dfrac{\sin(\tilde{\theta})}{d}, \end{cases} \tag{6}$$

with $k_d,\ k_\theta > 0$.

The control structure used in this work appears in Figure 5. The operations of detecting faces and facial features, recovering the 3D coordinates and determining the values of d and θ, are executed by the "Image Processing" block.

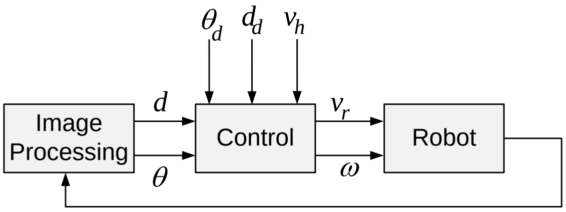


Figure 5: Control structure.

3.1 Stability Analysis

For the nonlinear controller proposed in (6) the following candidate Lyapunov function is taken to check the system stability

$$V(\tilde{d},\tilde{\theta}) = \frac{1}{2}\tilde{d}^2 + \frac{1}{2}\tilde{\theta}^2, \tag{7}$$

which is positive semi-defined and has the following time-derivative function

$$\dot{V}(\tilde{d},\tilde{\theta}) = \tilde{d}\dot{\tilde{d}} + \tilde{\theta}\dot{\tilde{\theta}}. \tag{8}$$

Thus, differentiating (5) with respect to time and replacing the result in (8) we will have

$$\dot{V}(\tilde{d},\tilde{\theta}) = -\tilde{d}\dot{d} - \tilde{\theta}\dot{\theta}. \tag{9}$$

Now, the values of $\dot{d}$ and $\dot{\theta}$ in (9) are substituted by the respective values from (4). Then we have

$$\dot{V}(\tilde{d},\tilde{\theta}) = -\tilde{d}[v_h - v_r\cos(\theta)] - \tilde{\theta}\left[\omega + \frac{v_r\sin(\theta)}{d} + \frac{v_h\sin(\tilde{\theta})}{d}\right]. \tag{10}$$

Replacing the control actions proposed (6) in (10), the time-derivative of the Lyapunov function is rewritten as

$$\dot{V}(\tilde{d},\tilde{\theta}) = -k_d\tanh(\tilde{d})\tilde{d} - k_\theta\tanh(\tilde{\theta})\tilde{\theta}. \tag{11}$$

As mentioned k_d and k_θ are positive, and functions like $x.\tanh(x) \geq 0 \,\forall\, x$. So, (8) is negative semi-defined, which means the asymptotic stability of the proposed nonlinear controller, i. e., $\tilde{d},\tilde{\theta} \rightarrow 0$ when $t \rightarrow \infty$.

Notice that to estimate the angular velocity it is necessary to know the human velocity, v_h. This velocity can be estimated as

$$v_h \approx \frac{d(t) - d(t-1)}{\Delta t} - v_r\cos(\tilde{\theta}). \tag{12}$$

4 EXPERIMENTAL RESULTS

In order to prove the reliability of the proposed method, some experiments were performed and two of them will be shown in this section. These experiments were performed with a mobile robot Pioneer 3AT from ActivMedia Robotics, a stereo vision system built with two USB web cameras, which provide images with 320x240 pixels. The control system is written in C++ and each control loop is executed in 250ms. The robot with the stereo vision system used to perform the experiments can be seen in Figure 6.

4.1 Experiment 1

In this experiment, the human leads the robot from the initial position, which has coordinates $(X,\, Z) = (0m,\, 0m)$ to the coordinates $(X,\, Z) = (3m,\, -5m)$ through a “S-like” path. The trajectory described by the robot can be seen in Figure 7.

The linear and angular velocities performed by the robot are shown, respectively, in Figures 8 (a) and (b).

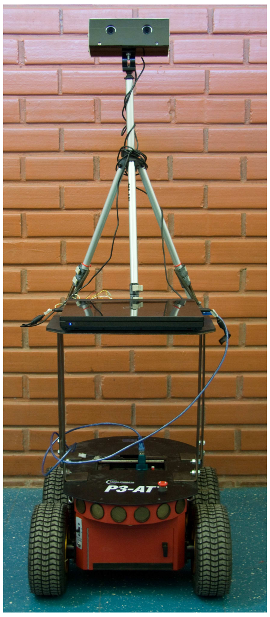


Figure 6: Robot used to perform the experiments.

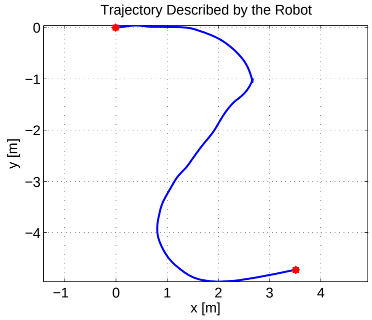
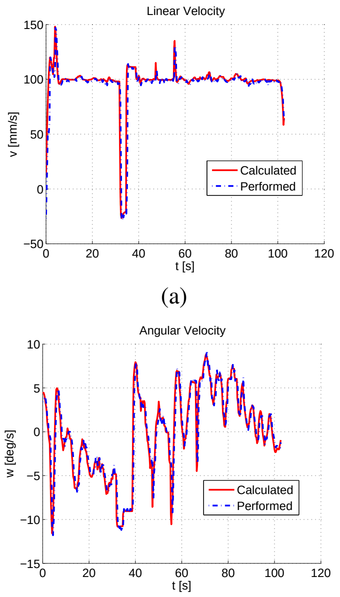


Figure 7: Experiment 01: Trajectory described by the robot.

The continuous line represents the velocities calculated by the controller and the velocities performed by the robot are plotted as dashed line.

In Section 3.1 the control system stability was demonstrated, i. e., it was proved that the system errors tend to zero when the time grows and if the human stops at some place. In order to illustrate this, the graphics shown in Figure 9 present the errors during the experiment. Figure 9 (a) shows the distance error during the experiment while the angular error is presented in Figure 9 (b).

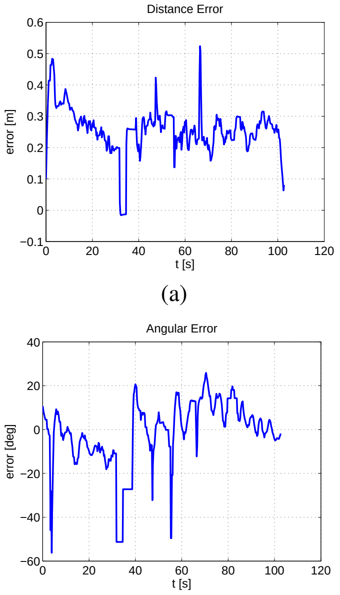
By observing Figure 7 it is possible to see that the robot reached the goal describing desired trajectory. Moreover, as can be seen in Figure 9, both the distance error and the orientation error tend to zero, which means the robot accomplished the task. For the next experiments, the errors will not be shown.



(a)

(b)

Figure 8: Experiment 01: Velocities performed by the robot. Linear (a) and Angular (b).



(a)

(b)

Figure 9: Experiment 01: Distance error (a) and Angular error (b).

4.2 Experiment 2

In this experiment the human should guide the robot from the initial position $(X, Y) = (0m, 0m)$ to the final coordinates $(X, Y) = (5m, 0m)$. However, there was a circular obstacle with diameter equal to $0.4m$. The trajectory described by the robot during the execution of this task is shown in Figure 10.

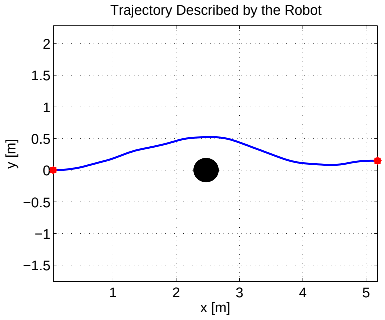
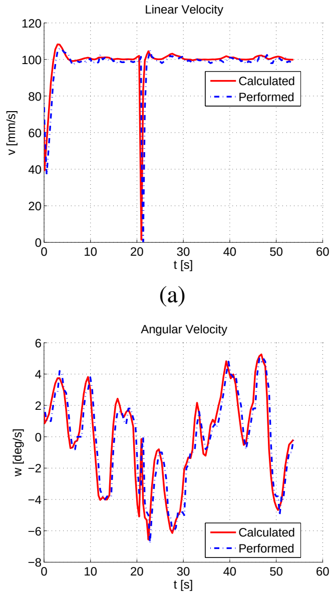


Figure 10: Experiment 02: Trajectory described by the robot.

It is important to mention that the robot does not have an obstacle avoidance algorithm. So, the deviation that can be seen in Figure 10 occurred due to the human action, i. e., the person who leads the robot noticed an object in the middle of the desired trajectory and guided the robot in order to deviate from it. Even though it seems that the person has to walk in an unnatural way in order to prevent the robot from crashing into obstacles or corner in a hallway, this problem will be solved by adding an obstacle avoiding strategy such as in (Chuy et al., 2007; Pereira, 2006).

The linear velocity is shown in Figure 11 (a) while the angular velocity appears in Figure 11 (b).



(a)

(b)

Figure 11: Experiment 02: Velocities performed by the robot. Linear (a) and Angular (b).

Notice that around the 20th second of the experiment, both linear and angular velocities calculated by the controller are equal to zero. This happened because in this instant, the robot lost the information about the person for, at least, one of the cameras of the stereo vision system. This behavior was adopted in order to avoid having the robot be out of control every time the vision system loses the information about the human. Thus, whenever the robot loses the information about its leader, it stops and waits for a new face detection.

Although an obstacle in the middle of the desired trajectory, the human managed to guide the robot to avoid this object and reach the desired point.

5 CONCLUSIONS AND FUTURE WORK

This paper presented an approach to a formation control between a human and a mobile robot using stereo vision. The strategy uses the recognition method presented by (Viola and Jones, 2001) in order to find the facial features needed to estimate the human position. A stable nonlinear controller is proposed to allow the robot to perform the task in cooperation with a human. The effectiveness of the proposed method is verified through experiments where a human guides the robot from an initial position to a desired point, when the human moves either forward or backwards.

Our future work is concerned in improving the features detection in order to better estimate the human position and orientation. Besides, we also intend to introduce a second robot to the formation. Thus, the robots will be able to carry a bigger and heavier load.

REFERENCES

Althaus, P., Ishiguro, H., Kanda, T., Miyashita, T., and Christensen, H. (2004). Navigation for human-robot interaction tasks. In *ICRA '04. 2004 IEEE International Conference on Robotics and Automation, 2004. Proceedings.*, volume 2, pages 1894–1900 Vol.2.

Bicho, E. and Monteiro, S. (2003). Formation control for multiple mobile robots: a non-linear attractor dynamics approach. In *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 2, pages 2016–2022 vol.2.

Bowling, A. and Olson, E. (2009). Human-robot team dynamic performance in assisted living environments. In *PETRA '09: Proceedings of the 2nd International Conference on PErvsive Technologies Related to Assistive Environments*, pages 1–6, New York, NY, USA. ACM.

Chuy, O., Hirata, Y., and Kosuge, K. (2007). Environment feedback for robotic walking support system control. In *ICRA*, pages 3633–3638.

Das, A., Fierro, R., Kumar, V., Ostrowski, J., Spletzer, J., and Taylor, C. (2002). A vision-based formation control framework. *Robotics and Automation, IEEE Transactions on*, 18(5):813–825.

Egerstedt, M. and Hu, X. (2001). Formation constrained multi-agent control. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 4, pages 3961–3966 vol.4.

Egerstedt, M., Hu, X., and Stotsky, A. (2001). Control of mobile platforms using a virtual vehicle approach. *Automatic Control, IEEE Transactions on*, 46(11):1777–1782.

Jadbabaie, A., Lin, J., and Morse, A. (2002). Coordination of groups of mobile autonomous agents using nearest neighbor rules. In *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, volume 3, pages 2953–2958 vol.3.

Pereira, F. G. (2006). Navegação e desvio de obstáculos usando um robô móvel dotado de sensor de varredura laser. Master's thesis, Universidade Federal do Espírito Santo - UFES.

Vidal, R., Shakernia, O., and Sastry, S. (2003). Formation control of nonholonomic mobile robots with omnidirectional visual servoing and motion segmentation. In *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, volume 1, pages 584–589 vol.1.

Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001. CVPR 2001.*, 1:I–511–I–518 vol.1.

NAVIGATION AND FORMATION CONTROL EMPLOYING COMPLEMENTARY VIRTUAL LEADERS FOR COMPLEX MANEUVERS

Martin Saska, Vojtěch Vonásek and Libor Přeučil
Department of Cybernetics, Czech Technical University in Prague, Technická 2, Prague, Czech Republic
{*saska, vonasek, preucil*}*@labe.felk.cvut.cz*

Keywords: Autonomous mobile robots, Compact formations, Virtual leaders, Trajectory planning.

Abstract: Complex maneuvers of formations of car-like autonomous vehicles are investigated in this paper. The proposed algorithm provides a complete plan for the formation to solve desired tasks and actual control inputs for each robot to ensure collision-free trajectories with respect to neighboring robots as well as dynamic obstacles. The method is based on utilization of complementary virtual leaders whose control inputs are obtained in one merged optimization process using receding horizon control methodologies. The functionality of the system, which enables reverse driving and arbitrary rotations of formations of nonholonomic robots, is verified by simulations of multi-robot tasks and by hardware experiments.

1 INTRODUCTION

In this paper, we present a novel approach for a nonholonomic robots formation driving and a trajectory planning to reach a target region. The method allows arbitrary maneuvers, including reverse driving and turning of the formation in limited environment with static as well as dynamic obstacles. Such skills are crucial for the control of autonomous ploughs during cooperative airport snow shoveling (Saska et al., 2008; Hess et al., 2009) being our target applications.

Tasks for which autonomous vehicles in formation are used usually involve physical constraints imposed by the vehicles (mobility constraints) or physical obstacles (environment constraints) and constraints enforced by inter vehicle relations (shape of the formation). Thus, it is important for control methodologies to incorporate system's constraints into the controllers design while preserving overall system stability which makes the Receding Horizon Control (RHC) especially appealing. Receding horizon control (also known as model predictive control) is an optimization based control approach often used for stabilizing linear and nonlinear dynamic systems (*e.g.*, see (Alamir, 2006) and references reported therein). For a detailed survey of RHC methods we refer to (Mayne et al., 2000) and references reported therein. The works applying RHC for formation driving control are presented in (Dunbar and Murray, 2006) (Franco et al., 2008). These papers have utilized RHC for the formation forming and/or following predefined trajectory in a workspace without obstacles. We will apply the Receding Horizon Control for the virtual leaders trajectory planning to desired goal area and for the followers stabilization in the formation.

Our contribution is a general approach that accounts for a nonholonomic robots' formation stabilization and a trajectory planning (enforced by a stability constraint) to reach a target region, also considering the response to dynamic changes in the environment. We developed a new concept of RHC by combining both, the trajectory planning to the desired goal region and the immediate control of the formation, into one optimization process. Our method can continuously respond to changes in environment of the robots while the cohesion of the immediate control inputs with direction of the formation movement in future is kept. Furthermore, we extended the common leader-follower concept with the idea of two virtual leaders, one for the forward and one for the backward movement. Such an approach is necessary for the complicated maneuvers of the formation of car-like robots. The proposed method provides a complete plan together with decisions when and how often to switch between the guidance of the virtual leaders.

2 PRELIMINARIES

Let $\psi_L(t) = \{x_L(t), y_L(t), \theta_L(t)\}$ denote the configuration of a virtual leader R_L at time t,[1] and $\psi_i(t) = \{x_i(t), y_i(t), \theta_i(t)\}$, with $i \in \{1,\dots,n_r\}$, denote the configuration for each of the n_r followers R_i at time t. The Cartesian coordinates $(x_j(t), y_j(t))$, $j \in \{1,\dots,n_r,L\}$ for an arbitrary configuration $\psi_j(t)$ define the position $\bar{p}_j(t)$ of a robot R_j and $\theta_j(t)$ denotes its heading. Let us assume that the environment of the robots contains a finite number n_0 of compact obstacles collected in a set of regions $\mathcal{O}_{obs}$. Let us also define a target region S_F as a circle with radius r_{S_F} and center C_{S_F} such that $\mathcal{O}_{obs} \cap S_F = \emptyset$. Finally, we need to define a circular detection boundary with radius r_s and a circular avoidance boundary with radius r_a, where $r_s > r_a$. Single robots should not respond to obstacles detected outside the region with radius r_s. On the contrary, distance between the robots and obstacles less than r_a is considered as inadmissable. These concepts are adopted from the concept of avoidance control (Leitmann, 1980) to be used in this paper for the collision avoidance guaranties. Moreover, we need to extend these zones for the virtual leaders depending on size of the formation to ensure that the result of the leader trajectory planning is feasible also for the followers. We will denote the extended radiuses as $r_{s,L}$ and $r_{a,L}$.

2.1 Kinematic Model and Constraints

The kinematics for any robots R_j, $j \in \{1,\dots,n_r,L\}$, in the formation is described by the simple nonholonomic kinematic model: $\dot{x}_j(t) = v_j(t)\cos\theta_j(t)$, $\dot{y}_j(t) = v_j(t)\sin\theta_j(t)$, $\dot{\theta}_j(t) = K_j(t)v_j(t)$, where velocity $v_j(t)$ and curvature $K_j(t)$ represent control inputs $\bar{u}_j(t) = (v_j(t), K_j(t)) \in \mathbb{R}^2$.

Let us define a time interval $[t_0, t_{N+M}]$ containing a finite sequence with $N+M+1$ elements of nondecreasing times $\mathcal{T}(t_0, t_{N+M}) = \{t_0, t_1, \dots, t_{N-1}, t_N, \dots, t_{N+M-1}, t_{N+M}\}$, such that $t_0 < t_1 < \dots < t_{N-1} < t_N < \dots < t_{N+M-1} < t_{N+M}$.[2] Also, let us define a controller for a robot R_j starting from a configuration $\psi_j(t_0)$ by $\mathcal{U}_j(t_0, t_{N+M}, \mathcal{T}(\cdot)) := \{\bar{u}_j(t_0; t_1 - t_0), \bar{u}_j(t_1; t_2 - t_1), \dots, \bar{u}_j(t_{N+M-1}; t_{N+M} - t_{N+M-1})\}$ which is characterized as a sequence of constant control actions. Each element $\bar{u}_j(t_k; t_{k+1} - t_k)$, $k \in \{0,\dots,N+M-1\}$, of the finite sequence $\mathcal{U}_j(t_0, t_{N+M}, \mathcal{T}(\cdot))$ will be held constant during the time interval $[t_k, t_{k+1})$ with length $t_{k+1} - t_k$ (not necessarily uniform).

By integrating the kinematic model in over a given interval $[t_0, t_{N+M}]$, and holding constant control inputs over each time interval $[t_k, t_{k+1})$ (from this point we may refer to t_k using its index k), we can derive the following model for the *transition points* at which control inputs change:

$$
\begin{aligned}
x_j(k+1) &= \begin{cases} x_j(k) + \frac{1}{K_j(k+1)}\left[\sin\left(\theta_j(k) + \right.\right. \\ \left.\left. K_j(k+1)v_j(k+1)\Delta t(k+1)\right) - \right. \\ \left. \sin\left(\theta_j(k)\right)\right], \text{if} K_j(k+1) \neq 0; \\ x_j(k) + v_j(k+1)\cos\left(\theta_j(k)\right)\Delta t(k+1), \\ \text{if } K_j(k+1) = 0 \end{cases} \\
y_j(k+1) &= \begin{cases} y_j(k) - \frac{1}{K_j(k+1)}\left[\cos\left(\theta_j(k) + \right.\right. \\ \left.\left. K_j(k+1)v_j(k+1)\Delta t(k+1)\right) - \right. \\ \left. \cos\left(\theta_j(k)\right)\right], \text{if} K_j(k+1) \neq 0; \\ y_j(k) + v_j(k+1)\sin\left(\theta_j(k)\right)\Delta t(k+1), \\ \text{if } K_j(k+1) = 0 \end{cases} \\
\theta_j(k+1) &= \theta_j(k) + K_j(k+1)v_j(k+1)\Delta t(k+1),
\end{aligned}
\tag{1}
$$

where $x_j(k)$ and $y_j(k)$ are the rectangular coordinates and $\theta_j(k)$ the heading angle for the configuration $\psi_j(k)$ at the transition point with index k, $v_j(k+1)$ and $K_j(k+1)$ are control inputs at time index $k+1$, and $\Delta t(k+1)$ is the sampling time. This model and notation allows us to describe long trajectories exactly using a minimal amount of information such as: i) the initial configuration $\psi_j(t_0)$, ii) the sequence of switching times $\mathcal{T}(\cdot)$, and iii) the sequence of controls actions $\mathcal{U}_j(\cdot)$.

In real applications, the control inputs are limited by vehicle mechanical capabilities (i.e., chassis and engine). This constraints can be taken into account for each robot R_j, $j \in \{1,\dots,n_r,L\}$, limiting their control inputs by the following inequalities: $v_{min,j} \leq v_j(k) \leq v_{max,j}$ and $|K_j(k)| \leq K_{max,j}$, where $v_{max,j}$ is the maximal forward velocity of the j-th vehicle, $v_{min,j}$ is the limit on the backward velocity and $K_{max,j}$ is the maximal control curvature. These values can be different for each robot R_j.

2.2 Formation Driving using RHC

The presented approach relies on the well known leader-follower method frequently used in applications of car like robots (Barfoot and Clark, 2004). In the method, the followers R_i, $i \in \{1,\dots,n_r\}$ track the leader's trajectory which is distributed within the group. The followers are maintained in relative distance to the leader in curvilinear coordinates with two axes p and q, where p traces the leader's trajectory and q is perpendicular to p as is demonstrated in Figure 1. The positive direction of p is defined from R_L back to the origin of the movement and the positive

[1] Index L denotes the virtual leader which has assigned the leadership at the moment while indexes L_1 and L_2 distinguish between particular virtual leaders only if it is necessary.

[2] The meaning of constants N and M will be explained in Section 3.

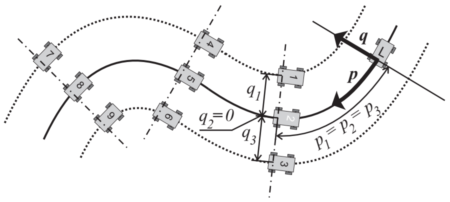


Figure 1: Formation described by curvilinear coordinates p and q.

direction of q is defined in the left half plane in direction of the forward movement.

The shape of the formation is then uniquely determined by the parameters $p_i(t)$ and $q_i(t)$, defined for each follower i, which can vary during the mission. To convert the state of the followers in curvilinear coordinates to the state in rectangular coordinates, the simple equations in (Barfoot and Clark, 2004) can be applied.

The main idea of the receding horizon control is to solve a moving finite horizon optimal control problem for a system starting from current states or configuration $\psi(t_0)$ over the time interval $[t_0, t_f]$ under a set of constraints on the system states and control inputs. In this framework, the length $t_f - t_0$ of the time interval $[t_0, t_f]$ is known as the control horizon. After a solution from the optimization problem is obtained on a control horizon, a portion of the computed control actions is applied on the interval $[t_0, \delta t_n + t_0]$, known as the receding step, where $\delta t_n := \Delta t n$.[3] This process is then repeated on the interval $[t_0 + \delta t_n, t_f + \delta t_n]$ as the finite horizon moves by *time steps* defined by the sampling time δt_n, yielding a state feedback control scheme strategy. Advantages of the receding time horizon control scheme become evident in terms of adaptation to unknown events and change of strategy depending on new goals or new events such as appearing obstacles in the environment.

3 METHOD DESCRIPTION

In this section, a concept of complementary virtual leaders, which enables backward driving of formations, will be proposed. The basic idea of the classical leader-follower concept (see Figure 1) is based on the fact that the followers continue with forward movement until the place where the leader changed the polarity of its velocity. Such a behavior could cause collisions or unacceptable disordered motion leading to a breakage of the shape of formation. Natural conception of the formation movement supposes that the entire group keeps compact shape and so all members should change the polarity of their velocity in the same moment.

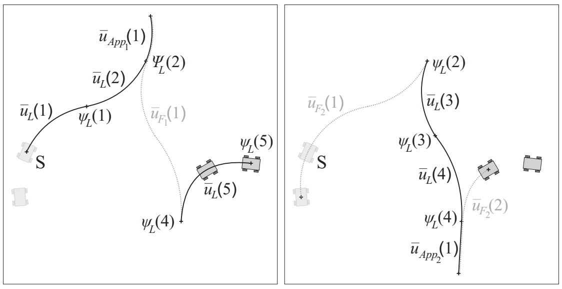
Such an approach requires an extension of the standard method employing one leader to an approach with two virtual leaders, one for the forward movement and one for the backward movement. Their leading role is switched always when the sign of the leader's velocity is changed. The suspended virtual leader becomes temporarily a virtual follower. The virtual follower traces the virtual leader similarly as the other followers to be able to undertake its leading duties at time of the next switching. Therefore, all robots in our system will be considered as followers and there is no physical leader. The virtual leaders will be positioned at the axis of the formation, one in front of the formation and one behind the formation.

In the presented approach, we propose to solve collision free trajectory planning and optimal control together for both virtual leaders in one optimization step. This ensures integrity of the solution for the separated plants. Beyond this, we extend the standard RHC method with one control horizon into an approach utilizing two finite time intervals T_N and T_M. The first time interval T_N should provide immediate control inputs for the formation regarding the local environment. By applying this portion of the control sequence, the group is be able to respond to changes in workspace that can be dynamic or newly detected static obstacles. The difference $\Delta t(k+1) = t_{k+1} - t_k$ is kept constant (later denoted only Δt) in this time interval and should satisfy the requirements of the classical receding horizon control scheme.

The second interval T_M takes into account information about the global characteristics of the environment to navigate the formation to the goal and to automatically compose the entire maneuver containing usually multiple switching between the virtual leaders. Here, we should highlight that also the number of switchings between the virtual leaders is designed automatically via the optimization process. The transition points in this part can be distributed irregularly to effectively cover the environment. During the optimization process, more points should be automatically allocated in the regions where a complicated maneuver of the formation is needed. This is enabled due to the varying values of $\Delta t(k+1) = t_{k+1} - t_k$ that will be for the compact description collected into the vector $\mathcal{T}^{\Delta}_{L,M} = \{\Delta t(N+1), \ldots, \Delta t(N+M)\}$.

To define the trajectory planning problem with two time intervals in a compact form we need to gather states $\psi_L(k), k \in \{1, \ldots, N\}$ and $\psi_L(k), k \in$

[3] Number of applied constant control inputs n is chosen according to computational demands as was explained in (Saska et al., 2009).



(a) Trajectory of the first virtual leader. (b) Trajectory of the second virtual leader.

Figure 2: Trajectories with denoted control inputs and states of the two virtual leaders. The solid black curves denote trajectories where the virtual leader is leading the formation while the gray curves denote trajectories where the virtual leader is the virtual follower. The initial positions of the leaders are depicted by gray shadows.

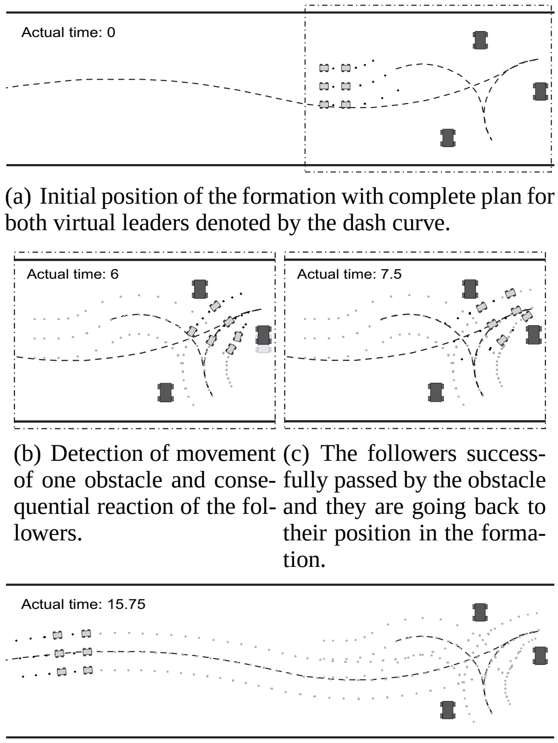
$\{N+1,\dots,N+M\}$ into vectors $\Psi_{L,N} \in \mathbb{R}^{3N}$ and $\Psi_{L,M} \in \mathbb{R}^{3M}$. Similarly the control inputs $\bar{u}_L(k), k \in \{1,\dots,N\}$ and $\bar{u}_L(k), k \in \{N+1,\dots,N+M\}$ can be gathered into vectors $\mathcal{U}_{L,N} \in \mathbb{R}^{2N}$ and $\mathcal{U}_{L,M} \in \mathbb{R}^{2M}$. In the case of the direction of movement alternation, the formation has to travel into the distance $\max_{i=\{1\dots n_r\}} p_i$ to place the second virtual leader on the position of the first one. Such a movement creates an *appendix* in the entire trajectory that is followed. The minimal length of this additional trajectory is defined by the size of the formation and only the curvature of the leading robot is not uniquely determined. Therefore, we have to define a vector $\mathcal{K}_{App}$ collecting variables $K_{App}(j), \forall j \in \{1,\dots,N+M-1\}$ applied for the movement of the virtual leaders in the appendixes. $N+M-1$ is maximal possible number of switchings. Usually the number of switchings is significantly lower and the unused values of $K_{App}(\cdot)$ will not be considered in the evaluation of the optimization vector. All variables describing the complete trajectory from the actual position of the first virtual leader until target region can be then collected into the unique optimization vector $\Omega_L = [\Psi_{L,N}, \mathcal{U}_{L,N}, \Psi_{L,M}, \mathcal{U}_{L,M}, \mathcal{T}^{\Delta}_{L,M}, \mathcal{K}_{App}] \in \mathbb{R}^{6N+7M-1}$.u The vector Ω_L contains necessary information for both leaders, but we have to decide, which part of the vector will be assigned to which leader. Firstly, let us define an ordered set of indexes of samples where the polarity of velocity is changed as $I_{sw} := \{i : v_L(i)v_L(i+1) < 0\}, \forall i \in \{1,\dots,N+M-1\}$, where the values of $v_L(\cdot)$ can be extracted from the vector $\mathcal{U}_{L,MN} = [\mathcal{U}_{L,N}, \mathcal{U}_{L,M}] \in \mathbb{R}^{2(N+M)}$. Now, we can propose to collect the control inputs used for the first virtual leader as $\mathcal{U}_{L_1,MN} = [\bar{u}_L(1),\ \bar{u}_L(2),\ \dots,\ \bar{u}_L(I_{sw}(1)),\ \bar{u}_{App_1}(1),\ \bar{u}_{F_1}(1),\ \bar{u}_L(I_{sw}(2)+1),\ \bar{u}_L(I_{sw}(2)+2), \dots, \bar{u}_L(I_{sw}(3)), \bar{u}_{App_1}(2), \bar{u}_{F_1}(2), \dots,\ \bar{u}_L(I_{sw}(2n_{sw_1}-2)+1),\ \bar{u}_L(I_{sw}(2n_{sw_1}-2)+2),\ \dots,\ \bar{u}_L(I_{sw}(2n_{sw_1}-1)), \bar{u}_{App_1}(n_{sw_2}), \bar{u}_{F_1}(n_{sw_2})]$, if $n_{sw_1} = n_{sw_2}$ and $\mathcal{U}_{L_1,MN} = [\bar{u}_L(1),\ \bar{u}_L(2),\ \dots, \bar{u}_L(I_{sw}(1)),\ \bar{u}_{App_1}(1), \bar{u}_{F_1}(1),\ \bar{u}_L(I_{sw}(2)+1),\ \bar{u}_L(I_{sw}(2)+2),\ \dots,\ \bar{u}_L(I_{sw}(3)),\ \bar{u}_{App_1}(2),\ \bar{u}_{F_1}(2),\ \dots,\ \bar{u}_{App_1}(n_{sw_2}),\ \bar{u}_{F_1}(n_{sw_2}),\ \bar{u}_L(I_{sw}(2n_{sw_1}-2)+1),\ \bar{u}_L(I_{sw}(2n_{sw_1}-2)+2),\ \dots,\ \bar{u}_L(N+M)]$, if $n_{sw_1} \neq n_{sw_2}$. The control inputs for the second virtual leader as $\mathcal{U}_{L_2,MN} = [\bar{u}_{F_2}(1),\ \bar{u}_L(I_{sw}(1)+1),\ \bar{u}_L(I_{sw}(1)+2),\ \dots,\ \bar{u}_L(I_{sw}(2)),\ \bar{u}_{App_2}(1),\ \bar{u}_{F_2}(2),\ \bar{u}_L(I_{sw}(3)+1),\ \bar{u}_L(I_{sw}(3)+2), \dots, \bar{u}_L(I_{sw}(4)), \bar{u}_{App_2}(2), \bar{u}_{F_2}(3), \dots,\ \bar{u}_{App_2}(n_{sw_1}-1),\ \bar{u}_{F_2}(n_{sw_1}),\ \bar{u}_L(I_{sw}(2n_{sw_2}-1)+1),\ \bar{u}_L(I_{sw}(2n_{sw_2}-1)+2), \dots, \bar{u}_L(N+M)]$, if $n_{sw_1} = n_{sw_2}$ and $\mathcal{U}_{L_2,MN} = [\bar{u}_{F_2}(1),\ \bar{u}_L(I_{sw}(1)+1),\ \bar{u}_L(I_{sw}(1)+2),\ \dots,\ \bar{u}_L(I_{sw}(2)),\ \bar{u}_{App_2}(1),\ \bar{u}_{F_2}(2),\ \bar{u}_L(I_{sw}(3)+1),\ \bar{u}_L(I_{sw}(3)+2), \dots, \bar{u}_L(I_{sw}(4)), \bar{u}_{App_2}(2), \bar{u}_{F_2}(3),\ \dots,\ \bar{u}_L(I_{sw}(2n_{sw_2}-1)+1),\ \bar{u}_L(I_{sw}(2n_{sw_2}-1)+2),\ \dots, \bar{u}_L(I_{sw}(2n_{sw_2})),\ \bar{u}_{App_2}(n_{sw_1}-1),\ \bar{u}_{F_2}(n_{sw_1})]$, if $n_{sw_1} \neq n_{sw_2}$.
The value n_{sw_1} (resp. n_{sw_2}) is number of time intervals in which the first (resp. the second) virtual leader has the leadership. Control inputs $\bar{u}_{F_1}(j), \forall j \in \{1,\dots,n_{sw_2}\}$ (resp. $\bar{u}_{F_2}(j), \forall j \in \{1,\dots,n_{sw_1}\}$) are obtained by the formation driving method when the first (resp. the second) virtual leader is the virtual follower. The control inputs $\bar{u}_{App_1}(j), \forall j \in \{1,\dots,n_{sw_1}\}$ (resp. $\bar{u}_{App_2}(j), \forall j \in \{1,\dots,n_{sw_2}-1\}$) are applied for the formation driving in the appendixes and are obtained using $\mathcal{K}_{App}$ and $v_{max,L}$.

The vectors $\mathcal{T}^{\Delta}_{L_1,MN}$ (resp. $\mathcal{T}^{\Delta}_{L_2,MN}$) will be compiled similar as the vector $\mathcal{U}_{L_1,MN}$ (resp. $\mathcal{U}_{L_2,MN}$), because the variables $\bar{u}(\cdot)$ and $\Delta t(\cdot)$ are inseparably joined. An example illustrating this approach is presented in Figure 2 where the parameters and outputs of the optimization are: $N = 0$, $M = 5$, $I_{sw} \in \{2,4\}$, $\mathcal{U}_{L_1} = [\bar{u}_L(1), \bar{u}_L(2), \bar{u}_{App_1}(1), \bar{u}_{F_1}(1), \bar{u}_L(5)]$, $\mathcal{U}_{L_2} = [\bar{u}_{F_2}(1), \bar{u}_L(3), \bar{u}_L(4), \bar{u}_{App_2}(1), \bar{u}_{F_2}(2)]$.

The trajectory planning and the static as well as dynamic obstacle avoidance problem for both virtual leaders can be transformed to the minimization of single cost function subject to sets of equality and inequality constraints. Accordingly the leader-follower concept, the trajectory computed as the result of this minimization will be used as an input of the trajectory tracking for the followers. Details of the description of both, the leader trajectory planning and the trajectory tracking for followers, can be found in our previous publication in (Saska et al., 2009), where a method with one leader enabling only forward movement is presented.

4 EXPERIMENTAL RESULTS

The results presented in this section have been obtained using the introduced algorithm with parameters: $n = 2$, $N = 4$, $M = 8$, and $\Delta t = 0.25s$. In the scenario of a road with obstacles in Figure 3 a formation of 6 robots is aimed to reach the target region in minimum time. In accordance with real application, the maximum forward and backward velocities of the virtual leader of the formation were set unsymmetrically as $v_{min,L} = -v_{max,L}/2$. Putting the final region sufficiently far behind the formation, the solution of the *formation to target zone problem* is to turn the formation and then continue forward to reach the desired area. Such a maneuver, which contains two switching between the virtual leaders, is denoted by dashed curves in Figure 3(a). The solution of the task keeps the formation outside the obstacles during the whole maneuver. Beyond this, the ability to avoid dynamic obstacles with collision course detected by rear sensors of the robots during the reverse driving is demonstrated in the simulation (see snapshots in Figure 3(b) and in Figure 3(c)). The complete history of the turning can be seen in Figure 3(d).



(a) Initial position of the formation with complete plan for both virtual leaders denoted by the dash curve.

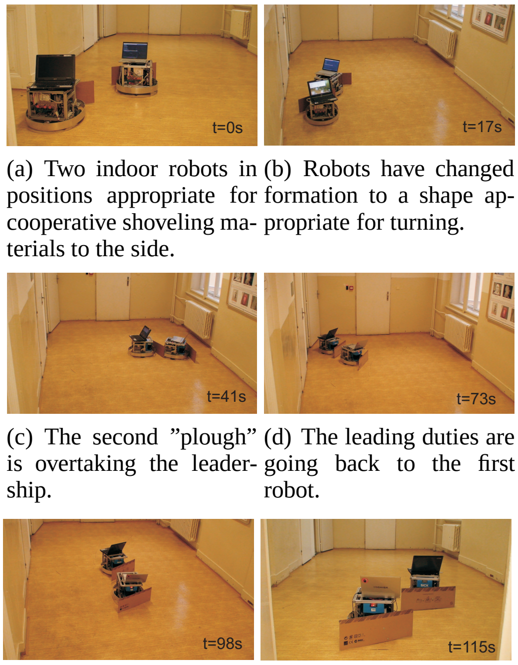
(b) Detection of movement of one obstacle and consequential reaction of the followers.

(c) The followers successfully passed by the obstacle and they are going back to their position in the formation.

(d) Accomplished task with denoted trajectories of the robots.

Figure 3: Snapshots of turning 180 degrees in the environment with static as well as dynamic obstacles.

The presented simulations have shown only simple experiments to demonstrate a basic behavior of the presented approach. More complicated scenarios proving the ability to avoid a damaged follower crossing trajectories of neighbors, to utilize different shapes of formations or to deal with local minima in a complex environment cannot be offered due to space limitation of this paper, but can be found in (Saska et al., 2009; Saska et al., 2007).



(a) Two indoor robots in positions appropriate for cooperative shoveling materials to the side.

(b) Robots have changed formation to a shape appropriate for turning.

(c) The second "plough" is overtaking the leadership.

(d) The leading duties are going back to the first robot.

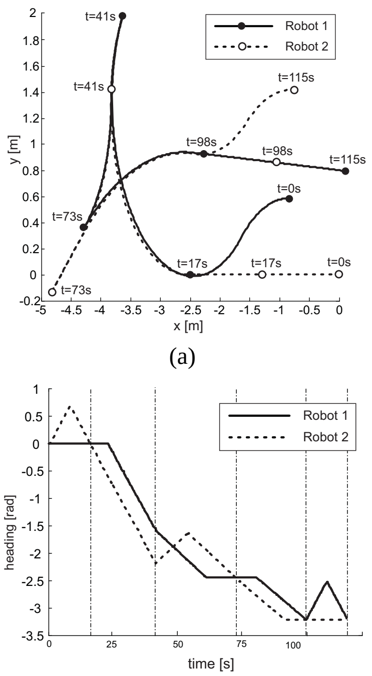
(e) The formation has finished the turning maneuver.

(f) Robots are again ready for shoveling.

Figure 4: Formation turning at the end of a blind corridor.

The presented hardware experiment is a part of airport snow shoveling project using formations of autonomous ploughs, which has been reported in (Saska et al., 2008; Hess et al., 2009). Here, we applied the approach presented in this paper for turning the formation of snow ploughs at the end of cleaning runway. The experiment has been performed on G2Bot robotic platform equipped with SICK laser range finder (not utilized in the experiment) and odometry.

In a simplified version of the experiments (see snapshots in Figure 4, data from odometry in Figure 5 and a movie of the experiment in (MOVIE, 2010)), two autonomous indoor robots are facing a blind corridor with an aim to turn at the end. The map of the environment and positions of the vehicles are known before the mission. During the experiment, positions are updated using a dead reckoning because an exter-



(a)

(b)

Figure 5: Data captured from odometry of both ploughs with depicted times of snapshots in Figure 4.

nal positioning system is not necessary in such a short term experiment. The actual position of robots and their plans are shared via wireless communication. The plan of robots is as follows: 1) build a compact formation appropriate for turning, 2) turn 180 degrees and 3) return back to the shoveling formation. Such an approach is necessary in a case of larger formations covering the whole runway and it can also better clarify the method employing two virtual leaders. The transitions between the different formations as well as the turning maneuver are computed automatically using the methods presented in this paper. Only the positions of the vehicles within the formations (safety distances, required overlapping of shovels etc.) are given by experts.

5 CONCLUSIONS

In this paper, we have presented an RHC approach to formation driving of car-like robots reaching a target area in environments with obstacles. The proposed approach is novel in the sense of utilized pair of complementary virtual leaders necessary for guidance of the formation during backward driving. This allows us arbitrary maneuvers of formations including the U-turn in bordered environment, which is necessary for complete autonomous airport snow shoveling system. The applicability of the presented approach has been verified with simulations and hardware experiment.

ACKNOWLEDGEMENTS

This work is supported by the Czech Technical University under the grant $SGS10/195/OHK3/2T/13$ and by grants $2C06005$ and $MSM6840770038$.

REFERENCES

Alamir, M. (2006). *Stabilization of Nonlinear Systems Using Receding-Horizon Control Schemes*, volume 339 of *Lecture Notes in Control and Information Sciences*. Springer, Berlin / Heidelberg, Germany.

Barfoot, T. D. and Clark, C. M. (2004). Motion planning for formations of mobile robots. *Robotics and Autonomous Systems*, 46:65–78.

Dunbar, W. and Murray, R. (2006). Distributed receding horizon control for multi-vehicle formation stabilization. *Automatica*, 42(4):549–558.

Franco, E., Magni, L., Parisini, T., Polycarpou, M., and Raimondo, D. (2008). Cooperative constrained control of distributed agents with nonlinear dynamics and delayed information exchange: A stabilizing receding-horizon approach. *IEEE Transactions on Automatic Control*, 53(1):324–338.

Hess, M., Saska, M., and Schilling, K. (2009). Application of coordinated multi vehicle formations for snow shoveling on airports. *Inteligent Service Robotics*, 2(4):205 – 217.

Leitmann, G. (1980). Guaranteed avoidance strategies. *Journal of Optimization Theory and Applications*, 32:569–576.

Mayne, D. Q., Rawlings, J. B., Rao, C. V., and Scokaert, P. O. M. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814.

MOVIE (2010). Two indoor ploughs turning 360 degrees [online]. [cit. 2010-1-13]. *http://imr.felk.cvut.cz/ demos/videos/files/shoveling.avi*.

Saska, M., Ferenczi, I., Hess, M., and Schilling., K. (2007). Path planning for formations using global optimization with sparse grids. In *In Proceedings of The 13th IASTED International Conference on Robotics and Applications (RA 2007).*, Wuerzburg, Germany.

Saska, M., Hess, M., and Schilling, K. (2008). Efficient airport snow shoveling by applying autonomous multi-vehicle formations. In *Proceedings of IEEE International Conference on Robotics and Automation*, Pasadena, USA.

Saska, M., Mejia, J. S., Stipanovic, D. M., and Schilling, K. (2009). Control and navigation of formations of car-like robots on a receding horizon. In *Proc of 3rd IEEE Multi-conference on Systems and Control*.

SMA CONTROL FOR BIO-MIMETIC FISH LOCOMOTION

Claudio Rossi, Antonio Barrientos
Robotics and Cybernetics Research Group, Universidad Politécnica de Madrid, Madrid, Spain
{claudio.rossi, antonio.barrientos}@upm.es

William Coral Cuellar
Dep. Génie Électrique et Sistèmes de Commande, Université de Technologie de Belfort-Montbéliard, Belfort, France
William.Coral@utbm.fr

Keywords: Shape memory alloys, Biologically-inspired robots, Smart actuators control, Underwater robotics.

Abstract: In this paper, we describe our current work on bio-inspired locomotion systems using smart materials. The aim of this work is to investigate alternative actuation mechanisms based on smart materials, exploring the possibility of building motor-less and gear-less robots. A swimming underwater robot is being developed whose movements are generated using such materials, concretely Shape Memory Alloys. This paper focuses on the actuators control in order to obtain a a sufficiently fast and accurate positioning.

1 INTRODUCTION

Robotics actuator technology is basically dominated by two kind of actuators: electric motors/servomotors and pneumatic/hydraulic actuators. In mobile robotics, the former is mostly used, with exceptions being e.g. large legged robots. The (rotatory) motion of the motors is then transmitted to the effectors through gearboxes, belts and other mechanical devices in the case that linear actuation is needed. Although applied with success in uncountable robotic devices, such systems can be complex, heavy and bulky[1]. In underwater robots, propellers are most used for locomotion an maneuvering. Propellers however may have problems of cavitation, noise, efficiency, can get tangled with vegetation and other objects and can be dangerous for sea life.

Underwater creatures are capable of high performance movements in water. Thus, underwater robot design based on the mechanism of fish locomotion appears to be a promising approach. Over the past few years, researches have been developing underwater robots based on underwater creatures swimming mechanism (Hu, 2006), (Anderson and Chhabra, 2002), (Morgansen et al., 2007). Yet, most of them still rely on servomotor technology and a structure made of a discrete number of elements. One of the most advanced fishe-like robot is the MIT fish (Valdivia y Alvarado and Youcef-Toumi, 2006). This fish has a continuous soft body. A single motor generates a wave that is propagated backwards in order to generate propulsion.

In the last years, actuation technology in active or ”smart” materials has opened new horizons as far as simplicity, weight and dimensions. New materials such as piezo-electric fiber composite, electro-active polymers and shape memory alloys (SMA) are being investigated as a promising alternative to standard servomotor technology. The potential gain in weight and dimension would allow building lighter and smaller robots, and even devising soft-bodied robots (Cowan and Walker, 2008).

In order to reproduce the undulatory body motion of fishes, smart materials appear to be extremely suited. In fact, over the last years, there has been an increasing activity in this field. Research in the field of smart materials for underwater locomotion is focused into mechatronics design and actuators control. As far as mechatronic design, much work is devoted to building hydrofoils using, e.g. piezo-electric fiber composite (Ming et al., 2009), embedding SMA wires into an elastic material such as silicone (Wang et al., 2008) or using SMAs as linear actuators (Rediniotis et al., 2002). An important challenge is the control

[1] Robotuna, a robot fish developed at MIT in 1994, had 2,843 parts controlled by six motors (font: MIT News, http://web.mit.edu/newsoffice/2009/robo-fish-0824.html)

of such materials. In the case of SMAs, excellent results have been achieved by (Teh, 2008) and (Meier et al., 2009). In this paper, we present our work on a swimming robot, focussing on the control of the SMA-based actuation system.

2 MECHATRONICS DESIGN

Fishes can swim bending their body in such a way to produce a backward-propagating propulsive wave. Such bending comes in different ways. Anguilliform swimmers show a snake-like motion: their body can be divided into numerous segments from head to tail and can reproduce at least one complete wavelength along the body. Conversely, subcarangifor, carangiform and thunniform swimmers only bends the second half of the body (roughly) and the number of segments is reduced to one or two.

For our model, we have chosen to imitate the subcarangiform swimming style because of the reduced number of segments w.r.t. anguilliform fishes, which simplifies the study and the implementation, while having enough degrees of freedom that allow complex motion patterns to be reproduced. Our fish model can also bend the front part of its body, which makes a total of three bendable segments (cf. Figure 1).

The fish is formed by a continuous structure made of polycarbonate 1 mm thick, which represents the fish backbone and spines. This material has been chosen for its temperature resistance, impact resistance and flexibility. Additional supporting structure made of PVC is employed to support the silicon-based skin of the robot. The overall length of the fish is 30 cm (not including the caudal fin). Along the backbone, six SMA actuators are used to bend the body. Their length is $1/3$ body length (i.e. $8.5cm$, not counting the caudal fin and the head) and are positioned in pairs, in such a way to produce an antagonistic movement. Thanks to this arrangement, the body segments can bend up to ≈ 30 degrees. The diameter size of the wires has been chosen as a trade-off between current consumption, pull force and contraction time. We have adopted a SMA with a diameter size of $150\mu m$ that has a pull force of 230 grams, a consumption of 250 mA at room temperature, and a nominal contraction time of 1 second. Such contraction time allows an undulation frequency that is enough for producing motion in water.

2.1 Shape Memory Alloys

SMAs are materials capable of changing their crystallographic structure (from *austensite* and *marteniste* phases), due to changes in temperature. When an SMA wire is subjected to an electrical current, Joule resistive heating causes the SMA actuator to contract. SMAs have the advantage that they work at low currents and voltages, are extremely cheap and are easily available commercially. Nitinol, one of the most commercially available SMAs, is an alloy of nickel and titanium (NiTi). It is characterized by a high recovery stress ($> 500MPa$), low operational voltage ($4-5\ V$), a reasonable operational strain ($\approx 4\%$) and a long life (up to 10^6 cycles).

The behavior of SMAs is more complex than many common materials: the stress-strain relationship is non-linear, hysteretic, exhibits large reversible strains, and it is temperature dependent. For this reason, a low-level control electronics has to be designed in order to have a position control close-loop precise enough for the application at hand. An important characteristics of SMAs is that they can also be used as sensors. In fact, once heated applying a given current, one can measure their resistance and calculate the actual percentage of shrinking. This measurement can be used as feedback for achieving precise position control.

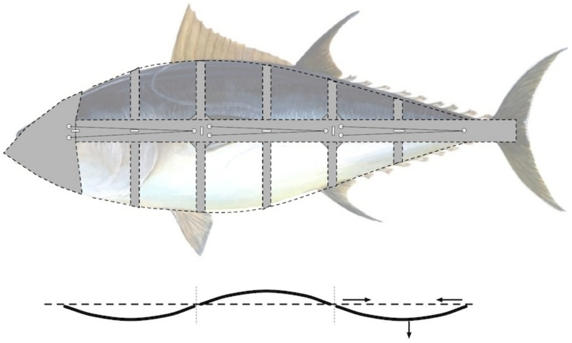


Figure 1: Lateral and upper view of the deformable structure. Note the location of the SMA wires along the body.

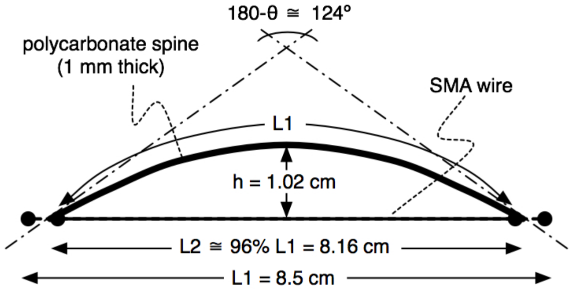


Figure 2: Principle of the bendable structure. The SMA wire is parallel to the spine segment. As it contracts, it causes the polycarbonate strip to bend.

3 SMA CONTROL

The control accuracy of smart actuators such as SMAs is limited due to their inherent hysteresis nonlinearities (see Figure 3) with a local memory. The existence of minor loops in the major loop because of a local memory also makes the mathematical modeling and design of a controller difficult for SMA actuators. Therefore, to enhance the controllability of a smart actuator, the Preisach hysteresis model (Visintin, 1995) has emerged as an appropriate behavioral model. Nevertheless, the modeling is difficult and the model equation remains complex. So even though this model is commonly used (Choi et al., 2004), the use of a heat transfer model and sensor hardware has also been proposed.

As pointed out earlier, SMAs provide the possibility to develop controller systems without sensor hardware. The detection of inner electrical resistance allows to regulate the actuator movement (Ikuta et al., 1988). The method consists in measuring the electrical resistance of an SMA element (Teh, 2008) as a form of temperature measurement. An advantage of this method is that the hysteresis on the resistance curve is smaller than the hysteresis on the temperature curve, which makes the linear approximation more accurate.

The maximum contraction of the wire can be measured as

$$\Delta L_{A_f} = \frac{R_{SMA_{Mf}} - R_{SMA_{Af}}}{L_R}, \qquad (1)$$

where $R_{SMA_{Mf}}(\Omega)$ is the SMA resistance in martensite finish temperature (relaxed SMA), $L_R(\Omega/m)$ is the linear resistance and $R_{SMA_{Af}}(\Omega)$ is the resistance at austenite finish temperature (i.e. at maximum contraction).

Figure 5 shows the performance of a mock-up for a current of 350mA. The angle shown in the figure is in good accordance with the theoretical value of 28° (cf. Figure 2). Note that the wires' speed and strain contraction depends on how fast and by how much the wire temperature is increased. In our tests, we have verified that SMAs wires can be fed with a current of up to 500mA without compromising their behavior, achieving faster response and higher percentages of contraction.

3.1 Controller Tuning

In order to tune the control system, we set up a mock-up of a segment of the fish's backbone, corresponding to a $10 \times 2cm$ stripe of 1mm thick polycarbonate, with a 174mm long SMA wire in a V-shaped configuration, in order to double the pull force (see Figure 4).

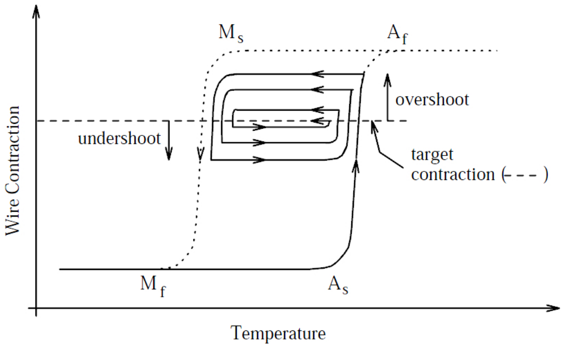


Figure 3: Histeresis of the SMA. (A_s, the austenite start temperature; A_f, the austenite finish temperature; M_s, the martensite start temperature; and M_f, the martensite finish temperature).

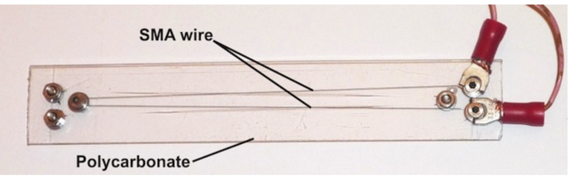


Figure 4: The test mock-up.

For the control, we used a PID (proportional-integral-derivative) controller, which responds to the equation

$$u(t) = K_p e(t) + K_i \int_0^t e(t)dt + K_d \frac{de(t)}{dt}, \qquad (2)$$

where $e(t)$ is the signal error and $u(t)$ is the control input of the process. K_p, K_i, K_e are the proportional, integrative and derivative gains. Then by posing $K_i = \frac{K_p}{T_i}, K_d = K_p T_d$, where T_i is the integral time constant and T_d is the derivative time constant, the PID controller can be written in the s domain as

$$U(s) = K_p \left[1 + \frac{1}{T_i s} + T_d s\right] E(s). \qquad (3)$$

Following Ziegler-Nichols, we have tuned the values to the three parameters (K_p, T_i, T_d) of the PID controller based on the analysis of the open and close loop of the system to be controlled. The dynamics behavior of the system is defined using the following first-order linear transfer function:

$$G(s) = \frac{K_0 e^{-s\tau_0}}{1 - s\gamma_0}, \qquad (4)$$

where the coefficients K_0, τ_0 and γ_0 are obtained from the response of the open loop system to a step input. Starting from the stabilized system at $y(t) = y_0$ to $u(t) = u_0$, a step input is applied from u_0 to u_1.

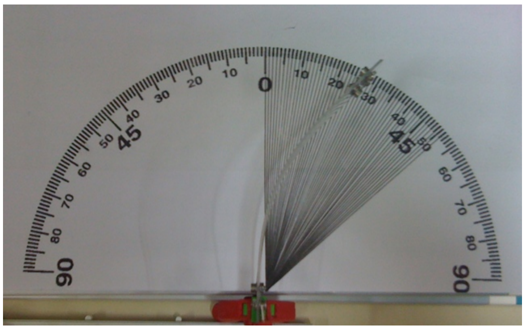


Figure 5: Angle achieved of $\approx 28°$ at a $350mA$ current.

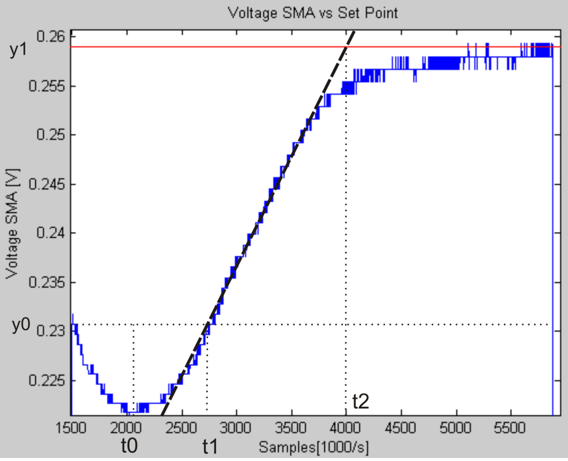


Figure 6: Voltage SMA vs. a given set point. $y_1 = 0.258, y_0 = 0.232, t_1 = 2.725 t_0 = 2.2 t_2 = 4 u_1 = 1 u_0 = 0$, experimentally determined.

The output is registered from transition to stabilization stage.

From the experimental data corresponding to Figure 6, we can compute the following parameters:

$$\tau_0 = t_1 - t_0, \gamma_0 = t_2 - t_1, K_0 = (y_1 - y_0)/(u_1 - u_0).$$

According to Ziegler-Nichols, the relations between these coefficients and the controller parameters are:

$$K_p = (1.2\gamma_0)/(K_0\tau_0), \quad T_i = 2T_0, \quad T_d = 0.5.$$

Then, according to the Z transform the discrete PID controller is:

$$U(z) = E(z)K_p \left[1 + \frac{T}{T_i(1-z^{-1})} + T_d \frac{1-z^{-1}}{T}\right]. \quad (5)$$

Also:

$$\frac{U(z)}{E(z)} = a + \frac{b}{1-z^{-1}} + c(1-z^{-1}). \quad (6)$$

Where:

$$a = Kp, \quad b = \frac{K_p T}{T_i}, \quad c = \frac{K_p T_d}{T}.$$

The parameters obtained were: $\tau_0 = 0.525, \gamma_0 = 1.276, K_0 = 0.03, K_p = 97.22, T_i = 1.05, T_d = 0.2625$. The discrete controller parameters are calculated based at the time $T = 0.1 < \tau_0/4$. Finally, $a = 97.22, b = 1.05, c = 255.202$. Figure 6 shows the answer to a $260mv$ step input, that corresponds to $300mA$ of SMA arousal. We used this value for tuning the PID controller because it allows a stable SMA response in open loop.

3.2 SMA Control Electronics

The control electronics is composed by a micro controller where the basic PID equations are encoded. The digital output of the PID controller is converted to DC in two steps: first, it is used to set the parameters of a PWM signal, and second, the PWM signal is converted to DC. Then, a Voltage Controlled Current Source (VCCS) transforms the DC voltage in a constant current that feeds the SMA. This stage has been designed to have a low power consumption ($< 20mA$). The current that passes trough the SMA when it is resting is less than $1.1\mu A$. Figure 7 shows the control block diagram and the electronics. The PWM duty cycle goes from 5% to 95% with a $1kHz$ frequency in order to avoid an excessive current on the SMA. The response of the SMA (V_{SMA} and I_{SMA}) is fed back to the micro controller, where it is converted to digital signal by the integrated A/D conversion hardware. Since the hardware used (18F458 PIC) has a 10-bits A/D converter and the maximum voltage measured at the SMA is $V_{SMA} = 0.55V$, we obtain a resolution of $0.537mV$.

Thus, taking into consideration the maximum current through the wire ($500mA$) we can measure the SMA resistances variations of $1.074m\Omega$. Since the maximum variation in the SMA length is $0.34cm$ and the maximum variation of the resistance is 1.6Ω, the system theoretical error on the SMA length is 0.067%. i.e. $0.12mm$.

A problem associated to these kind of designs is the integral windup, which can provoke long overshoot periods, encouraged by the excessive values that reaches the control signal due to the accumulation in the integrator. In order to avoid this problem and accelerate the heating, (Teh, 2008) and (Meier et al., 2009) proposed to feed the SMA with a high pulse of current. However, while in (Teh, 2008) such pulse is maintained throughout the SMA excitation time, our control systems sends a high pulse of current only for a small period of time, precisely $500mA$ for $150mS$. This step is a pre-heating stage. Then, it feeds the SMA with a ramp from 10% of the target current to 100% of the target current (Figure 8). The advantage

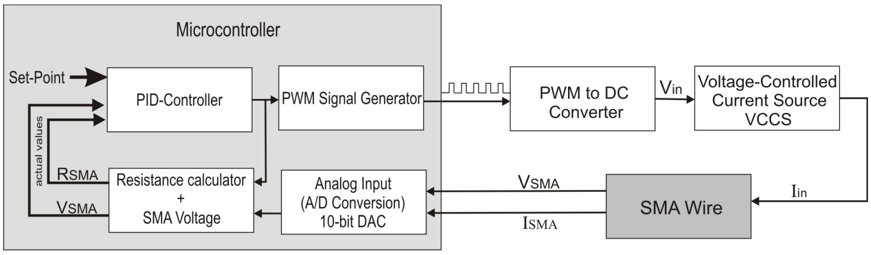


Figure 7: Control loop of a single SMA actuator.

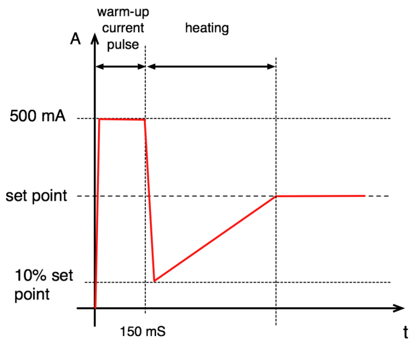


Figure 8: Control strategy: pre-heating and heating.

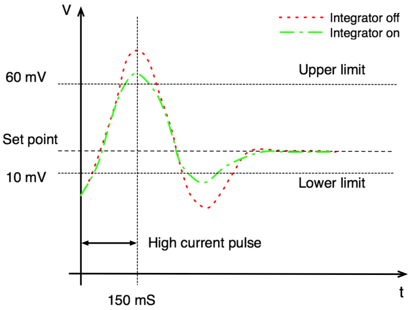


Figure 9: Behavior of the control signal.

of this approach is that the wire supports less stress and the energy consumption is lower.

Moreover, to reduce the over and undershoot, the voltage was limited between a minimum and maximum value of $10mV$ and $60mV$, respectively, making the integrator act only when these limits are overcome (cf. Figure 9). Such values have been determined experimentally.

4 EXPERIMENTAL RESULTS

Using the mock-up mentioned above, we have tested the control system by applying different target resistances (i.e. desired lengths). Given the small displacements, it is very difficult to measure with sufficient precision the final length of the wire. Therefore, in order to assess the actual precision of the system we have measured the final resistance of the SMA wire.

Figure 10 shows the comparison of the transient period for different currents, ranging form $300mA$ to $500mA$. The plots have been post-processed in order to improve readability. The overshoot that can be noticed in the plots is produced by the initial pulse. Note how, thanks to the action of the integrator, the overshoot is reduced w.r.t the behavior of the SMA without it.

Table 1 summarizes the results for various set points (target resistance) and the resulting bending error. The length error has been calculated as function of the the resistance error (cf. Eq. 1). We refer to bending as the angle between the tangent between to the arc and the chord. As it can be noticed, the precision obtained is satisfactory, although higher than the one theoretically achievable. Only for one value (corresponding to the smallest strain) the precision obtained was not satisfactory, since it induces a bending error of almost 6 degrees. The error was expected to be smaller for higher temperatures, due to the hysteresis in the resistanceas, also reported in (Meier et al., 2009). For all the other values, on the practical point of view the precision obtained is satisfactory for the purposes of our mechatronic system, since such error translates into a bending error which has a negligible impact for the mechanics of swimming.

Table 1: Summary of the performances of the control on the SMA's resistance and on the bending structure.

Target resist. (Ω)	Target length (mm)	Resist. error (Ω)	Length error (mm)	Bend error (deg.)
12.12	173.1	0.095 (0.8%)	1.385	5.99
11.86	169.4	0.086 (0.7%)	1.186	2.81
11.72	167.4	0.019 (0.2%)	0.335	0.69
11.46	163.7	0.057 (0.5%)	0.819	1.36

5 CONCLUSIONS

The objective of this work is to investigate the use of active materials such as SMAs as alternative actuation

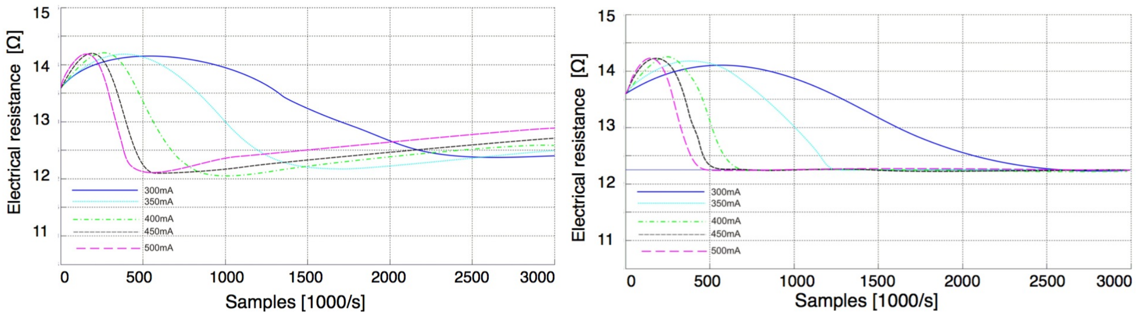


Figure 10: Comparison of the transient period for various currents, without (left) ant with the control (right).

system for bio-inspired locomotion.

Thanks to the proposed control system we have achieved a position control for the SMAs that is accurate enough for the application at hand, since produces an error that is negligible as far as undulatory motion of the fish. Nevertheless, for applications where more precision is needed (e.g. manipulation) more accurate controllers are needed. As far as reaction speed is concerned, the nominal contraction time can be reduced to almost one half by a carefully designed control, as other authors have demonstrated.

The main advantage of our mechatronics design is its simplicity, extremely light weight and practically null volume of the actuators, at advantage of the available payload. Moreover, the actuators are absolutely silent and do not produce any vibration, a feature which can be exploited, e.g. in the observation of sea wild life since the robot would not disturb.

In conclusion, smart actuators and flexible continuous structures can be a promising field for making alternative robots, that are simpler and lighter, and that can have interesting application domains.

ACKNOWLEDGEMENTS

This work has been partially funded by projetct Robocity II, S2009/DPI-1559 of the Madrid Autonomic Government.

REFERENCES

Anderson, J. M. and Chhabra, N. K. (2002). Maneuvering and stability performance of a robotic tuna. *Integrative and Comparative Biolog*, 42(1):118–126.

Choi, B.-J., Lee, Y.-J., and Choi, B.-Y. (2004). Fast preisach modeling method for shape memory alloy actuators using major hysteresis loops. *Smart Mater. Struct.*, 23:1069–1080.

Cowan, L. and Walker, I. (2008). Soft continuum robots: The interaction of discrete and continuous elements. In *Artificial Life XI*, pages 126–133.

Hu, H. (2006). Biologically inspired design of autonomous robotic fish at essex. In *IEEE SMC UK-RI Chapter Conference, on Advances in Cybernetic Systems*, pages 3–8.

Ikuta, K., Tsukamoto, M., and Hirose, S. (1988). Shape memory alloy servo actuator system with electric resistance feedback and application for active endoscope. In *IEEE International Conference on Robotics and Automation*, pages 423–430.

Meier, H., Czechowicz, A., and Haberland, C. (2009). Control loops with detection of inner electrical resistance and fatigue-behaviour by activation of niti - shape memory alloys. In *European Symposium on Martensitic Transformations*.

Ming, A., Park, S., Nagata, Y., and Shimojo, M. (2009). Development of underwater robots using piezoelectric fiber composite. In *IEEE International Conference on Robotics and Automation*.

Morgansen, K., Triplett, B., and Klein, D. (2007). Geometric methods for modeling and control of free-swimming fin-actuated underwater vehicles. *IEEE Transactions on Robotics*, 23(6):1184–1199.

Rediniotis, O. K., Wilson, L. N., Lagoudas, D. C., and Khan, M. M. (2002). Development of a shape-memory-alloy actuated biomimetic hydrofoil. *Journal of Intelligent Material Systems and Structures*, 13(1):35–49.

Teh, Y. H. (2008). An architecture for fast and accurate control of shape memory alloy actuators. *International Journal of Robotics Research*, 27(5):595–611.

Valdivia y Alvarado, P. and Youcef-Toumi, K. (2006). Design of machines with compliant bodies for biomimetic locomotion in liquid environments. *ASME Journal of Dynamic Systems, Measurement, and Control*, 128:3–13.

Visintin, A. (1995). *Differential Models of Hysteresis*. Applied Mathematical Sciences. Springer, Berlin.

Wang, Z., Hang, G., Wang, Y., Li, J., and Du, W. (2008). Embedded sma wire actuated biomimetic fin: a module for biomimetic underwater propulsion. *Smart Materials and Structures*, 17(2):25–39.

EJS+EJSRL: A FREE JAVA TOOL FOR ADVANCED ROBOTICS SIMULATION AND COMPUTER VISION PROCESSING

Carlos A. Jara, Francisco A. Candelas, Jorge Pomares, Pablo Gil and Fernando Torres
Physics, Systems Engineering and Signal Theory Department, University of Alicante, Spain
{carlos.jara, francisco.candelas, jpomares, pablo.gil, fernando.torres}@ua.es

Keywords: Modelling, Robotics simulation, Virtual reality.

Abstract: This paper presents a free Java software platform which enables users to easily create advanced robotic applications together with image processing. This novel tool is composed of two layers: 1) Easy Java Simulations (EJS), an open-source tool which provides support for creating applications with a full 2D/3D interactive graphical interface, and 2) EjsRL, a high-level Java library specifically designed for EJS which provides a complete functional framework for modeling of arbitrary serial-link manipulators and computer vision algorithms. The combination of both components sets up a software architecture which contains a high number of functionalities in the same platform to develop complex simulations in robotics and computer vision fields.

1 INTRODUCTION

Robotics and Computer Vision (R&CV) systems have highly complex behaviours. For this reason, throughout the last two decades there has been a strong development of simulation tools devoted to R&CV systems. Some of these tools have been designed for professional applications, while others for educational and research purposes. In the field of industrial Robotics, several graphical software environments as for example Easy-ROB3D (Easy-ROB3D, 2004), have been created in the form of stand-alone business packages for well defined problems. These are powerful tools, but some of them lack of resources in some aspects for higher education. Otherwise, numerous open-source tools such as GraspIt (Pelossoft et. al, 2004), RoboMosp (Jaramillo et al., 2006) and Microsoft Robotics Studio (Jackson, 2007), overcome these deficiencies. Other open-source tools are in the form of toolboxes such as SimMechanics (SMC) (Babuska, 2005), RobotiCad (RBC) (Falconi and Melchiorri, 2008) and Robotics Toolbox for Matlab (Corke, 1996). With regard to Computer Vision tools, several libraries have been developed for education and research, such as the Open Computer Vision Library (OpenCV, 2001) and VXL (VXL, 2001), developed in C++ language, and Java Advanced Imaging (JAI, 2004), written in Java.

However, the majority of the above commented tools are independent software platforms which have been developed in a separated way. This feature represents a drawback when time comes to develop complex models which combine R&CV systems. Perhaps, only Robotics/Vision Matlab toolboxes provide a set of functions suitable for synthesis and simulation which can be programmed under the same environment. Nevertheless, both toolboxes do not provide a user-friendly graphical interface support for both creating a personalized application and building 3D virtual environments. Thus, educators and researches have to spend time and effort searching the suitable libraries and they must have programming skills to develop the application.

The approach presented in this paper is a new tool called EJS+EjsRL, which provides a complete functional framework for modeling and simulation of R&CV systems, all embedded in the same toolbox. In addition, this software platform gives full 2D and 3D graphical support both for creating user interfaces and complex robotic environments with computer vision algorithms in an easy and simplified way. The main novel feature of this approach is that its software architecture contains a higher number of functionalities in the same platform than the existing software applications for that purpose (see table 1). Most of these functionalities are included as high-level tools, with the advantage of allowing users to easily create R&CV applications with a minimum of programming. The tool presented contains several features for Robotics such as kinematics, programming, dynamics, world modeling, importation of 3D model files, etc. and a higher number of Computer Vision algorithms than the JAI.

Table 1: Feature comparison with other toolboxes.

Feature	SMC	RBC	Matlab toolbox	EJS+ EjsRL
Kinematics	●	●	●	●
Dynamics	●	●	●	●
Programming		●	●	●
Importation VRML/OBJ			●	●
World Modeling				●
Computer Vision			●	●
Interface design				●
Software connection			●	●

Another meaningful problem is the platform dependency. Some C++ tools are not portable for all the operating systems. EJS+EjsRL is based on Java, a well-known programming language which is platform independent.

The remainder of this paper is organized as follows: section 2 describes the overall software architecture of the platform. Section 3 shows a complete application design. Section 4 shows other advanced features of the system. Section 5 shows the simulation capabilities of EJS+EjsRL by means of several test cases. Finally, some conclusions are discussed in section 6.

2 SYSTEM DESCRIPTION

2.1 Components

There are two main blocks that represent the functional core of this software platform: an object-oriented Java library (EjsRL) which allows users to model both arbitrary serial-link robots and computer vision algorithms, and *Easy Java Simulations* (EJS), powerful software for developing simulations. The combination of both tools (EJS+EjsRL) permits to easily and quickly create R&CV simulations.

EJS is a freeware, open-source tool developed in Java, specifically created for the creation of interactive dynamic simulations with higher graphical support (Esquembre, 2004). EJS has been designed for people who do not need complex programming skills. In order to develop a simulation, the user only provides the most relevant core of the algorithm and EJS automatically generates all the Java code needed to create a complete interactive application. There are a lot of applications which have been developed with EJS for research and teaching activities (Jara et al., 2008; Jara et al., 2009).

EjsRL is a Java library specifically designed for EJS which provides a complete functional framework that enables it to model and design advanced R&CV applications. All the components belonging to this software layer have been structured and organized in an object-oriented form. Figure 1 shows a simplified class diagram of EjsRL, specifying the most important packages and classes. For a complete description of all the classes, readers can visit the web page: http://www.aurova.ua.es/rcv.

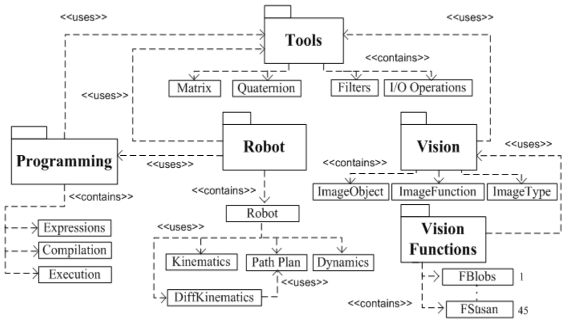


Figure 1: Package and class diagram of EjsRL.

2.2 Software Architecture

The software design is based on a hierarchical coordination between EJS and EjsRL. Each of them is divided into systems which must interchange data in order to develop R&CV simulations (figure 2).

A specific simulation within the EJS' environment must include the definition of the model and the definition of the view or graphical interface (figure 2). In order to describe the model, users must write the differential equations that establish how these variables change in time. For this last step, EJS offers two options. The first is a built-in editor of Ordinary Differential Equations (ODEs) in which users write the system equations in a similar way to how they would write on a blackboard. Users can choose different standard algorithms to numerically solve them (Euler, Runge-Kutta, etc.). The second facility is a connection with Matlab/Simulink that lets users to model systems with the help of these tools (Sanchez et al., 2005) (see section 5). In relation to the view, EJS provides a set of standard Java Swing, Java 2D and Java 3D components to build the interface in a simple drag-and-drop way. In addition, VRML and OBJ extern graphic files can be imported to the view. These graphical components have certain properties that the user can connect with the model variables and set a link between the model and the view. Therefore, the simulation turns into an interactive application where users can change the model variables and observe the simulation behaviour in the view.

There are three important modules which define the most high-level API of EjsRL (figure 2): Robotics, Matrix Computation and Computer Vision. This library works as an external interface to give to model variables of EJS the corresponding value in order to create R&CV applications.

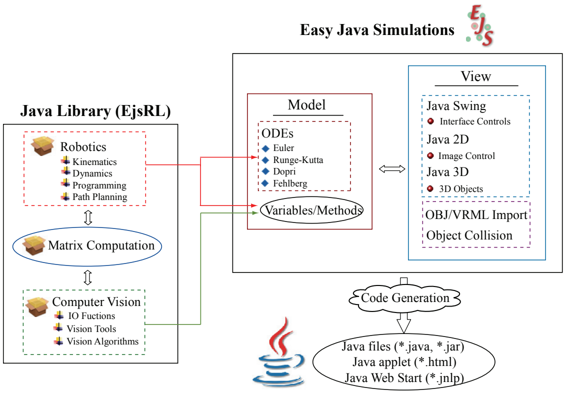


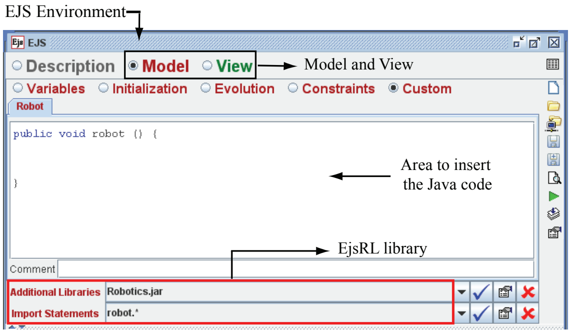
Figure 2: Software architecture of EJS+EjsRL.

3 DESIGN OF A ROBOTICS APPLICATION

3.1 Creating the Robot Arm and its Workspace

The first step in order to create a robotic simulation is to execute EJS and to insert the library EjsRL as external resource (figure 3). In this way, all the methods and classes of EjsRL can be used within EJS' environment. Secondly, it is necessary to create a specific robot in the model part. This action implicates to define the variables and to program a robot object specifying a minimum code.

For creating an arbitrary robot arm object, users only have to know its Denavit-Hartenberg parameters, its physical features and the type of joints. With these data, a Java object variable defined in the EJS' environment has to be initialized using the robot's constructor of the Robotics module of EjsRL. Figure 4 shows the Java code which must be inserted in the model of EJS' environment and the necessary variables to program a robot of 6 rotational DOF.

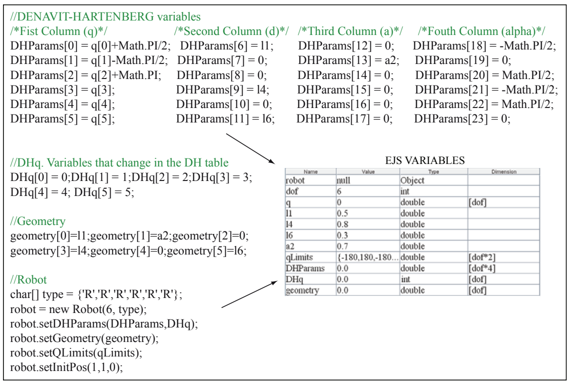


```
public void robot () {


}
```

Additional Libraries: Robotics.jar

Import Statements: robot.*

Figure 3: Overview of the EJS' environment



```
//DENAVIT-HARTENBERG variables
/*Fist Column (q)*/                /*Second Column (d)*/   /*Third Column (a)*/   /*Fouth Column (alpha)*/
DHParams[0] = q[0]+Math.PI/2;   DHParams[6] = l1;       DHParams[12] = 0;      DHParams[18] = -Math.PI/2;
DHParams[1] = q[1]-Math.PI/2;   DHParams[7] = 0;        DHParams[13] = a2;     DHParams[19] = 0;
DHParams[2] = q[2]+Math.PI;     DHParams[8] = 0;        DHParams[14] = 0;      DHParams[20] = Math.PI/2;
DHParams[3] = q[3];             DHParams[9] = l4;       DHParams[15] = 0;      DHParams[21] = -Math.PI/2;
DHParams[4] = q[4];             DHParams[10] = 0;       DHParams[16] = 0;      DHParams[22] = Math.PI/2;
DHParams[5] = q[5];             DHParams[11] = l6;      DHParams[17] = 0;      DHParams[23] = 0;

//DHq. Variables that change in the DH table
DHq[0] = 0;DHq[1] = 1;DHq[2] = 2;DHq[3] = 3;
DHq[4] = 4; DHq[5] = 5;

//Geometry
geometry[0]=l1;geometry[1]=a2;geometry[2]=0;
geometry[3]=l4;geometry[4]=0;geometry[5]=l6;

//Robot
char[] type = {'R','R','R','R','R','R'};
robot = new Robot(6, type);
robot.setDHParams(DHParams,DHq);
robot.setGeometry(geometry);
robot.setQLimits(qLimits);
robot.setInitPos(1,1,0);
```

EJS VARIABLES

Name	Value	Type	Dimension
robot	null	Object	
dof	6	int	
q	0	double	[dof]
l1	0.5	double	
l4	0.8	double	
l6	0.3	double	
a2	0.7	double	
qLimits	{-180,180,-180...	double	[dof*2]
DHParams	0.0	double	[dof*4]
DHq	0.0	int	[dof]
geometry	0.0	double	[dof]

Figure 4: Java code for modeling a 6 rotational DOF robot.

After programming the robot object, the next step is to develop the interface or view for the final user. As stated, EJS provides a set of components to build the interface in a simple drag-and-drop way. In the case of a robotic simulation, the interface can be composed by the 3D solid links of the robot and its workspace, and other standard components to control the application (panels, buttons, sliders, plots, etc.). Figure 5 shows the construction of the interface for the example proposed. The component *drawingPanel3D* is the 3D environment where the robot and its workspace will be displayed. Here, it is defined each one of the 3D links of the robot by means of the VRML component, which allows to import models from existent VRML files. As mentioned, all the interface components of EJS have certain properties which are used for the simulation. Figure 5 shows the properties of the VRML component (*Position and Size*, *Visibility and Interaction* and *Graphical Aspect*). The position and transform fields will be used to move the robot since they will be connected with the model variables which define the robot. Figure 5 also shows a dialog

(*Move Joints*) where some sliders controls (*q1…q6*) have been added from the view components. These sliders are connected with the *q* variable of the robot model (see figure 4).

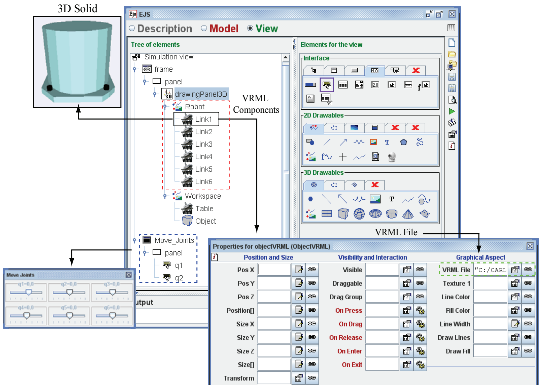
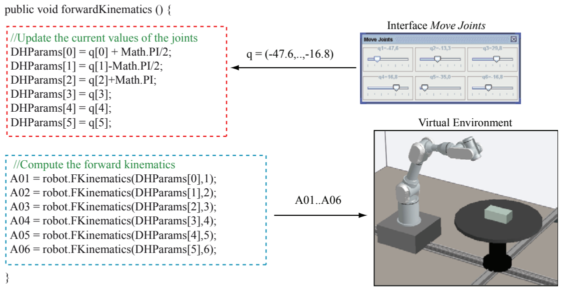


Figure 5: Interface construction of a Robotics application.

3.2 Kinematics and Path Planning Simulation

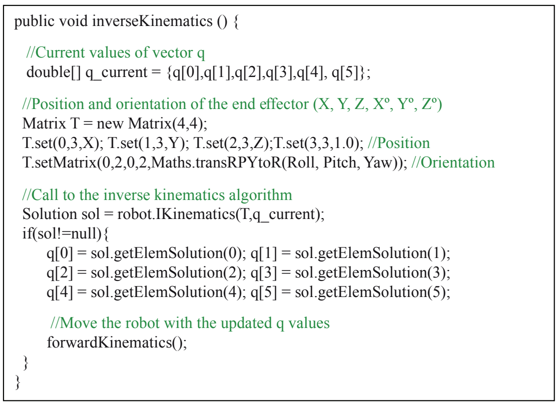
The implementation of the forward kinematics can be easily programmed and simulated with EJS+EjsRL. Figure 6 shows the Java code to resolve the forward kinematics of the robot proposed. The joint values are got from the interface Move Joints for updating the *DHParams* array of the model. Afterwards, the homogeneous transformations of each link are computed using the method *FKinematics* of the Robotics module of EjsRL. Finally, these matrix objects (*A01…A06*) are inserted in the property *Transform* (figure 5) of the VRML components in order to move them according this kinematics algorithm.



```
public void forwardKinematics () {

//Update the current values of the joints
DHParams[0] = q[0] + Math.PI/2;
DHParams[1] = q[1]-Math.PI/2;
DHParams[2] = q[2]+Math.PI;
DHParams[3] = q[3];
DHParams[4] = q[4];
DHParams[5] = q[5];

 //Compute the forward kinematics
A01 = robot.FKinematics(DHParams[0],1);
A02 = robot.FKinematics(DHParams[1],2);
A03 = robot.FKinematics(DHParams[2],3);
A04 = robot.FKinematics(DHParams[3],4);
A05 = robot.FKinematics(DHParams[4],5);
A06 = robot.FKinematics(DHParams[5],6);
}
```

Figure 6: Java code for the forward kinematics of a 6 rotational DOF robot.

EjsRL contains some methods to solve the inverse kinematics problem. Figure 7 shows an example for solving this based on the Jacobian operator. The method *IKinematics* receives the position and orientation of the end effector (Matrix *T*) and the current joint values of the robot (array *q_current*) as input parameters. Finally, the robot is moved to the suitable position using the forward kinematics method described before.

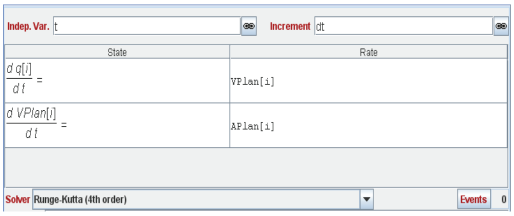


```
public void inverseKinematics () {

  //Current values of vector q
  double[] q_current = {q[0],q[1],q[2],q[3],q[4], q[5]};

 //Position and orientation of the end effector (X, Y, Z, Xº, Yº, Zº)
 Matrix T = new Matrix(4,4);
 T.set(0,3,X); T.set(1,3,Y); T.set(2,3,Z);T.set(3,3,1.0); //Position
 T.setMatrix(0,2,0,2,Maths.transRPYtoR(Roll, Pitch, Yaw)); //Orientation

 //Call to the inverse kinematics algorithm
 Solution sol = robot.IKinematics(T,q_current);
 if(sol!=null){
      q[0] = sol.getElemSolution(0); q[1] = sol.getElemSolution(1);
      q[2] = sol.getElemSolution(2); q[3] = sol.getElemSolution(3);
      q[4] = sol.getElemSolution(4); q[5] = sol.getElemSolution(5);

      //Move the robot with the updated q values
      forwardKinematics();
  }
}
```

Figure 7: Java code for the inverse kinematics problem.

With regard to trajectory planning, EJS+EjsRL allows users to easily perform the simulation of many path planning algorithms for n-axis robot arms. The ODEs editor implemented in EJS is employed to generate the position, velocity and acceleration values. The Robotics classes of EjsRL contain a path planning module which computes the acceleration parameters of several trajectories from their imposed constrains. Thus, two steps are only necessary to create a planning algorithm for a n-axis robot manipulator:

- To write the equations of the basic motion of a multi-body system. Figure 8 shows these equations in the ODEs editor of EJS. These equations compute the sequence values of the position (*q*) and velocity (*VPlan*) of all the robot joints from the acceleration of the trajectory (*APlan*);
- To compute the acceleration of the path planning algorithm proposed using one of the functions provided by the Robotics package. The trajectory planning module returns the acceleration parameters of several kinds of trajectories which can be used in the motion equations;



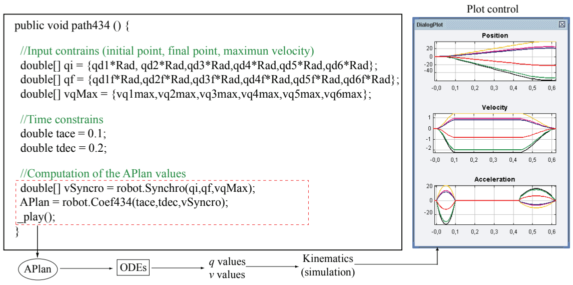
Indep. Var. t   Increment dt

State	Rate
$\frac{d\,q[i]}{dt}$ =	VPlan[i]
$\frac{d\,VPlan[i]}{dt}$ =	APlan[i]

Solver Runge-Kutta (4th order)   Events 0

Figure 8: ODEs of basic robot motion.

There are a lot of methods implemented in the Robotics module: splines, cubic interpolators, synchronous, asynchronous and linear trajectories, and the 4-3-4 polynomial path planning algorithm. Figure 9 shows the Java code to program this last interpolator in order to determinate the acceleration array for the differential equations.



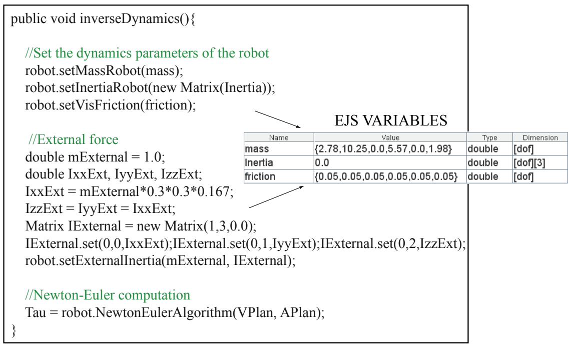
```
public void path434 () {

  //Input contrains (initial point, final point, maximun velocity)
  double[] qi = {qd1*Rad, qd2*Rad,qd3*Rad,qd4*Rad,qd5*Rad,qd6*Rad};
  double[] qf = {qd1f*Rad,qd2f*Rad,qd3f*Rad,qd4f*Rad,qd5f*Rad,qd6f*Rad};
  double[] vqMax = {vq1max,vq2max,vq3max,vq4max,vq5max,vq6max};

  //Time constrains
  double tace = 0.1;
  double tdec = 0.2;

  //Computation of the APlan values
  double[] vSyncro = robot.Synchro(qi,qf,vqMax);
  APlan = robot.Coef434(tace,tdec,vSyncro);
  _play();
}
```

Figure 9: Simulation of a 4-3-4 polynomial trajectory.

The generated joint values are automatically given to the kinematics model to simulate the robot movement (method *_play*). In addition, EJS plot controls can be used to visualize the trajectory variables (figure 9).

3.3 Dynamics Features

The Robotics module of EjsRL implements numerical methods to solve the forward and inverse dynamics problems (Newton-Euler and Walker-Orin, respectively). Figure 10 shows an example which obtains the inverse dynamics with an external force. Mass, inertias and friction properties must be known in order to solve this algorithm. The array variables *VPlan* and *APlan* belong to the velocity and acceleration of the path planning previously computed.



```
public void inverseDynamics(){

   //Set the dynamics parameters of the robot
   robot.setMassRobot(mass);
   robot.setInertiaRobot(new Matrix(Inertia));
   robot.setVisFriction(friction);

    //External force
   double mExternal = 1.0;
   double IxxExt, IyyExt, IzzExt;
   IxxExt = mExternal*0.3*0.3*0.167;
   IzzExt = IyyExt = IxxExt;
   Matrix IExternal = new Matrix(1,3,0.0);
   IExternal.set(0,0,IxxExt);IExternal.set(0,1,IyyExt);IExternal.set(0,2,IzzExt);
   robot.setExternalInertia(mExternal, IExternal);

   //Newton-Euler computation
   Tau = robot.NewtonEulerAlgorithm(VPlan, APlan);
}
```

EJS VARIABLES

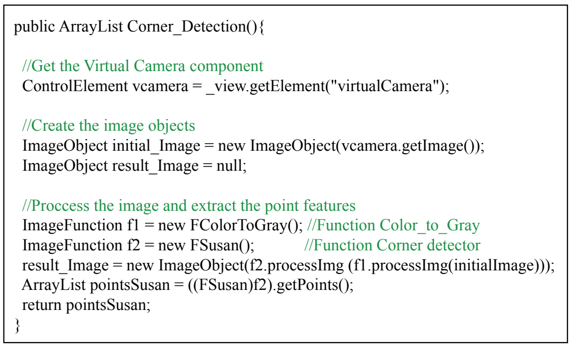
Name	Value	Type	Dimension
mass	{2.78,10.25,0.0,5.57,0.0,1.98}	double	[dof]
Inertia	0.0	double	[dof][3]
friction	{0.05,0.05,0.05,0.05,0.05,0.05}	double	[dof]

Figure 10: Programming the inverse dynamics with an external force.

3.4 Using Computer Vision Features

The Computer Vision classes of EjsRL provide a complete library for the development of image processing algorithms within EJS' environment. There are approximately fifty different functions implemented in this module, ranging from basic operations (format conversion, image adjustment, histogram, etc.) to image feature extraction (point and edge features). As example, authors implement a computer vision algorithm in the virtual robotic environment previously created. The aim is to perform an Eye-In-Hand (EIH) vision based control using four corner features in the control loop.

First of all, it is necessary to obtain a view projection from the end effector of the robot. For that end, EJS has an option which allows users to create a virtual camera in the 3D robotic environment. Figure 12 shows the appearance of the interface developed where the window "Virtual Camera" shows the projection of the EIH virtual camera. Secondly, this projection must be processed in order to extract the corner features of the object. Figure 11 shows the Java code which computes corner detection in the virtual camera's image (this code can also be used for real images). Initially, the image of the virtual camera control is obtained (variable *vcamera*) and the image objects are created. Afterwards, the processing algorithm is defined by means of the *ImageFunction* interface. Finally, the image is processed (*processImag* method), the point features are detected using one of the implemented algorithms, for example the SUSAN method (Smith and Brady, 1997), and these are returned as an array variable. These point features can be seen in the window "Virtual Image" of the figure 12.



```
public ArrayList Corner_Detection(){

  //Get the Virtual Camera component
  ControlElement vcamera = _view.getElement("virtualCamera");

  //Create the image objects
  ImageObject initial_Image = new ImageObject(vcamera.getImage());
  ImageObject result_Image = null;

  //Proccess the image and extract the point features
  ImageFunction f1 = new FColorToGray(); //Function Color_to_Gray
  ImageFunction f2 = new FSusan();           //Function Corner detector
  result_Image = new ImageObject(f2.processImg (f1.processImg(initialImage)));
  ArrayList pointsSusan = ((FSusan)f2).getPoints();
  return pointsSusan;
}
```

Figure 11: Java code for detecting corner features in the virtual image.

The control action and the interaction matrix used in this control algorithm are based on a

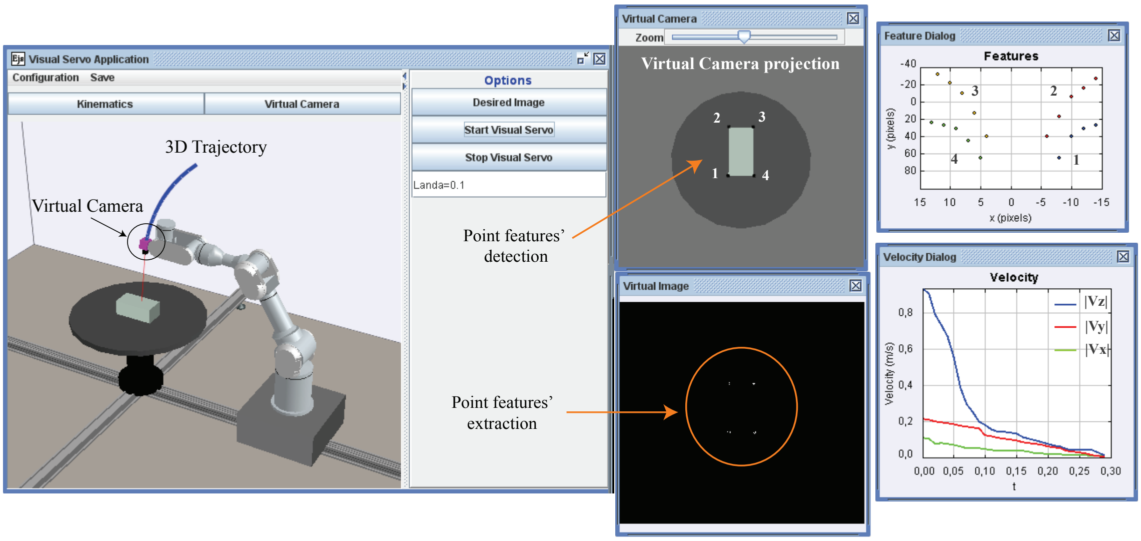


Figure 12: Simulation of a visual-servoing task using point features.

classical 2D visual servoing task, according to the following expressions:

$$\mathbf{v}_c = -\lambda \hat{\mathbf{L}}_s^{+} \left(\mathbf{s} - \mathbf{s}^{*} \right) \tag{1}$$

$$\hat{\mathbf{L}}_{si} = \begin{bmatrix} -1/Z_i & 0 & x_i/Z_i & x_i y_i & -(1+x_i^2) & y_i \\ 0 & -1/Z_i & y_i/Z_i & 1+y_i^2 & -x_i y_i & -x_i \end{bmatrix} \tag{2}$$

$$\hat{\mathbf{L}}_s^{T} = \left[\hat{\mathbf{L}}_{s1} \ \hat{\mathbf{L}}_{s2} \ \hat{\mathbf{L}}_{s3} \ \hat{\mathbf{L}}_{s4} \right] \tag{3}$$

where s are the current visual features, $s*$ are the desired visual features, and λ is the proportional controller; (xi, yi) are the point coordinates of each feature; and Zi is the current distance from the camera to the each feature. The evolution of both velocity module and point features are showed in figure 12, which validate the correct convergence of the visual servo task.

4 ADVANCED FEATURES

EJS has a connection with Matlab/Simulink which lets users specify and solve their models with the help of these tools (Sanchez et al., 2005). Next, authors show a decoupled control of a 3 rotational DOF robot where the electrical model is computed by a Simulink diagram and the 3D graphical interface is developed using EJS+EjsRL. Figure 13 shows the appearance of the application.

In the upper part of this figure, it can be seen the simulation of the robot with its respective plot controls, which show the input and the output values. Simulink diagram is set up by the PID control of each DOF, the power amplifier stage and the engine blocks with the model of a DC motor. The torque values are transferred to the forward dynamics method of EjsRL to compute the acceleration for the path planning algorithm. Feedback variables q and v are values obtained directly from the path planning and connected with the Simulink blocks.

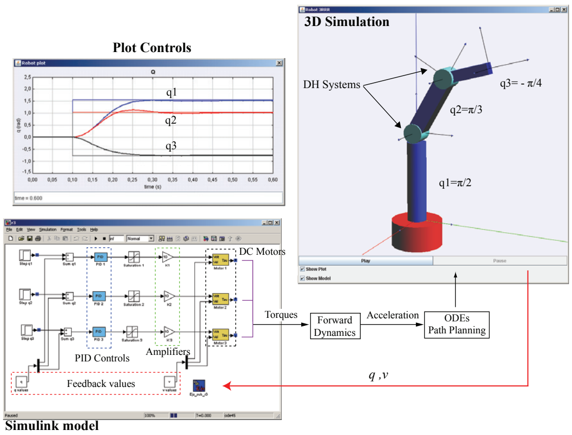


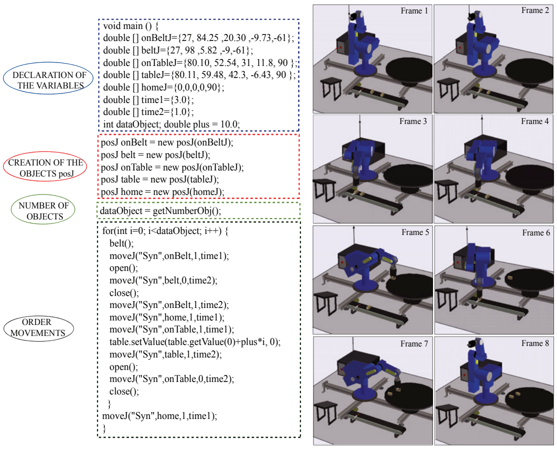
Figure 13: Position control of a robot using EJS+EjsRL and Simulink.

5 EXPERIMENTAL EXAMPLES

5.1 A Virtual and Remote Laboratory

Authors have developed with EJS+EjsRL a virtual and remote laboratory for training in Robotics. This system, called RobUALab.ejs (Jara et al., 2008), allows users to simulate path planning algorithms in a virtual robotic environment, as well as execute remote commands in a real robotic plant.

Programming classes of EjsRL (figure 1, package "Programming") enable users to develop Java routines in a robotic simulation. Figure 14 shows a programming experiment which consists of doing pick-and-place operations of virtual objects located in the conveyor belt using synchronous trajectories (parameter "Syn" in the method *moveJ*).



```
void main () {
double [] onBeltJ={27, 84.25 ,20.30 ,-9.73,-61};
double [] beltJ={27, 98 ,5.82 ,-9,-61};
double [] onTableJ={80.10, 52.54, 31, 11.8, 90 };
double [] tableJ={80.11, 59.48, 42.3, -6.43, 90 };
double [] homeJ={0,0,0,0,90};
double [] time1={3.0};
double [] time2={1.0};
int dataObject; double plus = 10.0;
posJ onBelt = new posJ(onBeltJ);
posJ belt = new posJ(beltJ);
posJ onTable = new posJ(onTableJ);
posJ table = new posJ(tableJ);
posJ home = new posJ(homeJ);
dataObject = getNumberObj();
for(int i=0; i<dataObject; i++) {
  belt();
  moveJ("Syn",onBelt,1,time1);
  open();
  moveJ("Syn",belt,0,time2);
  close();
  moveJ("Syn",onBelt,1,time2);
  moveJ("Syn",home,1,time1);
  moveJ("Syn",onTable,1,time1);
  table.setValue(table.getValue(0)+plus*i, 0);
  moveJ("Syn",table,1,time2);
  open();
  moveJ("Syn",onTable,0,time2);
  close();
 }
moveJ("Syn",home,1,time1);
}
```

Figure 14: States of the virtual robot during the execution of the programming experiment.

5.2 A Multi-robot System

EjsRL allows users the instantiation of different robot objects. Thus, it is possible to developed multi-robot simulations in an easy way. As example, a multi-robotic system composed by a PA-10 robot of 7 rotational DOF and a 3 rotational DOF robot (RRR) is presented here. This last serial robot is attached to a link of the upper part of the PA-10 (figure 15). In addition, the robot RRR has a virtual camera at the end as an EIH configuration. Figure 15 shows the interface of the application developed: on the left, the 3D virtual environment of the workspace; on the right, the virtual projection of the EIH camera located at RRR.

6 CONCLUSIONS

In this paper, a free Java-based software platform for the creation of advanced R&CV applications has been presented. EJS+EjsRL is a suitable tool to develop research and educational simulations in R&CV systems. The paper has showed several high-level applications which illustrate a part of the possibilities of EJS+EjsRL. More information can be obtained from http://www.aurova.ua.es/rcv, where readers can also execute a lot of test examples.

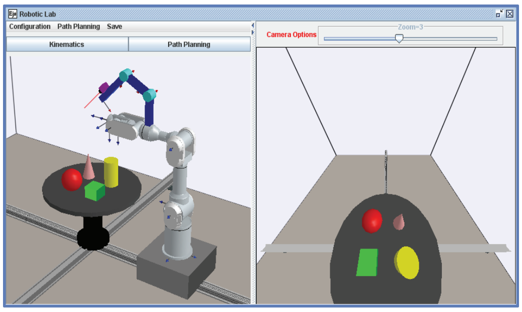


Figure 15: Appearance of the interface of a multi-robot simulation.

ACKNOWLEDGEMENTS

The work presented in this paper is supported by the Spanish Ministry of Education and Science (MEC) through the research project DPI2008-02647.

REFERENCES

Babuska, R. (2005). Design Environment for Robotic Manipulators. In *Proceedings of the 16th IFAC World Congress,* vol. 16, Prague.

Corke, P. (1996). A Robotics Toolbox for MATLAB. *IEEE Robotics and Automation Magazine*, 3(1): 24-32.

Easy-ROB Software (2004). http://www.easy-rob.com/.

Esquembre, F. (2004). Easy Java Simulations: A software tool to create scientific simulations in Java. *Computer Physics Communications,* 156(2): 199-204.

Falconi, R. and Melchiorri, C. (2008). Roboticad: An Educational Tool for Robotics. In *Proceedings of the 17th IFAC World Congress*, vol. 17, Seoul.

Gourdeau, R. (1997). Object-oriented programming for robotic manipulator simulation. *IEEE Robotics & Automation Magazine,* 4(1): 21-29.

Jackson, J. (2007). Microsoft Robotics studio: A technical introduction. *IEEE Robotics & Automation Magazine,* 14(1): 82-87.

Jara, C., Candelas, F. and Torres, F. (2008). An advanced interactive interface for Robotics e-learning. *International Journal of On-line Engineering*, 4(1): 17-25.

Jara, C., Candelas F., Torres, F., Esquembre, F., Dormido, S. and Reinoso, O. (2009). Real-time collaboration of virtual laboratories through the Internet. *Computers & Education*, 52(1): 126-140.

Jaramillo, A., Matta, A., Correa, F. and Perea, W. (2006). ROBOMOSP. *IEEE Robotics & Automation Magazine,* 13(1): 62-73.

Java Advanced Imaging library (JAI) (2004). http://java.sun.com/products/javamedia/jai.

Open Source Computer Vision Library (2001). http://www.intel.com/research/mrl/research/opencv.

Pelossof, R., Miller, A., Allen, P and Jebara, T. (2004). An SVM learning approach to robotic grasping. In *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 21, Taiwan.

Sanchez, J., Esquembre, F., Martín, C., Dormido, S., Dormido-Canto, S., Canto, R., Pastor, R. and Urquia, A. (2005). Easy Java Simulations: an open-source tool to develop interactive virtual laboratories using MATLAB/Simulink. *International Journal of Engineering Education*, 21(5): 789-813.

Smith, S. and Brady, J. (1997). SUSAN: a new approach to low level image processing. *International Journal of Computer Vision*, 23(1): 45-78.

VXL libraries (2001). http://vxl.sourceforge.net/.

FORMATION CONTROL OF MULTI-ROBOTS VIA SLIDING-MODE TECHNIQUE

Razvan Solea, Daniela Cernega, Adrian Filipescu and Adriana Serbencu
Control Systems and Industrial Informatics Department, Computer Science Faculty
“Dunarea de Jos” University of Galati, Domneasca 111, 800201, Galati, Romania
{*razvan.solea, daniela.cernega, adrian.filipescu, adriana.serbencu*}*@ugal.ro*

Keywords: Formation control, Sliding-mode control, Skid-steering mobile robot.

Abstract: This paper addresses the control of a team of nonholonomic mobile robots. Indeed, the most work, in this domain, have studied extensively classical control for keeping a formation of mobile robots. In this work, the leader mobile robot is controlled to follow an arbitrary reference path, and the follower mobile robot use the sliding-mode controller to keep constant relative distance and constant angle to the leader robot. The efficiency and simplicity of this control laws has been proved by simulation on different situations.

1 INTRODUCTION

The multi robots systems is an important robotics research field. Such systems are of interest for many reasons; tasks could be too complex for a simple robot to accomplish; using several simple robots can be easier, cheaper and more flexible than a single powerful robot (Mazo et al., 2004), (Zavlanos and Pappas, 2008), (Murray, 2007), (Liu et al., 2007), (Klančar et al., 2009).

Formation control has been one of the important research topics in multiple robot systems as it is applicable to many areas such as geographical exploration, rescue operations, surveillance, mine sweeping, and transportation. Different approaches have been developed recently, for example, behavior-based control, LQ control, visual servoing control, Lyapunov-based control, input and output feedback linearization control, graph theory, and nonlinear control.

In leader-follower formation control, the most widely used control technique is feedback linearization based on the kinematics model of the system. In this study, we focus on the problem of leader-follower robot formation control using a sliding-mode controller.

The referenced robot is called a leader, and the robot following it called a follower. Thus, there are many pairs of leaders and followers and complex formations can be achieved by controlling relative positions of these pairs of robots respectively. This approach is characterized by simplicity, reliability and no need for global knowledge and computation.

Figure 1: A skid-steered four wheel mobile robot.

In this paper, it will be developped a method based on the leader-following approach to investigate formation control problem in a group of nonholonomic mobile robots. For this purpose, we design a new controller based on sliding-mode control to drive a fleet of mobile robots in a leader-follower configuration.

Sliding Mode Control (SMC) method has been widely noticed because of its superior robust control performance for systems with highly uncertainty (Chwa, 2004), (Yang and Kim, 1999), (Floquet et al., 2003), (Solea and Nunes, 2007), (Solea and Cernega, 2009).

The rest of this paper is organized as follows. First formulation of the nonholonomic mobile robot system is revealed. Then, the leader-following formation model method used by the robots is exposed. After that, the architecture of the siling-mode controller is described. This paper concludes with some simulation and results.

2 FORMATION CONTROL

2.1 Formulation of the Nonholonomic Mobile Robot System

As indicated in Figure 1, the mobile robots are skid-steering mobile platforms. To develop the kinematic model for a skid-steering mobile robot (SSMR) that is assumed to move in a plane (for simplicity) with an inertial coordinate system, denoted by $(X_g,\ Y_g)$, and a local coordinate system, denoted by $(X_r,\ Y_r)$, where the origin of $(X_r,\ Y_r)$ is fixed to the center of mass (CG) of the SSMR as illustrated in Figure 2. The position and orientation of the CG, denoted by $q(t) \in R^3$, is defined as $q = [x_r\ y_r\ \theta_r]^T$ (i.e., the CG position, x_r and y_r, and the orientation θ_r of the local coordinate frame with respect to the inertial frame).

$$\begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\theta}_r \end{bmatrix} = \begin{bmatrix} cos(\theta_r) & -sin(\theta_r) & 0 \\ sin(\theta_r) & cos(\theta_r) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v_{xr} \\ v_{yr} \\ \omega_r \end{bmatrix} \tag{1}$$

It is obvious that Eqn. (1) does not impose any restrictions on the skid-steering mobile robot plane movement, since it describes free-body kinematics only. Therefore it is necessary to analyze the relationship between wheel velocities and local velocities.

Suppose that the $i-th$ wheel rotates with an angular velocity $\omega_{ir}(t)$, where $i = 1, 2, ..., 4$, which can be seen as a control input. For simplicity, the thickness of the wheel is neglected and is assumed to be in contact with the plane at point P_{ir} as illustrated in Figure 3. In contrast to most wheeled vehicles, the lateral velocity of the SMRR, v_{iyr}, is generally nonzero. This property comes from the mechanical structure of the SSMR that makes lateral skidding necessary if the vehicle changes its orientation. Therefore the wheels are tangent to the path only if $\omega_r = 0$, i.e., when the robot moves along a straight line.

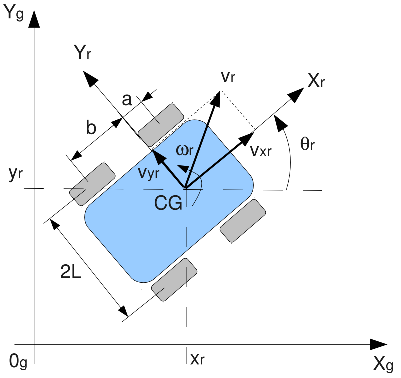


Figure 2: Free body diagram.

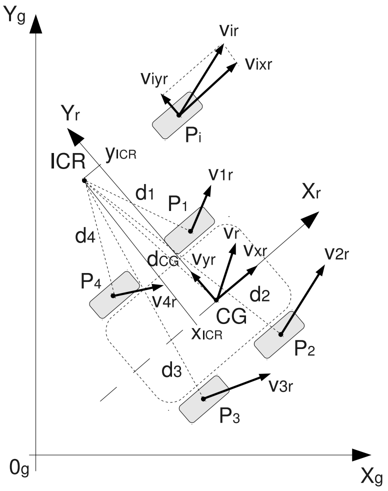


Figure 3: Wheel velocities.

In this description we consider only a simplified case of the SSMR movement for which the longitudinal slip between the wheels and the surface can be neglected. For traditional mobile robots, the wheel rotation is translated into a linear motion along the tangent of a curve without longitudinal slippage as described by the following expressions:

$$v_{ixr} = R_i \cdot \omega_i \tag{2}$$

where v_{ixr} is the longitudinal component of the total velocity vector v_{ir} of the $i-th$ wheel expressed in the local frame and R_i denotes the so-called effective rolling radius of that wheel.

The vectors $d_i(t) = [d_{ix}(t), d_{iy}(t)]^T$ and $d_{CG}(t) = [d_{CGx}(t), d_{CGy}(t)]^T \in R^2$ are expressed in (X_r, Y_r) and are defined from the instantaneous center of rotation (ICR) of the vehicle to P_i $\forall i = 1, 2, ..., 4$ and $d_{CG}(t)$ from the ICR to the vehicle to CG, respectively, as illustrated in Figure 3. Based on the geometry of Figure 3, the following expressions can be developed:

$$\frac{v_{ixr}}{d_{iy}} = \frac{v_{xr}}{y_{ICR}} = \frac{v_{iyr}}{d_{ix}} = -\frac{v_{yr}}{x_{ICR}} = \omega_r \tag{3}$$

where it is used the fact that the coordinates of the ICR expressed in (X_r, Y_r), denoted by $x_{ICR}(t)$ and $y_{ICR}(t) \in R$, are defined as $[x_{ICR}, y_{ICR}]^T = [-d_{CGx}, d_{CGy}]^T$.

From Figure 3 it is clear that the coordinates of vectors d_{ir} satisfy the following relationships:

$$\begin{aligned} d_{1y} = d_{4y} &= d_{CGy} - L \\ d_{2y} = d_{3y} &= d_{CGy} + L \\ d_{1x} = d_{2x} &= d_{CGx} + a \\ d_{3x} = d_{4x} &= d_{CGx} - b \end{aligned} \tag{4}$$

After combining Eqns. (3) and (4), the following relationships between wheel velocities can be obtained:

$$\begin{aligned} v_L = v_{1xr} = v_{4xr} \quad & v_R = v_{2xr} = v_{3xr} \\ v_F = v_{1yr} = v_{2yr} \quad & v_B = v_{3yr} = v_{4yr} \end{aligned} \tag{5}$$

where v_L and v_R denote the longitudinal coordinates of the left and right wheel velocities, v_F and v_B are the lateral coordinates of the velocities of the front and rear wheels, respectively.

Using (3) - (5) it is possible to obtain the following transformation describing the relationship between the wheel velocities and the velocity of the robot:

$$\begin{bmatrix} v_L \\ v_R \\ v_F \\ v_B \end{bmatrix} = \begin{bmatrix} 1 & -L \\ 1 & L \\ 0 & -x_{ICR}+a \\ 0 & -x_{ICR}-b \end{bmatrix} \cdot \begin{bmatrix} v_{xr} \\ \omega_r \end{bmatrix} \qquad (6)$$

Notice that ω_L and ω_R which denote angular velocities of left and right wheels, respectively, can be regarded as control inputs at kinematic level and can be used to control longitudinal and angular velocity according to the following relationships:

$$v_{xr} = R \cdot \frac{\omega_R + \omega_L}{2}, \; \omega_r = R \cdot \frac{\omega_R - \omega_L}{2 \cdot L} \qquad (7)$$

while R is so called effective radius of wheels and $2 \cdot L$ is a spacing wheel track depicted in Figure 2.

It is interesting to see that the analysed kinematic model of the SSMR is quite similar to the kinematics of the two-wheel mobile robot.

From the last equations it is clear that, theoretically, the pair of velocities ω_L and ω_R can be treated as a control kinematic input signal as well as velocities v_{xr} and ω_r. However, the accuracy of the relations (7) mostly depends on the longitudinal slip and can be valid only if this phenomenon is not dominant. In addition, the parameters R and L may be identified experimentally to ensure a high validity of the determination of the angular robot velocity with respect to the angular velocities of the wheels.

To complete the kinematic model of the SSMR, the following velocity constraint can be considered:

$$v_{yr} + x_{ICR} \cdot \omega_r = 0 \qquad (8)$$

The last equation is not integrable. In consequence, it describes a nonholonomic constraint which can be rewritten like:

$$\begin{bmatrix} -sin(\theta_r) & cos(\theta_r) & x_{ICR} \end{bmatrix} \cdot \begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\theta}_r \end{bmatrix} = 0 \qquad (9)$$

2.2 Leader-follower Formation Models

Figure 4 is a leader-follower control model where the formation pattern is specified by the separate distance d and the relative bearing ψ for two robots $r1$ and $r2$. The desired formation pattern can be defined as the desired separate distance d^d and the relative bearing ψ^d. The follower $r2$ regulates the formation state errors of the separate distance and the relative bearing through its speed control signals $u_{r2} = [v_{xr2}, \omega_{r2}]^T$:

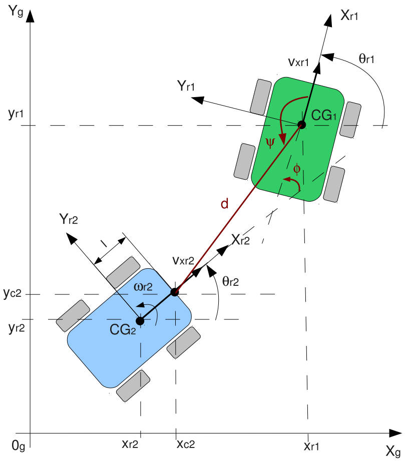


Figure 4: Leader-follower approach for SSMR.

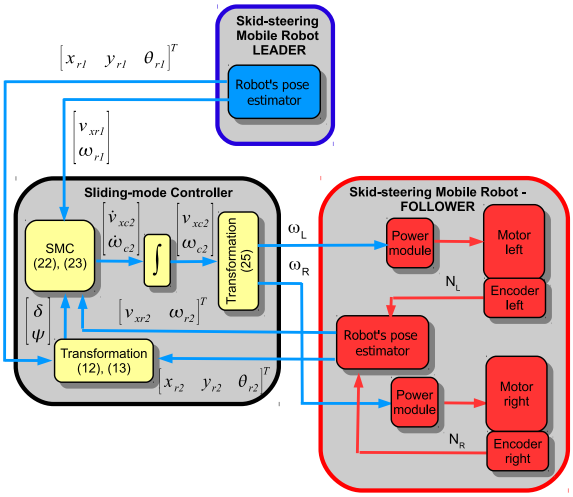

Skid-steering Mobile Robot LEADER
Robot's pose estimator
$[x_{r1} \; y_{r1} \; \theta_{r1}]^T$
$[v_{xr1} \; \omega_{r1}]$
Sliding-mode Controller
SMC (22), (23)
$[\dot{v}_{xc2} \; \dot{\omega}_{c2}]$
∫
$[v_{xc2} \; \omega_{c2}]$
Transformation (25)
$[\delta \; \psi]$
$[v_{xr2} \; \omega_{r2}]^T$
Transformation (12), (13)
$[x_{r2} \; y_{r2} \; \theta_{r2}]^T$
ω_L
ω_R
Skid-steering Mobile Robot - FOLLOWER
Power module
Motor left
N_L
Encoder left
Robot's pose estimator
Power module
Motor right
N_R
Encoder right

Figure 5: Block diagram.

$$\begin{bmatrix} \tilde{d} \\ \tilde{\psi} \end{bmatrix} = \begin{bmatrix} d^d \\ \psi^d \end{bmatrix} - \begin{bmatrix} d \\ \psi \end{bmatrix} \qquad (10)$$

The relative distance between the leader and the follower robot is denoted as d, the separation bearing angle is ψ, and they are given by:

$$d = \sqrt{(x_{r1} - x_{c2})^2 + (y_{r1} - y_{c2})^2} \qquad (11)$$

$$\psi = \pi - [\theta_{r1} - arctan2(y_{r1} - y_{c2}, x_{r1} - x_{c2})] \qquad (12)$$

where:

$$x_{c2} = x_{r2} + l \cdot cos(\theta_{r2}), \; y_{c2} = y_{r2} + l \cdot sin(\theta_{r2})$$

The formation control can be investigated by modeling the formation state error as follows (Das et al., 2002):

$$\begin{bmatrix} \dot{\tilde{d}} \\ \dot{\tilde{\psi}} \end{bmatrix} = G \cdot u_{r2} + F \cdot u_{r1},\ \dot{\phi} = \omega_{r1} - \omega_{r2} \qquad (13)$$

and

$$G = \begin{bmatrix} -cos(\phi+\psi) & -l \cdot sin(\phi+\psi) \\ \dfrac{sin(\phi+\psi)}{d} & -\dfrac{l \cdot cos(\phi+\psi)}{d} \end{bmatrix},$$

$$F = \begin{bmatrix} cos(\psi) & 0 \\ -\dfrac{sin(\psi)}{d} & 1 \end{bmatrix}$$

where $\phi = \theta_{r1} - \theta_{r2}$ and l is the distance between the robot position (x_{r2}, y_{r2}) and the robot hand position (x_{c2}, y_{c2}) as shown in Figure 4.

2.3 Sliding-mode Controller Design

In a leader-follower configuration, with the leader's position given and once the follower's relative distance and angle with respect to the leader are known, the follower's position can be determined. To use the leader-following approach, it is assumed that the angular and linear velocities of the leader are known. In order to achieve and maintain the desired formation between the leader and follower, it is only need to control the follower's angular and linear velocities to achieve the relative distance and angle between them as specified. Therefore, the leader-following based mobile robot formation control can be considered as an extension of the tracking control problem of the nonholonomic mobile robot.

A practical form of reaching the control law (proposed by Gao and Hung (Gao and Hung, 1993)) is defined as

$$\dot{s}_i = -p_i \cdot |s_i|^{\alpha} \cdot sgn(s_i),\ \ 0 < \alpha < 1,\ i = 1,2 \qquad (14)$$

This reaching law increases the reaching speed when the state is far away from the switching manifold, but reduces the rate when the state is near the manifold. The result is a fast reaching and low chattering reaching mode.

A new design of sliding surface is proposed, such that the separation bearing angle, ψ and the orientation error ϕ, are internally coupled with each other in a sliding surface leading to convergence of both variables. For that purpose the following sliding surfaces is proposed:

$$s_1 = \dot{\tilde{d}} + \gamma_d \cdot \tilde{d} \qquad (15)$$

$$s_2 = \dot{\tilde{\psi}} + \gamma_{\psi} \cdot \tilde{\psi} + \gamma_0 \cdot sgn(\tilde{\psi}) \cdot |\phi| \qquad (16)$$

here γ_0, γ_d, γ_{ψ} are positive constant parameters and $\tilde{d}$, $\tilde{\psi}$, ϕ are defined by (10), (13).

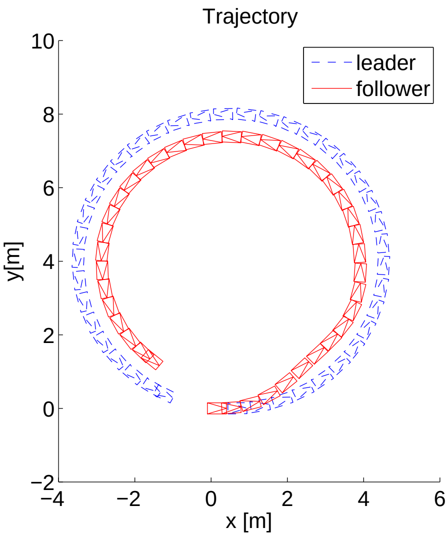


Figure 6: Simulation I - Trajectory of the leader and the follower.

If s_1 converges to zero, trivially $\tilde{d}$ converges to zero. If s_2 converges to zero, in steady-state it becomes $\dot{\tilde{\psi}} = -\gamma_{\psi} \cdot \tilde{\psi} - \gamma_0 \cdot sgn(\tilde{\psi}) \cdot |\phi|$. Since $|\phi|$ is always bounded, the following relationship between $\tilde{\psi}$ and $\dot{\tilde{\psi}}$ holds: $\tilde{\psi} < 0 \Rightarrow \dot{\tilde{\psi}} > 0$ and $\tilde{\psi} > 0 \Rightarrow \dot{\tilde{\psi}} < 0$.

From the time derivative of (15) and (16) and using the reaching law defined in (14) yields:

$$\dot{s}_1 = \ddot{\tilde{d}} + \gamma_d \cdot \dot{\tilde{d}} = -p_1 \cdot |s_1|^{\alpha} \cdot sgn(s_1) \qquad (17)$$

$$\begin{aligned} \dot{s}_2 = \ddot{\tilde{\psi}} + \gamma_{\psi} \cdot \dot{\tilde{\psi}} + \gamma_0 \cdot sgn(\tilde{\psi}) \cdot sgn(\tilde{\phi}) \cdot \dot{\phi} = \\ = -p_2 \cdot |s_2|^{\alpha} \cdot sgn(s_2) \end{aligned} \qquad (18)$$

After some mathematical manipulation, one can achieve:

$$\dot{v}_{xc2} = \frac{p_1 \cdot |s_1|^{\alpha} \cdot sgn(s_1) + \gamma_d \cdot \dot{\tilde{d}} - D_1}{cos(\phi+\psi)} \qquad (19)$$

$$\dot{\omega}_{c2} = \frac{(p_2 \cdot |s_2|^{\alpha} \cdot sgn(s_2) + \gamma_{psi} \cdot \dot{\tilde{\psi}}) \cdot d - D_2}{l \cdot cos(\phi+\psi)} \qquad (20)$$

where

$$D_1 = l \cdot \dot{\omega}_{r2} \cdot sin(\phi+\psi) - d \cdot (\dot{\phi} + \dot{\psi}) \cdot (\dot{\psi} + \omega_{r1}) -$$
$$-\dot{v}_{xr1} \cdot cos(\psi) - v_{xr1} \cdot \dot{\phi} \cdot sin(\psi)$$
$$D_2 = \gamma_0 \cdot sgn(\tilde{\psi} \cdot \phi) \cdot \dot{\phi} \cdot d - \dot{v}_{xr2} \cdot sin(\phi+\psi) +$$
$$+\dot{v}_{rx1} \cdot sin(\psi) - v_{xr1} \cdot \dot{\phi} \cdot cos(\psi) - (\dot{\phi} + \dot{\psi}) \cdot \dot{d} -$$
$$-d \cdot \dot{\omega}_{r1} - \dot{d} \cdot (\dot{\psi} + \omega_{r1})$$

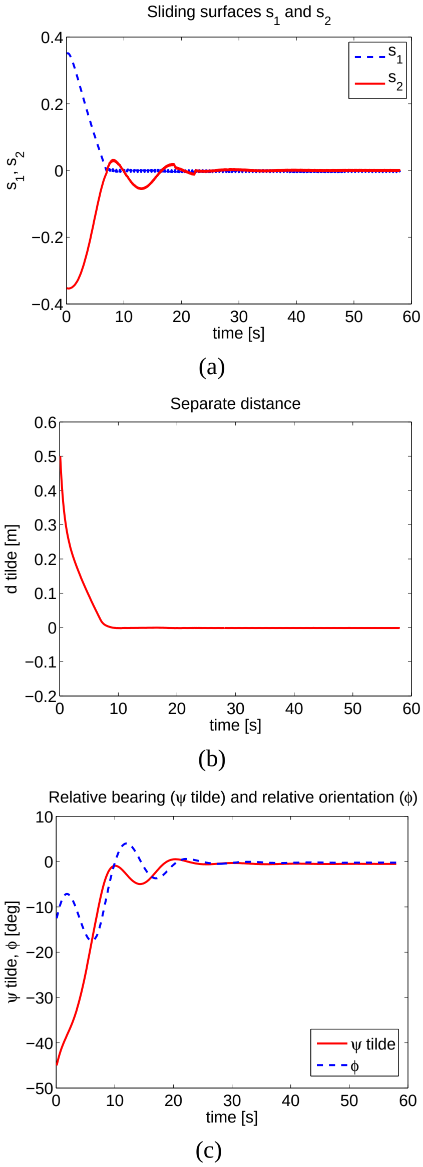
The *signum* functions in the sliding surface were replaced by *saturation* functions, to reduce the chattering phenomenon (Slotine and Li, 1991).

3 SIMULATION RESULTS

In this section, simulation results for the proposed SMC are presented. The simulation are performed in Matlab/Simulink environment to verify behavior of the controlled system. The parameters of the SSMR model were chosen to correspond as closely as possible to the real experimental robot presented in section 1 in the following manner: $a = 0.10[m]$, $b = 0.20[m]$, $L = 0.12[m]$, $R = 0.04[m]$. Wheel velocity commands,

$$\omega_R = \frac{v_{xc2} + L \cdot \omega_{c2}}{R}; \quad \omega_L = \frac{v_{xc2} - L \cdot \omega_{c2}}{R}; \qquad (21)$$

are sent to the power modules of the follower mobile robot, and encoder measures NR and NL are received in the robots pose estimator for odometric computations.



(a)

(b)

(c)

Figure 7: Simulation I - Sliding surfaces s_1 and s_2, separate distance ($\tilde{d}$), relative bearing ($\tilde{\psi}$) and relative orientation (ϕ).

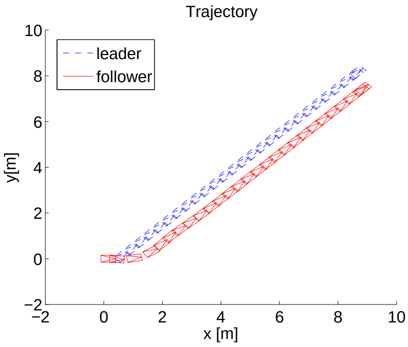


Figure 8: Simulation II - Trajectory of the leader and the follower.

Figure 5 shows a block diagram of the proposed sliding-mode controller.

In order to compute the actuating control input, equation (6) needs to be integrated and some initial values $v_{xc2}(0)$, $\omega_{c2}(0)$ to be fixed.

Two simulation experiments were carried out to evaluate the performance of the sliding mode controller presented in Section 2.3. The first simulation refers to the case of circular trajectory ($v_{xr1} = 0.4[m/s]$ and $\omega_{r1} = 0.1[rad/s]$). The initial conditions of the leader and the follower are, $x_{r1}(0) = 0.5$, $y_{r1}(0) = 0$, $\theta_{r1}(0) = 0$, $x_{r2}(0) = 0$, $y_{r2}(0) = 0$, $\theta_{r1}(0) = 0$, $d^d = 1.0[m]$, $\psi^d = 135[deg]$.

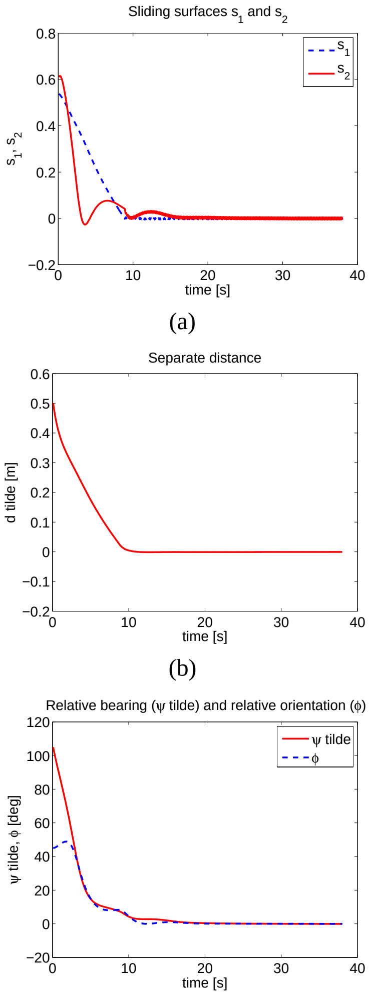
In the second simulation the leader robot execute a linear trajectory but with a non-zero initial orietation ($\theta_{r1} = 45[deg]$). The initial conditions of the leader and the follower in this second case are, $x_{r1}(0) = 0.5$, $y_{r1}(0) = 0$, $\theta_{r1}(0) = pi/4$, $x_{r2}(0) = 0$, $y_{r2}(0) = 0$, $\theta_{r1}(0) = 0$, $d^d = 1.0[m]$, $\psi^d = -120[deg]$.

Figure 6 shows the trajectory of the leader and the follower for the first simulation case. In order to have a temporal reference in the figure the robots are drawn each second: the blue car represent the leader and the red car represent the follower.

In Figure 7.a the sliding surfaces s_1 and s_2 asymptotically converge to zero. Finally, Figures 7.b and c show the time histories of $\tilde{d}$, $\tilde{\psi}$ and ϕ.

The results of the second case called Simulation II are given in Figures 8 - 9. Figure 8 shows the trajectory of the leader and the follower, Figures 9 the sliding surfaces and the time histories of $\tilde{d}$, $\tilde{\psi}$ and ϕ.

The good performance for controlling the formation with the developed control law can be observed from Figures 6 - 9. The outputs of the formation system ($\tilde{d}$, $\tilde{\psi}$ and ϕ) asymptotically converge to zero, as shown in Figures 7 and 9.



(a)

(b)

(c)

Figure 9: Simulation II - Sliding surfaces s_1 and s_2, separate distance $(\tilde{d})$, relative bearing $(\tilde{\psi})$ and relative orientation (ϕ).

4 CONCLUSIONS

In this paper is proposed a sliding mode formation tracking control scheme of nonholonomic mobile robots. The leader and follower are a skid-steering mobile robots. The desired formation, defined by two parameters (a distance and an orientation function) is allowed to vary in time. The effectiveness of the proposed designs has been validated via simulation experiments.

Future research lines include the experimental validation of our control scheme and the extension of our results to skid-steering mobile robots. For the sake of simplicity in the present paper a single-leader, single-follower formation has been considered. Future investigations will cover the more general case of multi-leader, multi-follower formations.

ACKNOWLEDGEMENTS

This work was supported by CNCSIS-UEFISCSU, projects PNII-IDEI 506/2008 and PNII-IDEI 641/2007.

REFERENCES

Chwa, D. (2004). Sliding-mode tracking control of nonholonomic wheeled mobile robots in polar coordinates. *IEEE Transactions on Control*, 12(4):637–644.

Das, A., Fierro, R., Kumar, V., Ostrowski, J., Spletzer, J., and Taylor, C. (2002). A vision-based formation control framework. *IEEE Transactions on Robotics and Automation*, 18(5):813–825.

Floquet, T., Barbot, J. P., and Perruquetti, W. (2003). Higher-order sliding mode stabilization for a class of nonholonomic perturbed systems. *Automatica*, 39(6):1077–1083.

Gao, W. and Hung, J. C. (1993). Variable structure control of nonlinear systems: A new approach. *IEEE Transactions on Industrial Electronics*, 40(1):45–55.

Klančar, G., Matko, D., and Blažič, S. (2009). Wheeled mobile robots control in a linear platoon. *Journal of Intelligent and Robotic Systems*, 54(5):709–731.

Liu, S.-C., Tan, D.-L., and Liu, G.-J. (2007). Robust leader-follower formation control of mobile robots based on a second order kinematics model. *Acta Automatica Sinica*, 33(9):947–955.

Mazo, M., Speranzon, A., Johansson, K., and Hu, X. (2004). Multi-robot tracking of a moving object using directional sensors. In *IEEE International Conference on Robotics and Automation, ICRA '04*, volume 2, pages 1103–1108.

Murray, R. M. (2007). Recent research in cooperative control of multi-vehicle systems. *Journal of Dynamic Systems, Measurement, and Control*, 129(5):571–583.

Slotine, J. J. E. and Li, W. (1991). *Applied Nonlinear Control*. Prentice-Hall.

Solea, R. and Cernega, D. (2009). Sliding mode control for trajectory tracking problem - performance evaluation. In *Lecture Notes in Computer Science, Springer*, volume 5769, pages 865–874.

Solea, R. and Nunes, U. (2007). Trajectory planning and sliding-mode control based trajectory-tracking for cybercars. *Aided Engineering, IOS Press*, 14(1):33–47.

Yang, J.-M. and Kim, J.-H. (1999). Sliding mode control for trajectory tracking of nonholonomic wheeled mobile robots. *IEEE Transactions on Robotics and Automation*, 15(3):578–587.

Zavlanos, M. M. and Pappas, G. J. (2008). Dynamic assignment in distributed motion planning with local coordination. *IEEE Transactions on Robotics*, 24(1):232–242.

KINEMATIC IDENTIFICATION OF PARALLEL MECHANISMS BY A DIVIDE AND CONQUER STRATEGY

Sebastián Durango[a], David Restrepo[a], Oscar Ruiz[a], John Restrepo-Giraldo[b] and Sofiane Achiche[b]
[a]*CAD CAM CAE research laboratory, EAFIT University, Medellín, Colombia*
[b]*Management Engineering Dept., Technical University of Denmark, Lyngby, Denmark*
{*sdurang1, drestr21, oruiz*}*@eafit.edu.co*, {*jdrg, soac*}*@man.dtu.dk*

Keywords: Parallel mechanisms, Kinematic identification, Robot calibration.

Abstract: This paper presents a Divide and Conquer strategy to estimate the kinematic parameters of parallel symmetrical mechanisms. The Divide and Conquer kinematic identification is designed and performed independently for each leg of the mechanism. The estimation of the kinematic parameters is performed using the inverse calibration method. The identification poses are selected optimizing the observability of the kinematic parameters from a Jacobian identification matrix. With respect to traditional identification methods the main advantages of the proposed Divide and Conquer kinematic identification strategy are: (i) reduction of the kinematic identification computational costs, (ii) improvement of the numerical efficiency of the kinematic identification algorithm and, (iii) improvement of the kinematic identification results. The contributions of the paper are: (i) The formalization of the inverse calibration method as the Divide and Conquer strategy for the kinematic identification of parallel symmetrical mechanisms and, (ii) a new kinematic identification protocol based on the Divide and Conquer strategy. As an application of the proposed kinematic identification protocol the identification of a planar 5R symmetrical mechanism is simulated. The performance of the calibrated mechanism is evaluated by updating the kinematic model with the estimated parameters and developing simulations.

1 INTRODUCTION

In mechanisms and manipulators the accuracy of the end-effector critically depends on the knowledge of the kinematic model governing the control model (Zhuang et al., 1998). Therefore, to improve the accuracy of a mechanism its kinematic parameters have to be precisely estimated by means of a kinematic identification procedure (Renaud et al., 2006).

Kinematic identification is an instance of the robot calibration problem. The estimation of rigid-body inertial parameters and the estimation of sensor gain and offset are instances of calibration problems at the same hierarchical level of the kinematic calibration problem (Hollerbach et al., 2008).

This paper is devoted to the kinematic identification of parallel symmetrical mechanisms. Parallel mechanisms are instances of closed-loop mechanisms typically formed by a moving platform connected to a fixed base by several legs. Each leg is a kinematic chain formed by a pattern of links, actuated and passive joints relating the moving platform with the fixed base. If the pattern of joints and links is the same for each leg and each leg is controlled by one actuator, then the parallel mechanism is denoted symmetrical (Tsai, 1999).

For parallel mechanisms the kinematic identification is usually performed minimizing an error between the measured joint variables and their corresponding values calculated from the measured end-effector pose through an inverse kinematic model (Zhuang et al., 1998; Renaud et al., 2006). This method is preferred for the identification of parallel mechanisms because:
1. Inverse kinematics of parallel mechanisms is usually derived analytically avoiding the numerical problems associated with any forward kinematics solution (Zhuang et al., 1998; Renaud et al., 2006).
2. The inverse calibration method is considered to be the most numerically efficient among the identification algorithms for parallel mechanisms (Renaud et al., 2006; Besnard and Khalil, 2001).
3. With respect to forward kinematic identification no scaling is necessary to balance the contribution of position and orientation measurements (Zhuang et al., 1998).

In the case of parallel symmetrical mechanisms the inverse kinematic modeling can be formulated using independent loop-closure equations. Each loop-closure equation relates the end-effector pose, the geometry of a leg, and a fixed reference frame. In consequence, an independent kinematic constraint equation is formulated for each leg forming the mechanism. For the case of parallel symmetrical mechanisms the set of constraint equations is equal to the number of legs and to the number of degrees of freedom of the mechanisms. Each kinematic constraint equation can be used for the independent identification of the parameters of the leg correspondent to the equation.

The independent identification of the kinematic parameters of each leg in parallel mechanisms allows to improve:
1. The numerical efficiency of the identification algorithm (Zhuang et al., 1998).
2. The kinematic calibration performance by the design of independent experiments optimized for the identification of each leg.
The independent identification of leg parameters in parallel mechanisms was sketched in (Zhuang et al., 1998) and developed for the specific case of Gough platforms in (Daney et al., 2002; Daney et al., 2005). However, the idea of the independence in the kinematic identification of each leg in a parallel mechanism is not completely formalized.

This article presents a contribution to the improvement of the pose accuracy in parallel symmetrical mechanisms by a kinematic calibration protocol based on inverse kinematic modeling and a divide and conquer strategy. The proposed divide and conquer strategy takes advantage of the independent kinematic identification of each leg in a parallel mechanism not only from a numerical stand point but also from the selection of the optimal measurement set of poses that improves the kinematic identification of the parameters of the leg itself.

The layout for the rest of the document is as follows: section 2 develops a literature review on the inverse calibration of parallel mechanisms method, section 3 presents the divide and conquer identification of parallel mechanisms strategy, section 4 develops a kinematic identification of parallel mechanisms protocol, section 5 presents the simulated kinematic identification of a planar 5R symmetrical mechanism using the identification protocol, finally, in section 6 the conclusions are developed.

2 LITERATURE REVIEW

The modeling of mechanical systems include the design, analysis and control of mechanical devices. An accurate identification of the model parameters is required in the case of control tasks (Hollerbach et al., 2008). Instances of models of mechanical systems includes kinematic, dynamic, sensor, actuators and flexibility models. For parallel mechanisms updating the kinematic models with accurately estimated parameters is essential to achieve precise motion at high-speed rates. This is the case when parallel mechanisms are used in machining applications (Renaud et al., 2006).

The inverse calibration method is accepted as the natural (Renaud et al., 2006; Zhuang et al., 1998) and most numerically efficient (Besnard and Khalil, 2001) among the identification algorithms for parallel mechanisms. The inverse calibration method is based on inverse kinematic modeling and a external metrological system. The calibration is developed minimizing an error residual between the measured joint variables and its estimated values from the end-effector pose though the inverse kinematic model. The derivation of the inverse kinematic model of parallel mechanisms is usually straightforward obtained (Merlet, 2006).

For our Divide and Conquer kinematic calibration strategy we adopt the inverse calibration method. The method takes advantage of an intrinsic characteristic of parallel mechanisms: the straightforward calculation of the inverse kinematics. However, not all the intrinsic characteristics of parallel mechanisms are exploited. Specifically, (Zhuang et al., 1998; Ryu and Rauf, 2001) reported that for parallel mechanism, methods based on inverse kinematics allow to identify error parameters of each leg of the mechanism independently. The independent parameter identification of each leg is reported to improve the numerical efficiency of the kinematic identification algorithm, (Zhuang et al., 1998). However, it is not reported a general kinematic identification strategy based on the independent identification of the legs and its advantages with respect to traditional identification methods.

This article presents a contribution to the kinematic calibration of parallel mechanisms developing a kinematic identification protocol based on the inverse calibration method and on the independent identification of the parameters of each leg (Divide and Conquer strategy).

With respect to traditional identification methods, our Divide and Conquer strategy has the following advantages:
1. The identification poses can be optimized to the identification of reduced sets of parameters (the sets corresponding to each leg).
2. The independent identification of the parameters

of each leg improves the numerical efficiency of the identification algorithms.

3. By 1. and 2. the identified set of parameters is closer to the real (unknown) set of parameters than sets identified by other traditional calibration methods.

The divide and Conquer strategy for the independent kinematic identification of the parameters of each leg in a parallel symmetrical mechanism is presented in section 3.

3 DIVIDE AND CONQUER IDENTIFICATION STRATEGY

Parallel symmetrical mechanisms satisfy (Tsai, 1999):

1. The number of legs is equal to the number of degrees of freedom of the end-effector.

2. All the legs have an identical structure. This is, each leg has the same number of active and passive joints and the joints are arranged in an identical pattern.

In a practical way, the definition of parallel symmetrical mechanism covers most of the industrial parallel structures. For parallel symmetrical mechanisms the kinematic identification by inverse kinematics and a divide and conquer strategy is stated for each leg κ independently, $\kappa = 1, 2, \ldots, n_{limbs}$.

Given:

1. A set of nominal kinematic parameters of the κth leg (φ_κ). n_κ parameters are assumed to be identified:

$$\varphi_\kappa = \left[\varphi_{\kappa,1} \quad \ldots \varphi_{\kappa,n_\kappa}\right]^T. \tag{1}$$

2. An inverse kinematic function g_κ relating the κth active joint variable ($\mathbf{q}_\kappa$) with the end-effector pose ($\mathbf{r}$). For the jth pose of the mechanism the inverse function of the κth leg is defined to be:

$$\begin{gathered} g_\kappa^j : \varphi_\kappa \times \mathbf{r}^j \to \mathbf{q}_\kappa^j, \\ j = 1, 2, \ldots, N. \end{gathered} \tag{2}$$

3. A set of N end-effector measured configurations ($\hat{\mathbf{R}}_\kappa$) for the identification of the κth leg:

$$\hat{\mathbf{R}}_\kappa = \left[\hat{\mathbf{r}}_\kappa^1 \quad \cdots \quad \hat{\mathbf{r}}_\kappa^N\right]^T. \tag{3}$$

4. A set of measured input variables ($\hat{\mathbf{Q}}_\kappa$) corresponding to the set of end-effector measurements ($\hat{\mathbf{R}}_\kappa$):

$$\hat{\mathbf{Q}}_\kappa = \left[\hat{\mathbf{q}}_\kappa^1 \quad \cdots \quad \hat{\mathbf{q}}_\kappa^N\right]^T. \tag{4}$$

Goal. To find the set of unknown (real) kinematic parameters ($\bar{\varphi}_\kappa$) that minimizes an error between the measured joint variables ($\hat{\mathbf{Q}}_\kappa$) and their corresponding values ($\bar{\mathbf{Q}}_\kappa$) estimated from the measured end-effector poses by the inverse kinematic model g_κ. The problem can be formally stated as the following nonlinear minimization problem:

$$\begin{gathered} \bar{\varphi}_\kappa : \sum_{j=1}^{N} \left\|\hat{\mathbf{Q}}_\kappa - \bar{\mathbf{Q}}_\kappa\left(\hat{\mathbf{R}}_\kappa, \varphi_\kappa\right)\right\|^2 \text{is minimum,} \\ \text{subject to} : \mathbf{R}_\kappa \subset \mathbf{W}_\mathbf{R}, \\ \mathbf{W}_\mathbf{R} \text{ is the usable end} - \text{effector workspace.} \end{gathered} \tag{5}$$

The optimization problem is constrained by the useful workspace (a workspace without singularities) of the mechanism.

A kinematic identification of parallel symmetrical mechanisms protocol based on the Divide and Conquer identification strategy is developed in section 4.

4 KINEMATIC IDENTIFICATION PROTOCOL

Based on the Divide and Conquer strategy for the kinematic identification of parallel symmetrical mechanisms (section 3) the following kinematic identification protocol (Figure 1) is proposed.

1. Given the nominal parameters of the κth leg (φ_κ, Eq. 1) and the correspondent inverse kinematic function (g_κ, Eq. 2) to calculate the κth Jacobian identification matrix of a representative set of postures of the usable workspace:

$$C_\kappa(\mathbf{W}_\mathbf{R}, \varphi_\kappa) = \frac{\partial g_\kappa(\mathbf{W}_\mathbf{R}, \varphi_\kappa)}{\partial \varphi_\kappa^T}. \tag{6}$$

2. Given the Jacobian identification matrix calculated in the first step to select an optimal set of postures ($\mathbf{R}_\kappa(C_\kappa)$) for the kinematic identification of the κth leg. The set of postures is selected searching the improvement of the observability of the set of parameters φ_κ. To select the poses we adopt the active calibration algorithm developed by (Sun and Hollerbach, 2008) that reduces the complexity of computing an observability index reducing computational time for finding optimal poses. The optimized identification set of postures is then defined in the following manner:

$$\begin{gathered} \mathbf{R}_\kappa : O_1(\mathbf{R}_\kappa) \text{ is maximal,} \\ O_1(\mathbf{R}_\kappa) = \frac{\sqrt[n_\kappa]{s_1\ s_2 \cdots s_{n_\kappa}}}{n_\kappa}, \\ \mathbf{R}_\kappa \subset \mathbf{W}_\mathbf{R}, \end{gathered} \tag{7}$$

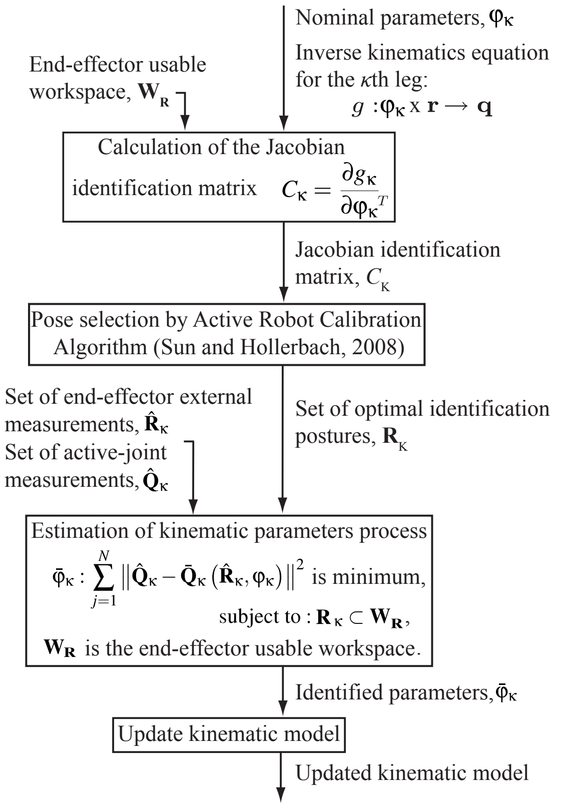

Nominal parameters, φ_κ
End-effector usable workspace, $\mathbf{W}_\mathbf{R}$
Inverse kinematics equation for the κth leg: $g : \varphi_\kappa \times \mathbf{r} \rightarrow \mathbf{q}$
Calculation of the Jacobian identification matrix $C_\kappa = \frac{\partial g_\kappa}{\partial \varphi_\kappa{}^T}$
Jacobian identification matrix, C_κ
Pose selection by Active Robot Calibration Algorithm (Sun and Hollerbach, 2008)
Set of end-effector external measurements, $\hat{\mathbf{R}}_\kappa$
Set of active-joint measurements, $\hat{\mathbf{Q}}_\kappa$
Set of optimal identification postures, $\mathbf{R}_\kappa$
Estimation of kinematic parameters process
$\bar{\varphi}_\kappa : \sum_{j=1}^{N} \|\hat{\mathbf{Q}}_\kappa - \bar{\mathbf{Q}}_\kappa(\hat{\mathbf{R}}_\kappa, \varphi_\kappa)\|^2$ is minimum,
subject to : $\mathbf{R}_\kappa \subset \mathbf{W}_\mathbf{R}$,
$\mathbf{W}_\mathbf{R}$ is the end-effector usable workspace.
Identified parameters, $\bar{\varphi}_\kappa$
Update kinematic model
Updated kinematic model

Figure 1: Kinematic identification of parallel symmetrical mechanisms protocol.

were O_1 is an observability index of the identification matrix ($C_\kappa(\mathbf{R}_\kappa, \varphi_\kappa)$) of the κth leg, n_κ is the number of parameters to be identified in the κth leg, and s_1, s_2 $\ldots, s_{n_\kappa}$ are the singular values of the identification matrix C_κ. As a rule of thumb, in order to suppress the influence of measurement noise, the number of identification poses should be two or three times larger than the number of parameters to be estimated (Jang et al., 2001).

3. Given the optimized set of identification postures obtained in the second step and the correspondent sets of active joint ($\hat{\mathbf{Q}}_\kappa$) and end-effector ($\hat{\mathbf{R}}_\kappa$) measurements to solve the optimization problem defined on Eq. 5 for the identification of the kinematic parameters (φ_κ) of the κth leg.

4. Given the identified set of parameters of the κth leg obtained in the third step to update the kinematic model of the parallel mechanism.

The protocol is repeated until all the legs in the mechanism are identified.

With respect to traditional identification algorithms for the kinematic identification of parallel mechanism (Renaud et al., 2006; Zhuang et al., 1998) the proposed kinematic identification protocol has the following advantages:

1. Reduction of the kinematic identification computational costs. If a linear least-squares estimation of the kinematic parameters is used to solve the identification problem (Eq. 5), then the correction to be applied to the kinematic parameters ($\Delta\varphi$) can be estimated iteratively as (Hollerbach and Wampler, 1996):

$$\Delta\varphi = \left(C^T C\right)^{-1} C^T \Delta\mathbf{Q}. \tag{8}$$

The computational cost of the matrix inversion $(C^T C)^{-1}$ is reduced proportionally to the square of the number of legs of the mechanism, Table 1.

2. Improvement of the numerical efficiency of the kinematic identification algorithm by the independent identification of the parameters of each leg.

3. Improvement of the kinematic identification by the design of independent experiments optimized for the identification of each leg.

Table 1: Computational and measurement costs of kinematic identification.

	Traditional kinematic identification	Divide and conquer identification
Regressor	$C^T C(N\, n_{limbs} \times N\, n_{limbs})$	$C_\kappa^T C_\kappa(N \times N)$
Computational cost (Matrix inversion)	$\propto N^3\, n_{limbs}{}^3$	$\propto N^3\, n_{limbs}$

The kinematic identification of parallel mechanisms protocol is used in the simulated identification of a planar 5R symmetrical mechanism in section 5.

5 RESULTS

The results on kinematic identification of parallel mechanisms by a Divide and Conquer strategy are presented using a case study: the simulated kinematic identification of the planar 5R symmetrical parallel mechanism.

The planar 5R symmetrical mechanism (Figure 2) has two degrees-of-freedom (DOF) that allows it to position the end-effector point (P) in the plane that contains the mechanism. The mechanism is formed by two driving links (l_1 and l_2) and a conducted dyad (L_1 and L_2), Figure 2. The planar 5R symmetrical mechanism is an instance of the parallel symmetrical mechanisms defined in section 3. A complete characterization of the assembly configurations (Cervantes-Sánchez et al., 2000), kinematic design (Cervantes-Sánchez et al., 2001; Liu et al., 2006a; Liu et al., 2006b; Liu et al., 2006c), workspace (Cervantes-Sánchez et al., 2001; Cervantes-Sánchez et al., 2000; Liu et al., 2006a), singularities (Cervantes-Sánchez et al., 2001; Cervantes-Sánchez et al., 2000; Liu et al., 2006a) and performance atlases (Liu et al., 2006b) are reported. However, no research is reported on kinematic identification.

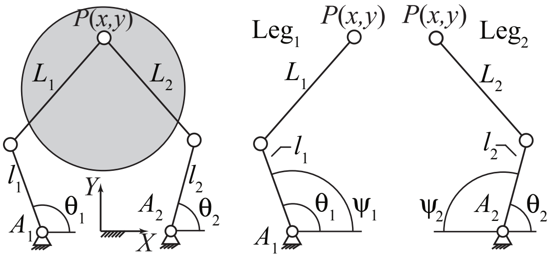


Figure 2: Planar 5R symmetrical mechanism.

The kinematic identification of the planar 5R symmetrical mechanism is simulated using the kinematic identification of parallel symmetrical mechanisms protocol (section 4) under the following conditions:

1. A linear model is assumed for the active joint A_κ:

$$\theta_\kappa = k\psi_\kappa + \gamma_\kappa, \tag{9}$$

where the k_κ represent the joint gain, γ_κ is the joint offset, ψ_κ is the measured active joint angle and θ_κ is the active joint angle, $\kappa = 1,2$.

2. In parallel mechanisms the principal source of error in positioning is due to limited knowledge of the joint centers, leg lengths and active joint parameters (Daney et al., 2002). In consequence, the parameters to be estimated are the attachment points (A_κ), the leg lengths (l_κ, L_κ), and the joint gain and offset (k_κ, γ_κ), $\kappa = 1,2$:

$$\varphi_\kappa = [l_\kappa \; L_\kappa \; A_{\kappa x} \; A_{\kappa y} \; k_\kappa \; \gamma_\kappa]^T . \tag{10}$$

3. The external parameters associated with the measuring device will not be identified. For the external measuring system this implies that its position is known and coincident with the reference frame $X - Y$ and the measurement target is coincident with the end-effector point.

4. The nominal kinematic parameters of the mechanism are disturbed adding a random error with normal distribution and a standard deviation σ. The nominal and disturbed parameters are shown in Table 2.

5. The constrain equation of the inverse kinematics is defined independently for the κth leg (Liu et al., 2006a), $\kappa = 1,2$:

$$L_\kappa{}^2 = (x - l_\kappa \cos\theta_\kappa - A_{\kappa x})^2 + (y - l_\kappa \sin\theta_\kappa - A_{\kappa y})^2. \tag{11}$$

6. The end-effector and joint workspace are limited by the maximal inscribed workspace (MIW), Figure 2. The MIW corresponds to the maximum singularity-free-end-effector workspace limited by a circle (Liu et al., 2006c).

Table 2: Planar 5R symmetrical mechanism. Nominal and real (disturbed) parameters.

	Nominal value	Real value		Nominal value	Real value
A_{1x} [m]	-0.5000	-0.4988	A_{2x} [m]	0.5000	0.4961
A_{1y} [m]	0.0000	0.0028	A_{2y} [m]	0.0000	0.0066
k_1	1.0000	1.004	k_2	-1.0000	-0.9984
γ_1 [rad]	0.0000	0.0048	γ_2 [rad]	3.1416	3.1418
l_1 [m]	0.7500	0.7507	l_2 [m]	0.7500	0.7559
L_1 [m]	1.1000	1.0995	L_2 [m]	1.1000	1.0959

7. Each leg is identified using a set of 18 postures of the mechanism to measure the end-effector position and the corresponding active joint variable. The designed sets of identification postures in the end-effector workspace are presented in Figure 4b.

8. The set of end-effector measurements $(\hat{\mathbf{R}}_\kappa)$ and its corresponding active joint measurements $(\hat{\mathbf{Q}}_\kappa)$ are simulated using forward kinematics and adding random disturbances with normal distribution and standard deviation $\sigma = 1 \cdot 10^{-4}$.

9. A linearization of the inverse kinematics is used for iteratively solving the non-linear optimization problem (Eq. 5), then, for the jth identification pose the identification problem of the κth leg is in the following form:

$$\begin{aligned} \Delta\mathbf{q}_\kappa^j &= \frac{\partial g_\kappa^j}{\partial\varphi_\kappa}\Delta\varphi_\kappa = C_\kappa^j \Delta\varphi, \\ \Delta\mathbf{q}_\kappa^j &= \hat{\mathbf{q}}_\kappa^j - \bar{\mathbf{q}}_\kappa^j, \\ \Delta\varphi_\kappa &= \bar{\varphi}_\kappa - \varphi_\kappa. \end{aligned} \tag{12}$$

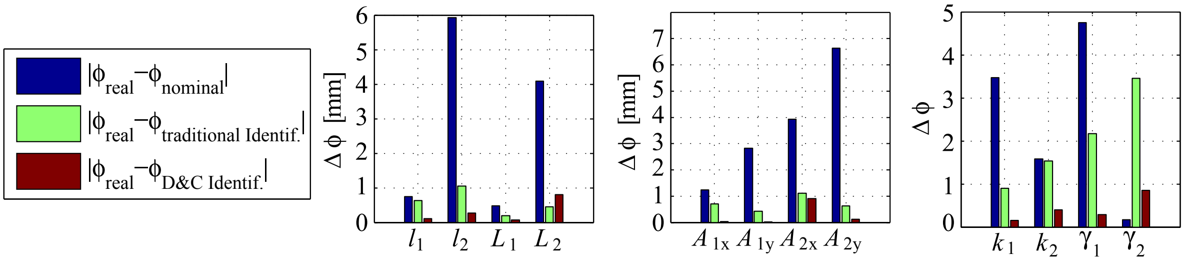
Using $N = 18$ measurements to identify the set of parameters φ_κ the identification problem is stated in the following manner:

$$\begin{aligned} \Delta\mathbf{Q}_\kappa &= C_\kappa \Delta\varphi_\kappa, \\ C_\kappa &= \left[C_\kappa^1 \cdots C_\kappa^N\right]^T, \\ \Delta\mathbf{Q}_\kappa &= \left[\Delta\mathbf{q}_\kappa^1 \cdots \Delta\mathbf{q}_\kappa^N\right]^T, \end{aligned} \tag{13}$$

were C_κ is the identification matrix of the κth leg. The parameters of the κth leg can be updated using a linear least-squares solution of Eq. 13, (Hollerbach and Wampler, 1996):

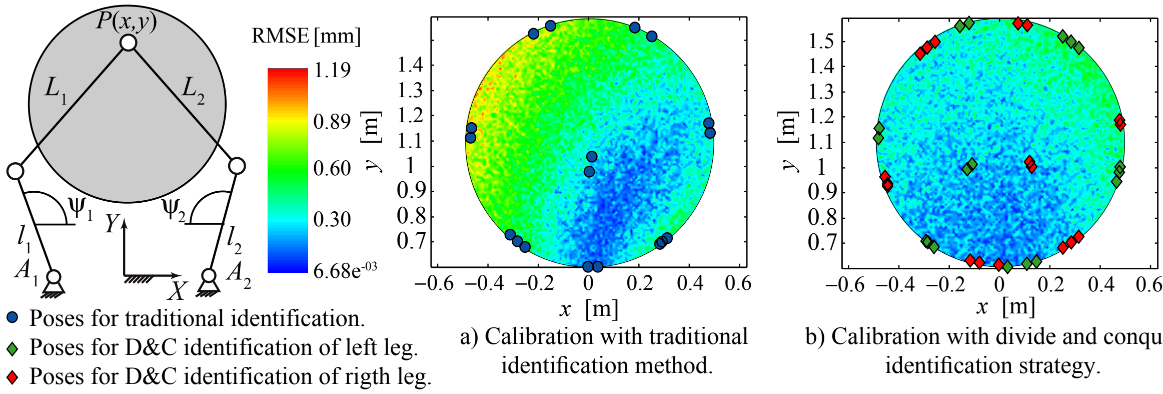
$$\Delta\varphi_\kappa = (C_\kappa^T C_\kappa)^{-1} C_\kappa^T \Delta\mathbf{Q}_\kappa. \tag{14}$$

10. An alternative traditional kinematic identification by inverse kinematic modeling is calculated and used as a comparison with respect to the proposed kinematic identification protocol. The traditional identification is performed by means of a set of 36 optimized postures selected in order to maximize the observability of the total identification matrix. The observability



a. Link lengths b. Position of fixed joints c. Active joint parameters

Figure 3: Planar 5R mechanism. Residual errors in the kinematic parameters before and after calibration.



a) Calibration with traditional identification method.

b) Calibration with divide and conquer identification strategy.

Figure 4: Planar 5R mechanism. Estimated end-effector local root mean square error for the maximal inscribed workspace (MIW) after calibration.

was defined as the Eq. 7. The designed set of identification postures is presented in Figure 4a.

The results of the kinematic identification under these conditions are presented in Figure 3 (residual errors in kinematic parameters before and after calibration). The residual errors are calculated as the difference between the real (virtually disturbed) parameters and the estimated parameters. Finally, Figure 4 presents the estimated local root mean square error for the MIW after calibration and the selected postures for kinematic identification. Additionally the computational and measurement identification costs are estimated for the identification of the planar 5R parallel mechanism, Table 3. The measurement costs of the Divide and Conquer strategy are incremented with respect to a traditional identification method. The increment of the measurements is required because each leg requires an independent set of end-effector measurements. In the case of a traditional identification the set of end-effector measurements is common to all the legs. In despite of the measurement increment the Divide and Conquer identification results in a superior estimation with respect to a traditional kinematic identification methods (Renaud et al., 2006; Zhuang et al., 1998). The conclusions of the paper are proposed in section 6.

Table 3: Planar 5R symmetrical mechanism. Computational and measurement costs of kinematic identification.

	Traditional kinematic identification	Divide and conquer identification
Regressor	$C^T C(36 \times 36)$	$C_\kappa^T C_\kappa(18 \times 18)$
Computational cost (Matrix inversion)	$\propto 18^3 \cdot 2^3$	$\propto 18^3 \cdot 2$
Measurement cost	$2 \cdot 18 \cdot 2 = 72$	$18 \cdot 2(2+1) = 108$

6 CONCLUSIONS

This article presents a new (Divide and Conquer) strategy for the kinematic identification of parallel symmetrical mechanisms. The new strategy develops a formalization of the inverse calibration method proposed by (Zhuang et al., 1998). The identification strategy (section 3) is based on the independent identification of the kinematic parameters of each leg of the parallel mechanism by minimizing an error between the measured active joint variable of the identified leg and their corresponding value, estimated through an inverse kinematic model. With respect to traditional identification methods the Divide and Conquer strategy presents the following advantages:

1. Reduction of the kinematic identification computa-

tional costs,
2. Improvement of the numerical efficiency of the kinematic identification algorithm and,
3. Improvement of the kinematic identification results.

Based on the Divide and Conquer strategy, a new protocol for the kinematic identification of parallel symmetrical mechanisms is proposed (section 4, Figure 1). For the selection of optimal identification postures the protocol adopts the active robot calibration algorithm of (Sun and Hollerbach, 2008). The main advantage of the active robot calibration algorithm is the reduction of the complexity of computing an observability index for the kinematic identification, allowing to afford more candidate poses in the optimal pose selection search. The kinematic identification protocol summarizes the advantages of the Divide and Conquer identification strategy and the advantages of the active robot calibration algorithm.

The kinematic identification protocol is demonstrated with the simulated identification of a planar 5R symmetrical mechanism (section 5). The performance of our identification protocol is compared with a traditional identification method obtaining an improvement of the identification results (Figs. 3, 4).

ACKNOWLEDGEMENTS

The authors wish to acknowledge the financial support for this research by the Colombian Administrative Department of Sciences, Technology and Innovation (COLCIENCIAS), grant 1216-479-22001.

REFERENCES

Besnard, S. and Khalil, W. (2001). Identifiable parameters for parallel robots kinematic calibration. In *IEEE International Conference on Robotics and Automation, 2001. Proceedings 2001 ICRA*, volume 3.

Cervantes-Sánchez, J. J., Hernández-Rodríguez, J. C., and Angeles, J. (2001). On the kinematic design of the 5r planar, symmetric manipulator. *Mechanism and Machine Theory*, 36(11-12):1301 – 1313.

Cervantes-Sánchez, J. J., Hernández-Rodríguez, J. C., and Rendón-Sánchez, J. G. (2000). On the workspace, assembly configurations and singularity curves of the rrrrr-type planar manipulator. *Mechanism and Machine Theory*, 35(8):1117 – 1139.

Daney, D., Lorraine, I., and LORIA, V. (2002). Optimal measurement configurations for Gough platform calibration. In *IEEE International Conference on Robotics and Automation, 2002. Proceedings. ICRA'02*, volume 1.

Daney, D., Papegay, Y., and Madeline, B. (2005). Choosing measurement poses for robot calibration with the local convergence method and Tabu search. *The International Journal of Robotics Research*, 24(6):501.

Hollerbach, J., Khalil, W., and Gautier, M. (2008). *Springer Handbook of Robotics, Model Identification*. Springer.

Hollerbach, J. and Wampler, C. (1996). The calibration index and taxonomy for robot kinematic calibration methods. *The International Journal of Robotics Research*.

Jang, J., Kim, S., and Kwak, Y. (2001). Calibration of geometric and non-geometric errors of an industrial robot. *Robotica*, 19(03):311–321.

Liu, X., Wang, J., and Pritschow, G. (2006a). Kinematics, singularity and workspace of planar 5r symmetrical parallel mechanisms. *Mechanism and Machine Theory*, 41(2):145 – 169.

Liu, X., Wang, J., and Pritschow, G. (2006b). Performance atlases and optimum design of planar 5r symmetrical parallel mechanisms. *Mechanism and Machine Theory*, 41(2):119 – 144.

Liu, X., Wang, J., and Zheng, H. (2006c). Optimum design of the 5r symmetrical parallel manipulator with a surrounded and good-condition workspace. *Robotics and Autonomous Systems*, 54(3):221 – 233.

Merlet, J. (2006). *Parallel Robots*. Springer Netherlands, second edition.

Renaud, P., Vivas, A., Andreff, N., Poignet, P., Martinet, P., Pierrot, F., and Company, O. (2006). Kinematic and dynamic identification of parallel mechanisms. *Control Engineering Practice*, 14(9):1099–1109.

Ryu, J. and Rauf, A. (2001). A new method for fully autonomous calibration of parallel manipulators using a constraint link. In *2001 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, 2001. Proceedings*, volume 1.

Sun, Y. and Hollerbach, J. (2008). Active robot calibration algorithm. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 1276–1281.

Tsai, L. (1999). *Robot analysis: the mechanics of serial and parallel manipulators*. Wiley-Interscience.

Zhuang, H., Yan, J., and Masory, O. (1998). Calibration of Stewart platforms and other parallel manipulators by minimizing inverse kinematic residuals. *Journal of Robotic Systems*, 15(7).

A CASTOR WHEEL CONTROLLER FOR DIFFERENTIAL DRIVE WHEELCHAIRS

Bernd Gersdorf
Safe and Secure Cognitive Systems, German Research Center for Artificial Intelligence, Bremen, Germany
bernd.gersdorf@dfki.de

Shi Hui
SFB/TR8 Spatial Cognition, University of Bremen, Germany
shi@informatik.uni-bremen.de

Keywords: Differential drive vehicle, Electric wheelchair, Castor wheel, Motion controller, Force compensation.

Abstract: This paper describes a motion controller for differential drive vehicles with a compensation for castor wheel turn forces. The controller has been developed for electric wheelchairs with two front castor wheels, which require large steering forces to move into the direction given by the joystick in situations where a sharp turn of the castor wheels is needed. It computes the required forces to turn the castor wheels and modifies the drive request adequately. This also helps physical or cognitive impaired users who otherwise have to compensate castor turn forces manually using the joystick, which requires fast reaction time (e.g. when the castor wheels turn quickly into the driving direction) to avoid collisions.

1 INTRODUCTION

Figure 1 presents the intelligent wheelchair *ROLLAND* (Lankenau and Röfer, 2001) based on a Meyra wheelchair of the model CHAMP (Meyra Ortopedia, 2010). As many other electric wheelchairs, CHAMP uses pneumatic tires for both back differential drive wheels and front castor wheels to achieve comfortable indoor and outdoor driving. The configuration of back differential drive and two front castor wheels is probably the most frequently used one for electric wheelchairs. With the large ground contact area of the castor wheels, the high load (typical 160kg for the wheelchair with the driver, about one third on the front axis), and the tire material (rubber), the additional force to change the angle of the castor wheel is significant, especially when maneuvering at low speed in a narrow environment. Recently, an empirical study (*reference skipped*) on the evaluation of the safety assistant modul developed for *ROLLAND* was carried out, which monitors the surrounding environment using sensor data gathered by the equipped laser scanners and brakes in time if an obstacle is dangerously close to the wheelchair. During the experiment a common phenomenon was observed: after the intervention of the safety assistant, the participants attempted to regain control over the wheelchair by giving driving commands via the joystick, but the wheelchair did not drive in the direction they expected or did not move at all.

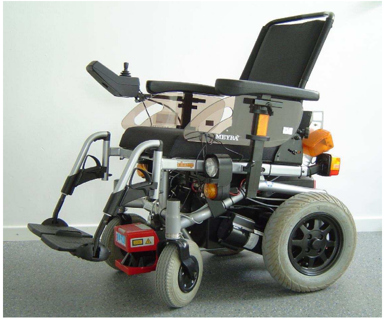


Figure 1: The intelligent wheelchair *ROLLAND*.

The primary reason for such problems is that the wheelchair requires a lot of motor force to turn the castor wheels, if the wheelchair starts in a standing position with castor wheels positioned in a blocking state (i.e., the two castor wheels stand transversely to the required drive direction). Moreover, users often change the joystick command if the wheelchair does not react to the previously given command, as most participants did in the above study. As a result they tried to give the wheelchair some arbitrary commands via the joystick or pressed the joystick powerfully, which caused an even less expected behaviour of the wheelchair. One solution to this problem, and as the focus of the current paper, is to apply a castor wheel controller, which enables the wheelchair to adjust castor wheels correctly, such that the time delay to realize the requested drive command can be reduced in those situations, and the joystick can remain in a stable position.

Castor wheels have been investigated in the literature mainly to model and minimize wheel shimmy (see (de Falco et al., 2009), (Brearley, 2009), (Kauzlarich et al., 2000)). In (Kauzlarich et al., 1984), castor turn forces are compared for different grounds and tire materials, but without a projection of these forces to the differential drive of an electric wheelchair. Therefore no compensation of these forces is discussed in that work.

This paper is structured as follows: We begin in Section 2 with the kinematic model for differential drive wheelchair with two front castor wheels, and introduce a model to predict castor turn forces and discuss possible compensation. Section 3 explains the integration of the castor force compensation into *ROLLAND*. Section 4 discusses some test results by comparing the wheelchair behaviour with and without the castor controller. Before concluding in Section 6 we discuss some related approaches in Section 5.

2 A KINEMATIC MODEL

We are going to present a kinematic model of castor turn forces for differential drive wheelchairs with two front castor wheels. The castor wheel angles are essential for the required turn forces, thus should be discussed first.

2.1 Castor Wheel Angles

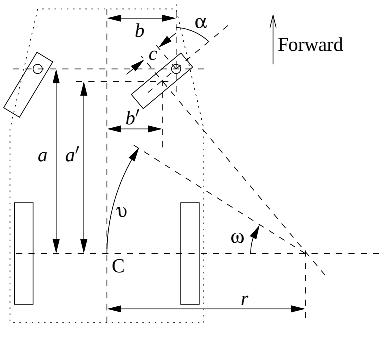


Figure 2: Determination of castor wheel angles.

To compute the required force to turn the castor wheels, the castor angles must be computed based on the current driving request. The motion state of differential drive vehicles can be described by its rotational speed ω (rad/s) and translational speed υ (m/s) measured at the center point C between the powered wheels. The radius r (see Figure 2) is determined by

$$\frac{\upsilon}{r} = \omega \Leftrightarrow r = \frac{\upsilon}{\omega} \tag{1}$$

However, there is always a small gap c between a castor's ground touch point and the intersection of its steering axis with the ground plane (also called *trail*). Furthermore, the steering axis inclination is usually 0 or almost 0 for castor wheels, and can be assumed to be 0. If c equals 0, the angle of the castor can be obtained (using trigonometry) by

$$\tan\alpha = \frac{r-b}{a} \tag{2}$$

For $0 < c \ll a$ (as usual), the castor wheel angle α obtained by Equation 2 is already a good approximation. However, a higher precision α' can be obtained using the ground touch point at distances a' and b' derived from the first estimation of the castor wheel angle α and the value of the gap c (see Equation 5), with which more precise castor turn forces can be calculated.

$$a' = a - c * \cos\alpha \tag{3}$$

$$b' = a - c * \sin\alpha \tag{4}$$

$$\tan\alpha' = \frac{r-b'}{a'} \tag{5}$$

2.2 Castor Turn Forces

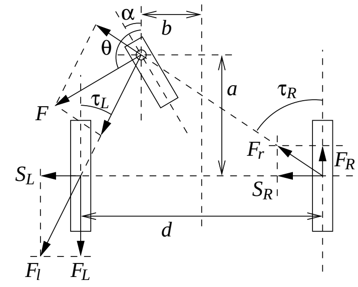


Figure 3: Castor turn force F projected to differential drive wheels.

Figure 3 shows the geometry of the *ROLLAND* electric wheelchair as an example for a differential drive robot with two castor wheels. Let the *castor turn force F* be the force that is required to start the castor wheel to swivel without rotating the wheel and applied at the castor steering joint. This force must be applied perpendicular to the driving direction of the castor wheel. F can be split into

$$F = F_{ground} + F_{joint} \tag{6}$$

where F_{ground} represents the friction between the ground and the tire at the ground touching point, and F_{joint} the friction inside the turning joint. The joint is typically so constructed that $F_{joint} \ll F_{ground}$. A difference between sticking and gliding forces could not be observed, due to the plasticity of the tire. As frictional forces grow proportionally with the normal force between the surfaces, the castor turn forces can be described as

$$F = c_{castor} L \tag{7}$$

where L is the load of the castor wheel and c_{castor} is constant for a given wheelchair and ground material. Equation 7 must be understood as an upper limit of the required castor force. A rolling castor wheel requires much smaller turn forces, which should be considered by the castor controller (see Section 3).

The castor turn force can be projected to the forces of the two differential drive wheels, F_L and F_R (see Figure 3):

$$\theta = \alpha \pm \frac{\pi}{2} \tag{8}$$

$$\overline{F} = \overline{F_l} + \overline{F_r} \tag{9}$$

$$F_L = F_l \cos \tau_l \tag{10}$$

$$\tan \tau_L = \frac{a}{b - \frac{d}{2}} \tag{11}$$

$$F_R = F_r \cos \tau_r \tag{12}$$

$$\tan \tau_R = \frac{a}{b + \frac{d}{2}} \tag{13}$$

$+$ or $-$ in Equation 8 is chosen depending on the intended castor steering direction (left or right swivel). Equation 9 splits the castor correction force $\overline{F}$ into forces of the differential drive wheels. The left engine produces $\overline{F_l}$ as the sum of the motor force $\overline{F_L}$ and a shearing force $\overline{S_L}$ (see Figure 3). The left engine force F_L is computed by Equations 10 and 11, and the right one by Equations 12 and 13.

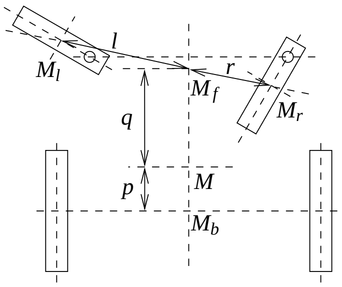


Figure 4: Mass distribution to castor and differential drive wheels.

Figure 4 shows the center of mass M of the wheelchair at the point with a distance of p to the back axis. The back wheels are connected with springs to the chassis, such that both wheels can be assumed to carry the same weight that sums up to M_b at the back axis center. The front wheels carry the remaining mass M_f, giving the following mass distribution:

$$M_f = M \frac{p}{p+q} \tag{14}$$

$$M_l = \frac{r}{l+r} M_f \tag{15}$$

$$M_r = \frac{l}{l+r} M_f \tag{16}$$

The values for p, l, and r can be obtained from the ground touch points as in Figure 2.

2.3 Maximal Castor Turn Forces

Using the kinematic model of Sections 2.1 and 2.2, Figure 5 gives an overview of the forces required by the differential drive vehicle using the geometry of the Meyra CHAMP wheelchair (in mm: $d = 585, p = 160, a = 470, b = 42$). For each value of the left castor angle (x-axis), the diagram shows the angle of the right castor wheel, which maximises the required correction force for one of the two differential drive motors. The force factor (right y-axis in Figure 5) is the relation between the required motor force and the standard situation in the left diagram of Figure 6, where castor turn forces and engine forces are all equal, and the castor wheels are in a blocking state when driving forward. The diagram in Figure 5 shows two symmetric maximal force factors -2.21 (left engine) and 2.21 (right engine). The -2.21 maximum for the left drive wheel is also shown in the right diagram of Figure 6 with castor angles $L = 46.08°$, $R = 8.28°$.

3 MOTION CONTROLLER

The kinematic model described in Section 2 has been implemented and integrated into the wheelchair *ROLLAND* in two ways:

- as an add-on for a classical proportional-integral (PI) controller using odometry information to control translational and rotational wheelchair speed (used for autonomous driving).
- as a pure castor force compensation controller to support joystick driving.

Although the tests reported in Section 4 have been run with the pure castor controller, both controllers have been implemented using the following principles:

1. The castor controller applies a slowly growing correction force. This helps to adapt to different

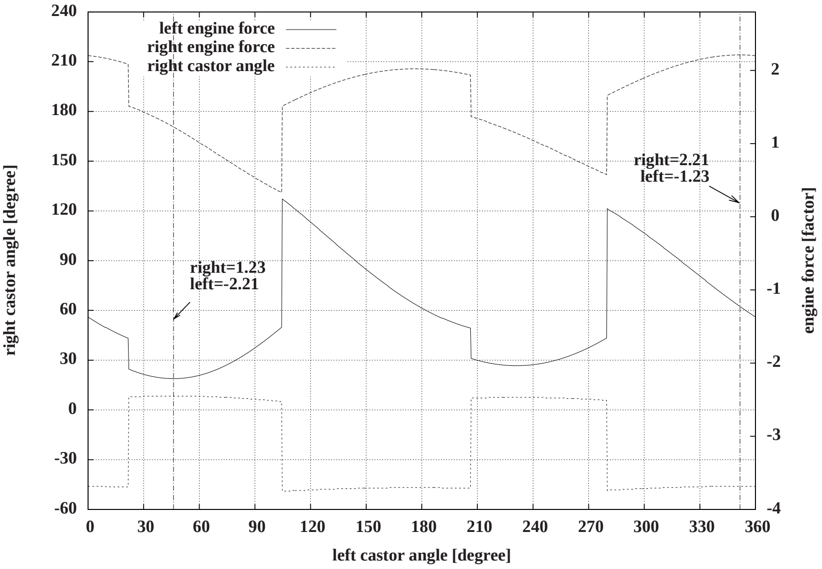


Figure 5: Castor angles with maximal differential drive load.

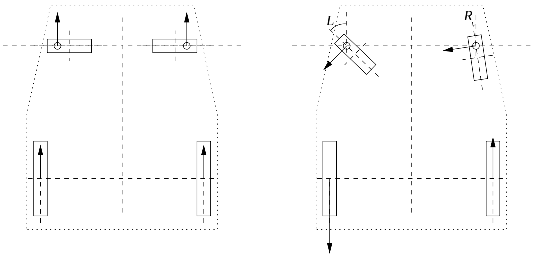


Figure 6: Castor correction forces in standard (left) and maximized situation (right).

frictional forces for varying undergrounds, standing or rolling castor wheels. It smoothly applies forces for the transition from *standing still* (no correction forces applied) to *driving* (correction applied).

2. The correction force will be significantly reduced, if the castor wheel turns too fast.
3. The correction force is reduced, when the difference between actual and requested castor angle is small.
4. If the given command changes the driving direction, the control intensity is reduced. A control intensity built for a specific driving direction should not be applied for a completely different driving direction.
5. The correction force of the controller is reduced to zero, if one of the castor correction forces changes its sign (the castor angle has crossed its target angle). This allows to reach the target castor angle exactly and reduces oscillations around the ideal castor angle.
6. The mass distribution to each individual wheel can be significantly influenced by the differential drive forces. The Meyra CHAMP, for example, has an independent wheel suspension using a trailing arm for each wheel. The forward torque of a wheel moves the trailing arm and its wheel downward, which increases the load of the castor wheel on the opposite side. The measurements of this effect have been used in the controller to correct the mass distribution described in Figure 4.
7. If the differential drive stands still, the engine torques grow nonlinear with the given driving command. The relation between them has been measured and recorded in a table to apply correct forces.

The combination of the castor controller with a PI-controller requires to join the drive requests produced by both controllers. A brief impression can be obtainded by the following two principles:

- For small deviating castor angles, the influence of the castor controller should be small. This gives priority to the PI-controller.
- Correction forces of the castor controller are subtracted from integral parts of the PI-controller. This gives priority to the castor controller, if castor wheels are in a blocking state.

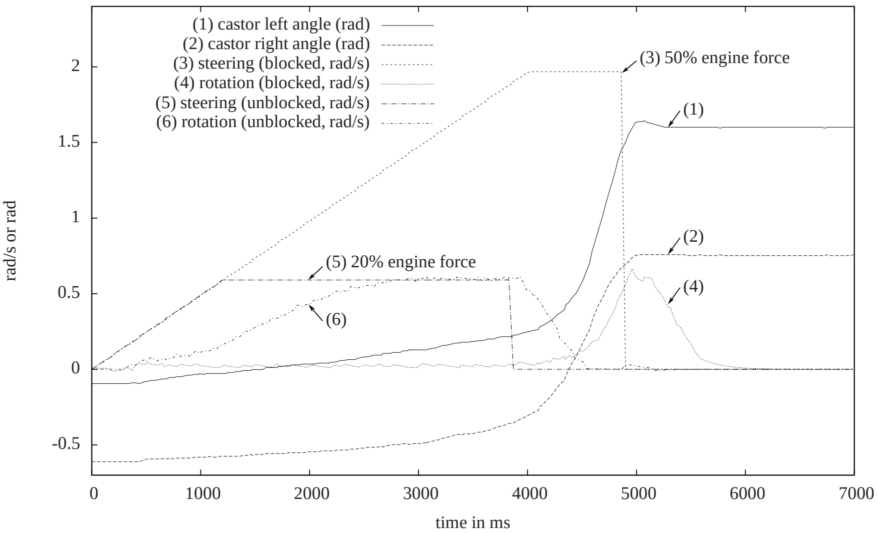


Figure 7: Wheelchair behaviour for a left turn with blocking or adjusted castor wheels.

4 TEST RESULTS

In order to evaluate the influence of the pure castor wheel controller, a number of tests have been carried out using the wheelchair *ROLLAND*. In the following we are going to discuss some test results.

The test shown in Figure 7 compares the reaction of the wheelchair between blocked and unblocked castor wheels while making a left turn without a castor controller. The steering command for blocked (3) and unblocked (5) castor wheels is slowly increased to a level that moves the wheelchair. With unblocked castor wheels, the engine force can be limited to 20% (5) to reach a wheelchair turn speed of about 0.6 rad/s (6). For blocked wheels, the engine force must be increased to 50% (3) to let the castor wheels turn around (angles are shown in (1) and (2)). In this case, the castor wheels start to swivel after about 5 seconds. The steering command is then reduced to zero (during this measurement) to avoid a too fast rotation of the wheelchair. Usually, the steering command must be reduced by the wheelchair user manually.

The adjustment of giving steering commands by the pure castor controller is demonstrated by Figure 8. The diagram shows a left-turn of the wheelchair starting with castor wheels in a blocking state using the castor controller. It compares the given steering command (5) from the user and the motor commands for steering (3) and forward motion (4) computed by the controller. The forward motor command is the mean value of left and right motor command, and the steering motor command the difference between them. The controller uses the driving command and the castor angles of the left (1) and right (2) wheels as input values. It produces a peak for the left engine of about 80% power (computed from forward (3) and steering (4) motor command), and will be reduced when the castor wheels start to swivel. The wheelchair starts to turn around significantly after the peak is reached, as indicated by curve (6) measured by the odometry of the wheelchair. The castor controller reacts much earlier than an odometry based controller and releases the user from doing this manually.

5 RELATED APPROACHES

An alternative approach is to turn the castor wheels using an additional actuator, as for the Toyota PM research vehicle (Toyota Motor Sales USA, 2009; Bonsor, 2004). The trail of the castor wheel can then be 0, which makes it 180° rotation symmetric. The wheelchair manufacturer Otto Bock recently presented the XENO wheelchair with a steering system called S^3 (Single Servo Steering, (Otto Bock HealthCare GmbH, 2010)) in combination with a differential drive using an approach similar to that of the PM research vehicle. For the additional cost of the castor turn servo engines, the Otto Bock system releases the differential drive from the burdon of castor turn forces, but the time to turn the front wheels can be significantly longer than the time needed to turn classical castor wheels. This can confuse a user who is unaware of this fact.

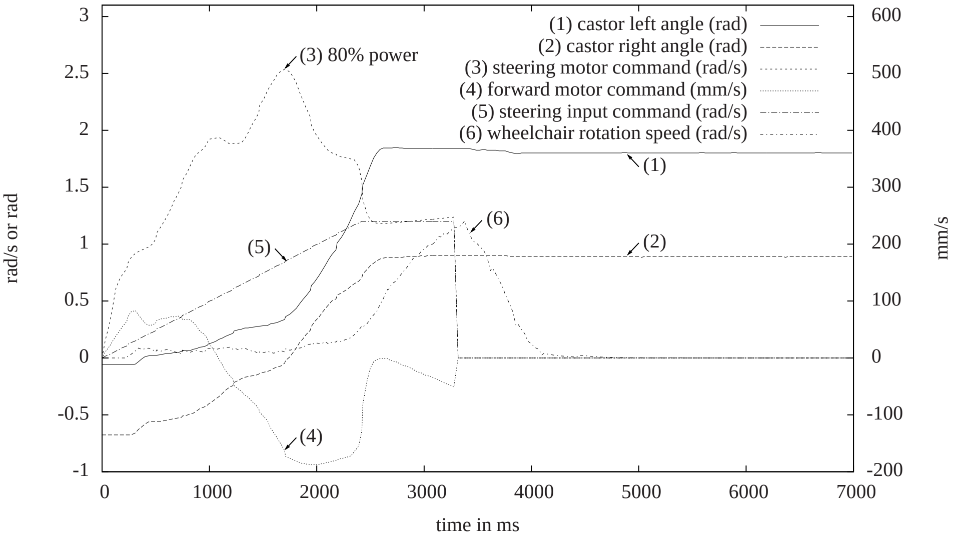


Figure 8: Castor controller for a left turn with blocking castor wheels.

6 CONCLUSIONS

In this paper we presented a castor wheel controller for differential drive wheelchairs with two front castor wheels (e.g., *ROLLAND* based on a Meyra wheelchair of type CHAMP). The controller uses the developed kinematic model to project turn forces of castor wheels to engine torques of differential drive vehicles. Some test data were given to compare the steering behaviour with and without the controller. The major benefit of the castor wheel controller is to release the user from the task to compensate the castor turn forces using the joystick to simplify the driving with the joystick.

One future work is to do more experimental evaluation and performance analysis of the pure and combined castor wheel controllers before carrying out real user studies. Moreover, the measurement of castor angles through potentiometers is unprecise in a certain range of angles. We will replace them by magnetic angle measurement sensors for full coverage.

REFERENCES

Bonsor, K. (2004). How the Toyota PM works. Technical report, Toyota Motor Sales, U.S.A., Inc,.

Brearley, M. N. (2009). Investigation of castor-wheel shimmy. *Journal of Mechanics and Applied Mathematics*, 33(4):491–505.

de Falco, D., Massa, G. D., and Pagano, S. (2009). On the castor dynamic behavior. *Journal of The Franklin Institute*, to appear.

Kauzlarich, J. J. P., Bruning, T. E., and Thacker, J. G. P. (1984). Wheelchair caster shimmy and turning resistance. *Rehabilitation Research and Development*, 20(2):15–29.

Kauzlarich, J. J. P., Bruning III, T. E., and Thacker, J. G. P. (2000). Wheelchair caster shimmy II: Damping. *Rehabilitation Research and Development*, 37(3):305–313.

Lankenau, A. and Röfer, T. (2001). A safe and versatile mobility assistant. *IEEE Robotics and Automation Magazine*, 1(7):29–37.

Meyra Ortopedia (2010). Electric-wheelchairs CHAMP 1.594. http://www.meyra.de/meyraweb/o_basis.pl.

Otto Bock HealthCare GmbH (2010). XENO - stand up and go! http://www.ottobock.com/.

Toyota Motor Sales USA (2009). The personal mobility vehicle. http://www.toyota.com/concept-vehicles/pm.html.

DISTRIBUTED OPTIMIZATION BY WEIGHTED ONTOLOGIES IN MOBILE ROBOT SYSTEMS

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, Technical University of Cluj Napoca, Cluj Napoca, Romania
Department of Computer Science, Technical University of Cluj Napoca, Cluj Napoca, Romania
Gheorghe.Lazea@aut.utcluj.ro, Octavian.Cret@cs.utcluj.ro

Keywords: Mobile cooperating robots, Heterogeneous agents, Subgradient optimization, Distributed ontology.

Abstract: Heterogeneous mobile robots are often required to cooperate in some optimal fashion following specific cost functions. A global cost function is the sum of all agents' cost functions. Under some assumptions it is expected that a provable convergent computation process gives the optimal global cost for the system. For agents that can exchange ontological information via a network, different variables in the global vector are relevant when ontology instances have been recognized and communicated among agents. It means the optimization depends on what is known to each agent at the current time. There are two ways to solve an optimization of this kind: (a) to weight agents according to the ontology instances or (b) to add ontology-defined optimization constraints. This paper illustrates the benefits of the weighted optimization method.

1 INTRODUCTION

Intended applications for the optimization method proposed in this paper are systems of cooperating robots, where each robot is designed using an agent architecture. Each robot has specific sensors and actuators, thus robots and agents are in general heterogeneous.

In order to fulfil goals at an abstract level, agents can exchange ontological information (Văcariu, Chintoanu, Lazea and Creţ, 2007). As shown in Section 2, ontological concepts are mapping the sensed world with non-trivial modes of actuation and goal seeking. Using a symbolic layer with ontologies means that agents can undergo rather discontinuous changes when recognizing different types of environment that are associated to complex actions and goals.

It is typical for applications with many sensors, actuators and high-level of intelligence that the same goal could be achieved in many ways and at different costs. Thus optimization of cost functions of agents is an important requirement for a practical solution. We consider a total number of M agents acting in all the robots; each agent has an index *i* (*i=1,…,M*) and a related convex cost function $f_i(x)$. The argument *x* is a vector in R^n of resources. Initially, agents know their own resources but they have a degree of uncertainty about the resources available to other agents. In time, each agent tries to compute the best estimate of the resource vector *x* that optimizes the agent's own cost function. By exchanging resource vectors and ontologies among agents, the optimization can make a better resource allocation using common resources. Thus, the cost function for each agent converges to an (almost) optimal value and the overall cost function $K = \sum_{i=1}^{M} f_i(x)$ will be also minimized.

In our model, the non-smooth cost functions obtained by discrete changes in resource vectors by ontology update are not differentiable over the whole domain, thus these functions are optimized by a subgradient method (Shor, Kiwiel, and Ruszcaynski, 1985).

2 AGENT MODEL

In traditional applications, the components of a resource vector used for optimization are physical measured values. In more advanced applications, resource vectors change, an interesting case being when agents exchange ontologies.

Here we describe the relationship between the resource vector x and the symbolic part of the agents ontological representation.

We consider that each robot is built on an agent architecture that can host a number of specialized agents. These can use local robot resources. We have a set of cooperating robots *Ro*, indexed from 1 to *M*. The set of sensors of a system of robots is:

$$Se\text{: } Ro \times Ns \tag{1}$$

where *Ns* = number of sensors.

The decisions made by agents are not based directly on raw sensor data. Normally, a so called *perception relation* is the result of processing the data from several sensors. For a robot with *Kr* sensors, a perception relation is defined as a mapping of sensors to indices *Np*:

$$Re\text{: } Se^{Kr} \times Np \tag{2}$$

Perception relations are associated symbolic names by a mapping from the set of perception relations *Re* to a vocabulary (set of words) *Tv* called *symbolic perceptions*:

$$RelS\text{: } |Re| \times N \rightarrow Tv \tag{3}$$

where $|Re|$ is the index of a relation in (2) and *N* is the index of the robot.

Ontology is a formal representation of a set of concepts within a domain and the relationship between these concepts (Noy and McGuinness, 2001). These concepts being specific to a domain, it generally means that there is no obvious one-to-one mapping between ontological concepts and the perception relations. If the set of ontological concepts is *Z*, a robot-specific ontology is defined as the relation between the perception relations and a relationship between a set of *L* ontological concepts:

$$Ont\text{: } Tv \rightarrow Z^L \tag{4}$$

Depending on the inclusion relation between the specific ontology and the perception relations, there are three cases. A specific ontology is (1) *fundamental* when $Se^{Kr} \equiv Z^L$; (2) is *minor* relative to the perception relations if $Se^{Kr} \subset Z^L$; (3) is *major* relative to perception relations if $Se^{Kr} \supset Z^L$.

We consider here *major* specific ontologies.

Agents are using the actuator gear of robots to achieve goals. Actuators are denoted:

$$Ac\text{: } Ro \times Na \tag{5}$$

with *Na* being the actuator index and *Ro* the set of robots. A relation of actuation is a mapping from the symbolic perceptions *Tv* to actuators:

$$RelA\text{: } Tv \rightarrow Ac \tag{6}$$

which shows the intended action for each symbolic perception in *Tv*. Finally, it is assumed that since each action is executed under known conditions, the intended effect of the action is known as a specified next symbolic perception. This is called the *actuation effect relation* (or actuation relation) *EfA*:

$$EfA\text{: } Ac \times Tv \rightarrow Tv \tag{7}$$

The dynamics of the data evaluation is shown in the Figure 1 below:

$$\begin{tikzcd} & & Z^L \\ RelS \arrow[r] & Tv \arrow[ur] \arrow[dr] & \\ & & EfA \end{tikzcd}$$

Figure 1: Data evaluation.

3 MULTI-AGENT MODEL

In the multi-agent system each robot has a number of agents, each agent having a specific set of relations of the type (1) – (5). All agents are assumed to use the same domain ontology since this ensures that common goals can be achieved and that communication is not robot manufacture-dependent. Each agent has only a subset of the common domain ontology, subset that was considered appropriate for its sensors, actuators and goals at the design time. During the cooperation phase, agents are exchanging information such as sensor data, ontology concepts or resource vectors. It is also assumed that all agents have a bounded communication time interval *T*. There are several mechanisms for ontological communication but we describe only the method called Simple Ontological Substitution (SOS).

Let t_a and t_c be two symbolic perceptions in the *Tv* set of a local agent such that according to relation (7), we have $t_c = EfA(a_c,t_a)$ (Figure 2).

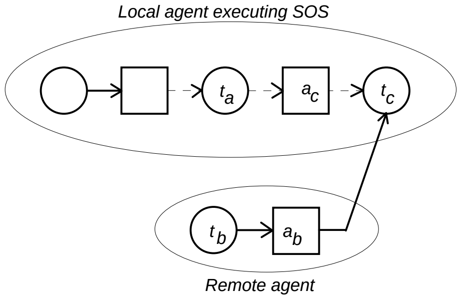


Figure 2: Simple Ontological Substitution.

Formally, the intended sequence towards a goal is as follows: sensors of the local agent expect reading data which is used to interpret the symbolic perception t_a as true. The associated actuation a_c is expected to materialize the symbolic perception t_c. However, the environment of the robot changes such that no symbolic perception among those of the $T\nu$ set of the local agent will become true. The agent cannot recognize the environment state in which it acts. The local agent has information according to its *EfA* relation about the wanted perception to realize t_c. Another agent in some remote robot has a corresponding relation *EfA* of the type $t_c = EfA(\_,\_)$. The local agent is sending a request for all agents in the system, with the following content: (a) a remote agent should have the *EfA* resulting in t_c and (b) the remote agent should send this *EfA* together with the corresponding Z^L (see Figure 1) to the local agent.

If such a remote agent exists, and answers with the $t_c = EfA(a_b,t_b)$, then the local agent will ask for the execution of a_b that will result in t_c. Next time, when a similar situation happens, the local agent has the ontology knowledge to cope with this situation.

4 OPTIMIZATION

4.1 Optimizing Individual Agents

Each agent i, $i = 1,\ldots,n$ has an individual cost function f_i: $R^n \rightarrow R$ expressing e.g. energy, material or wear and tear costs. A minimum value of the cost function means finding a so called optimal resource vector x^* such that:

$$f_i(x^*) \leq f_i(x)\ \forall x \in R^n \tag{8}$$

All cost functions are assumed convex:

$$f(ta + (1 - b)t) \leq tf(a) - (1 - t)f(b) \tag{9}$$

For each convex function a local optimum is also a global optimum. This makes the computation of an optimum easy when the function is smooth (it has derivatives of high order). For example, from the truncated Taylor expansion:

$$f(x+h) \approx f(x) + f'(x)h + \frac{f''(x)}{2}h^2,$$

by differentiation on h we obtain $2h \cdot dh \cdot \frac{f''(x)}{2} + f'(x) \cdot dh = 0$, and after reductions, $h = -\frac{f'(x)}{f''(x)}$. This gives the iteration which is Newton's method for solving $f'(x) = 0$ (k is the iteration index).

$$x_{k+1} = x_k + h \tag{10}$$

The example above shows the two important parts of a program solving an optimization application (Bonnans, Gilbert, Lemaréchal, and Sagastizábal, 2002):

(a) <u>The Algorithm for Searching the *x* Space.</u> This manages the update of the argument *x* to the fastest convergence rate of the sequence $x_1, x_2,\ldots, x_k,\ldots, x^*$.

There are two types of search methods: one is a direct search, i.e. the traditional Nelder-Mead method (Fletcher, 2003); the other is to use information about the functions to be optimized, such as its higher derivatives. For example, the corresponding of the Newton method (relation (10)) for the multi-dimensional case is:

$$x_{k+1} = x_k - H_f^{-1}(s_k)\nabla f(x_k) \tag{11}$$

To determine the iteration step, must be solve the linear system: $H_f(x_k)s_k = -\nabla f(x)$ such that relation (11) becomes $x_{k+1} = x_k + s_k$.

(b) <u>The Simulator.</u> For each new iteration determined by the search algorithm, a computation is done using the new *x* in the problem *P* to be optimized to find a gradient descent or an equivalent "good" direction. Often the initial problem *P* is replaced by a simpler problem P_k which computes a direction along a line of the solution.

The method can be illustrated with subgradients on non-smooth functions. Let *f(x)* to be a non-smooth, convex function, defined on R, with values in R (Figure 3). Let x_0 be a point of discontinuity of the function. A subgradient is any line of direction *c* containing the point $(x_0, f(x_0))$, such that: $f(x) - f(x_0) \geq c(x - x_0)$, a line "below" a convex function and

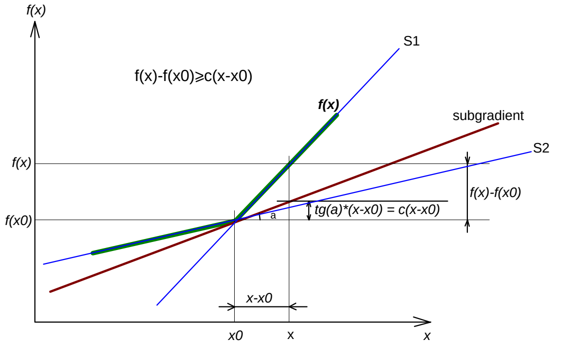


Figure 3: Subgradient in R^1.

which intersects the function in only one point.

For vectors in R^n a subgradient is a vector $s_f(x_0)$ such that:

$$f(x) - f(x_0) \geq s_f(x)^T(x - x_0) \qquad (12)$$

The set of all subgradients of f at x_0 is called the subdifferential of f at x_0 and it is denoted as $\partial f(x_0)$. In Figure 3 the subdifferential is the set of lines through the point (x_0, $f(x_0)$) placed between the lines S1 and S2 defined by the directions of the non-smooth function. Using the results obtained until now, the iterative algorithm for an agent i at iteration $k+1$ is:

$$x^i(k+1) = x^i(k) - \alpha^j(k)d_i(k) \qquad (13)$$

where $\alpha^j(k)$ is the size of the step used by agent i and $d_i(k)$ is a subgradient of agent's i objective function at $x^i(k)$.

4.2 Optimization of the Multi-agent System

A global optimization means that a vector x^* representing common resources to all agents, minimizes the sum of cost functions $f_i(x^*)$ of each agent i. Initially, before a new optimization cycle begins, each agent may have different values for its vector x. However, the size of the vector and the resource or physical meaning of each element in the vector is common to all agents.

An algorithm shown to converge for optimizing multi-agent systems (Nedic, Ozdaglar, 2009), uses the subgradient method shown in formula (13), but with the difference that at each step k, an agent combines its resource vector with a weighted value of the resource vectors of all other agents. For an agent i, its weight relative to an agent j at iteration k is $w_i^j(k) \in R$. Weights are normalized $\sum_{j=1}^{M} w_i^j(k) = 1$ for each agent i and each iteration k. With this change, formula (13) becomes:

$$x^i(k+1) = \sum_{j=1}^{M} w_i^j(k) \cdot x^j(k) - \alpha^i(k) \cdot d_i(k) \qquad (14)$$

The index j ranges over all M agents. As in relation (13), $\alpha^j(k)$ is a scalar being the step size used by agent i and $d_i(k)$ is a vector in R^n, the subgradient of agent's i objective function at $x^i(k)$.

While the weights in the work of Nedic and Ozdaglar (2009, p. 49) uses a concept of agents close to each other, in this work we use information about shared ontologies and *EfA* relations described in (7) and Figure 1.

4.3 Optimization in the Presence of Ontological Communication

The relative weight of agent i to agent $j, w_i^j(k)$, shows how "close" resources of the two agents are. Resources can be "close" even when their physical nature is very different.

Using ontologies, as shown in relations (3)-(7) and Figure 1, an agent can determine which resources are used as these are mapped to ontologies. As agents act continuously, the weights $w_i^j(k)$ change to reflect that some other set of resources are currently used.

The resource vector is a vector $x \in R^N$, so it contains as components only numerical, real values. In the chain of relations $RelS \rightarrow T\nu \rightarrow Z^L$, information from sensors are interpreting perception relations *RelS*, which are determining which symbols in $T\nu$ have the value True, which are determining which ontological concepts are identified. When an ontological concept is identified, it means all its properties are considered to hold.

It is important that at each new symbolic perception, the corresponding ontology is shared with other agents. Some of the previous resources will still be used while some new are going to be used. The choice of the weights $w_i^j(k)$ must reflect these transitions.

Thus, to find the weight factor among agents means determining a concept of distance among ontologies. One definition for distance uses a tree-ontology metrics: a distance using a formal representation of the ontology, by a measure of the path length between concepts. According to this measure, if c_1 and c_2 are two nodes in a formal representation of the ontology, l is the shortest path between the concepts and h is the depth of the deepest concept subsuming both concepts, then the distance between c_1 and c_2 is

$$sim(c1,c2) = e^{-k_1 l} \cdot \frac{e^{k_2 h} - e^{-k_2 h}}{e^{k_2 h} + e^{-k_2 h}} \qquad (15)$$

(k_1 and k_2 are scaling parameters for the shortest path vs. the depth) (Debenham, and Sierra, 2008). This type of measure can be done for asserted ontologies.

Another style of metrics is based on set-theoretic principles, by counting intersections and unions of ontological concepts. Best known is Tversky's similarity measure between two objects *a* and *b* and with properties (feature sets) A and B:

$$s(a,b) = \frac{|A \cap B|}{|A \cap B| + \alpha(a,b)\,|A - B| + (1-\alpha(a,b))\,|B - A|} \qquad (16)$$

where |.| is the cardinality of the set, minus is the set difference and *α(a,b)* in the interval [0,..,1] is a tuning factor that weights the contribution of the first reference model (Tversky's similarity measure is not symmetrical).

Besides Tversky's measure, similarity measure functions available in most scientific mathematical libraries are, for example Cosine, Dice, Euclidian, Manhattan or Tanimoto.

The algorithm for using any of the similarity measures above to determine the weight between two agents is the following:

1. Generate the set of all ontological properties (both numerical and non-numerical) of agent *i* as a union of all ontological properties valid at iteration *k*. Let this set be A and the cardinality of the set be |A|;
2. Generate in the same manner the set of all ontological properties for agent *j*; let this set be B with cardinality |B|;
3. Compute the cardinality of the intersection, union, differences, symmetric difference, as needed by the selected similarity formula;
4. Compute the similarity index. The resulting value is the weight $w_i^j(k)$.

5 EXAMPLE

To demonstrate the weighted method we present an example with a cooperation multi-robot system used in supermarket supervision. Two of the robots in the system have both common and distinct sensors. Robot 1, *Ro1*, has sensors (see relation (1)): Distance = *Se*(1,1); Shape (cube, cylinder, sphere) = *Se*(1,2); Dimensions (length, width, height, diameter) = *Se*(1,3); Temperature = *Se*(1,4); Colour = *Se*(1,5).

The second robot, *Ro2*, has sensors to obtain information for: Distance = *Se*(2,1); Shape = *Se*(2,2); Dimensions *Se*(2,3); Weight = *Se*(2,4).

Combining *Ro1* sensors information, we obtain possible perception relations (see relation (2)), e.g.: Re([Se(1,1), Se(1,2), Se(1,3)], 1) – for a combination of Distance, Shape and Dimension and Re([Se(1,1), Se(1,4), Se(1,5)], 5) – for a combination of Distance, Temperature and Colour.

These perception relations have associated symbolic perceptions (see relation (3)), as shown in Table 1.

Table 1: Symbolic perceptions *Ro*1.

Index Re	Perception Relation Re	Tv (symbol)
1	Distance < 10m AND Shape = Parallelepiped AND Dimension > 10m x 2m x 2m	shelf
2	Distance < 5m AND Shape = Parallelepiped OR Cube AND Dimension < 1m x 1m x 0.5m	box
3	Distance < 5m AND Shape = Sphere AND Dimension < 1m	ball
4	Distance < 5 m AND Shape = Sphere AND Dimension < 0.5m	balloon
5	Distance < 50m AND Temperature > 200^0C AND Colour = Red OR Orange	fire
6	Distance < 10m AND Temperature > 50^0C AND Colour = White OR Yellow	lamp

A similar computation for robot 2 gives the results in Table 2.

Table 2: Symbolic perceptions *Ro*2.

Index Re	Perception Relation Re	Tv (symbol)
1	Distance < 10m AND Shape = Parallelepiped AND Dimension > 10m x 2m x 2m	shelf
2	Distance < 5m AND Shape = Parallelepiped OR Cube AND Dimension < 1m x 1m x 0.5m AND Weight > 3 kg	box
3	Distance < 5m AND Shape = Parallelepiped AND Dimension < 1.5m x 0.5m x 1m AND Weight < 3 kg	cart

We have now two sets of symbols for the two robots (agents): P={*shelf, box, ball, balloon, fire, lamp*} and Q={*shelf, box, cart*}. We compute the set operations required e.g. using the "dice" similarity measure:

$$sim_{dice}(A,B)=\frac{2\cdot bothAB}{onlyA+onlyB+2\cdot bothAB}$$

|P ∪ Q| = |{shelf, box, ball, balloon, fire, lamp, cart}| = 7; |P-Q| = |{ball, balloon, fire, lamp}| = 4 and |Q-P| = |{cart}| = 1. So the weight w_1^2 between agent 1 and 2 is: $w_1^2=\frac{2\cdot 7}{4+1+2\cdot 7}=0{,}73\cdot$

The full matrix of weight is computed in this fashion for all agents.

The example is using three agents with three different cost functions of type $f(x)=Ax_1^2+Bx_2^2$, with parameters from Table 3.

Table 3: Agents cost functions.

Agent	A	B	Init vector
Agent 1	0.5	2.5	[1 3]
Agent 2	1.0	4.0	[3 6]
Agent 3	3.0	1.0	[4 5]

Figure 4 shows that indeed vectors converge towards the intended minimum value [0 0].

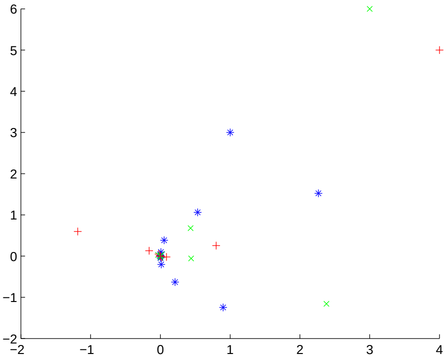


Figure 4: Convergence results.

The relation between the convergence precision and step size is shown in Figure 5.

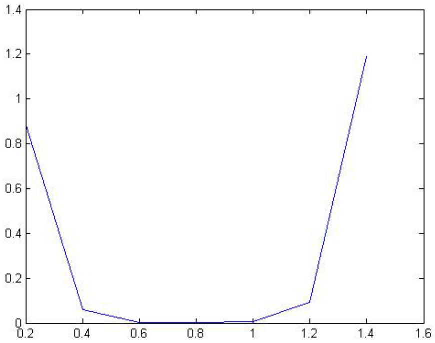


Figure 5: Convergence precision vs. step size.

An optimum step size is in the range 0.6 – 1; outside this range, the convergence precision decreases.

6 CONCLUSIONS

This paper capitalizes on recent results showing convergence of optimization in distributed agent applications. The optimization is based on establishing weights between agents, such that a weighted recursive computation shown in formula (14) gives a shared optimal resource allocation. This approach is appropriate for cooperative multi-robot system.

The paper's contribution is to demonstrate that a natural approach to compute a “distance” (the weight) among agents that exchange ontologies among them is the degree of shared ontologies the agents are using at the current computation step. This allows both numerical and symbolical ontological concepts to be used by set-theoretic similarity measures.

REFERENCES

Debenham, J, Sierra, C., 2008. Unifying trust, honour and reliability. In 2nd *IEEE Int'l Conf.on Digital Ecosystems and Technologies*, 2008, pp. 470-475.

Fletcher, R., 2003. *Practical Methods of Optimization*. Wiley. UK. 2nd edition.

Min, X., Luo, Z., Qianxing, X., 2009. Semantic Similarity between Concepts Based on OWL Ontologies. In *2nd Int'l Workshop on Knowledge Discovery and Data Mining*, 2009, Moscow, pp. 749-752.

Nedic, A., Ozdaglar, A., 2009. Distributed Subgradient Methods for Multi-Agent Optimization. In *IEEE Transactions On Automatic Control*, Vol. 54, No. 1, pp. 48-61.

Noy, N.F., McGuinness, D.L., 2001. Ontology Development 101: A Guide to Creating Your First Ontology. *Stanford Knowledge Systems Laboratory Technical Report KSL-01-05* and *Stanford Medical Informatics Technical Report SMI-2001-0880*.

Shor, N.Z., Kiwiel, K.C., Ruszcaynski, A., 1985. *Minimization Methods for Non-differentiable Functions*. Springer-Verlag. New York, USA.

Văcariu, L., Chintoanu, M., Lazea, G., Creţ, O., 2007. Communication at Ontological Level in Cooperative Mobile Robots System. In *Proceedings of the 4th International Conference on Informatics in Control, Automation and Robotics*, ICINCO 2007, May 9-12, 2007, Angers, France, pp. 455-460.

OFFROAD NAVIGATION USING ADAPTABLE MOTION PATTERNS

Frank Hoeller, Timo Röhling and Dirk Schulz
Fraunhofer Institute for Communication, Information Processing and Ergonomics FKIE, Germany
{*frank.hoeller, timo.roehling, dirk.schulz*}*@fkie.fraunhofer.de*

Keywords: Outdoor robotics, Local navigation, Motion patterns, Motion templates, Motion learning.

Abstract: This paper presents a navigation system which is able to steer an electronically controlled ground vehicle to given destinations considering all obstacles in its vicinity. The approach is designed for vehicles without a velocity controlled drivetrain and without an odometry system, making it especially useful for typical remote-controlled vehicles without upgrading the motor controllers. The vehicle is controlled by sets of commands, each set representing a specific maneuver. These sets are then combined in a tree-building procedure to form trajectories towards the given destination. While the sets of commands are executed the vehicle's movement is measured to refine the prediction used for path generation. This enables the approach to adapt to surface alterations. The technique requires a precise position estimation, which is provided in our implementation by a 3D laser mapping based relative localization system. We tested our approach using a 400kg EOD robot in an outdoor environment. The experiments confirmed that our navigation system is able to control the robot to its destination while avoiding obstacles and adapting to different ground surfaces.

1 INTRODUCTION

In the design of robot systems operating in unstructured outdoor environments, special care has to be taken that the robots do not accidentally collide with obstacles in their vicinity. Compared to indoor situations the robot can suffer drastically more damage by the more hazardous surrounding. Safe operation is commonly achieved by means of collision avoidance mechanisms which ensure a minimal distance to obstacles. This task is exacerbated by different ground surfaces which have a distinct effect on the wheel grip. This deviation has to be anticipated to ensure the reproducibility of planned motions and thus making collision avoidance possible.

Additional complications arise for robots which were designed for remote-control. Such robots are normally only equipped with relatively simple motor controllers. This has a significant impact on the techniques available for the collision avoidance because most classic navigation algorithms only generate velocity commands. To interpret such commands the controllers need to have an appropriate servo loop, which is not the case for such robots.

In this article we present an approach which allows a mobile robot with any kind of electronic motor controller to operate in cluttered outdoor environments. To be able to improve the navigation behavior of the robot in unknown locations, our approach follows a local navigation paradigm and does not need a map of the environment. Instead, the robot's motion control decides solely based on the robot's sensory input; in our current implementation this is a Velodyne 3D laser range scanner with 360 degree field of view which is used to compute a local 2.5D map. In addition, the laser scanner is used to exactly determine the robot position in a local frame of reference using a local mapping technique. This procedure also compensates the missing robot odometry.

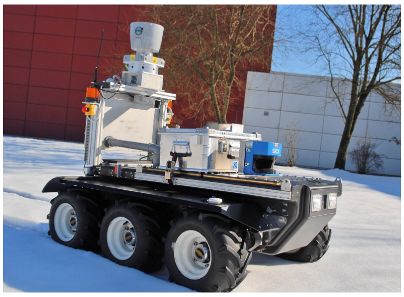


Figure 1: The Quinetiq Longcross robot Suworow equipped with a Velodyne 3D and an additional (unused) 2D laser range finder.

The motion planning for the robot is based on a tree-search technique which we developed to suit the special requirements of the robot's motor controllers. Our planning algorithm composes paths by combining predefined Motion Patterns. Each Motion Pattern consists of a set of robot commands and a series of poses which represent the robot's movement when the command set is executed by the controllers. With these Motion Patterns, the local navigation module repeatedly computes trees of collision-free command sequences. From each tree a path is extracted which brings the robot close to the destination coordinate as fast as possible. To tackle the surface traction problem, the robot's movements are measured on the fly. The collected data is used to update the measured movement part of the Motion Patterns. The upgraded Motion Patterns are handed over to the planning process and used for the tree generation.

The remainder of this article is organized as follows: After discussing related work in section 2, we introduce our Velocity Grid environment representation in section 3 followed by a description how they are utilized by our Motion Pattern based local navigation approach in Section 4. Section 5 then explains the movement measuring and Motion Pattern learning procedures. Before we conclude, we describe some experiments carried out with our robot to illustrate the capabilities and the robustness of our approach. We implemented our approach on a Quinetiq Longcross EOD robot (see Figure 1) and verified its feasibility in outdoor settings.

2 RELATED WORK

Waypoint navigation is one of the fundamental tasks for autonomous mobile robots. Many popular systems utilize the benefits of cars with ackermann steerings by first generating a feasable path and then try to follow this path as fast as possible. This can be done because, apart from physical limitations of the vehicle, the velocity does not affect the trajectory. For example, the DARPA Grand Challenge winning robot Stanley (Thrun et al., 2006) benefits from this principle. But because our robot does not have a velocity controlled drivetrain, the velocity cannot be controlled without affecting the trajectories.

Another approach (V. Hundelshausen et al., 2008) uses a different navigation technique by defining a limited set of commands or command-sequences and greedily decides in every computing interval which entity should be applied. Their path generator uses combinations of speeds and steering angles to generate trajectories which are then further evaluated. Therefor, an occupancy grid generated by a 3D-laser and different weighting functions are used. This is similar to the Dynamic Window Approach (DWA) (Fox et al., 1997; Brock and Khatib., 1999) because both methods restrict the search to a single time step, i.e. they select the next best controls based on the current sensor input and a model of the robot's dynamics.

In addition to these pure navigation algorithms, learning methods for robot motion have also been proposed in the past. Future robot positions can be estimated by regarding the current terrain type (Brunner et al., 2010). Therefor, intertial sensor data is processed with Gaussian process models to infer the movement velocities for position estimation and to deduce the terrain type from vibrations. Unfortunately, this method can only distinguish known surfaces and the used models require large amounts of computational power. Gaussian Processes in combination with reinforcement learning have been used to predict the movement of autonomous blimps (Ko et al., 2002), but just as the approach above it requires an extensive preparation phase. Seyr et al. use artificical neural networks to predict the trajectories of a two-wheeled robot (Seyr et al., 2005). It was shown that every trained situation can be recognized by their predictor. The used robot model is heavily bound to velocity parameters and thus again requires a suitable drivetrain.

The concept of motion template based learning has been previously employed to simplify the learning of complex motions (Neumann et al., 2009). In contrast to our approach the templates have parameters which are adjusted to fit the desired trajectory. This implies a feasible correlation between parameter input and drivetrain behavior.

3 VELOCITY GRIDS

In outdoor environments, a 3D-sensor (see Figure 1) is indispensable for an effective collision avoidance. Commonly used two-dimensional indoor variants are almost useless because of their insufficient environment coverage. However, the gain in coverage comes with the disadvantage of a drastically increased amount of data. Processing all this information while planning motions requires an unreasonable amount of computing power. Thus we simplify the 3D-sensor's data to a two-dimensional grid. In every grid cell the maximum admissible velocity of the corresponding area is stored. This type of map is referred to as Velocity Grid and it can be used for collision tests with less operating expense. This allows us to use modified two-dimensional planning algorithms. Although Velocity Grids seem to be similar to costmaps, they are utilized by our local navigation algorithm in a different way (see section 4).

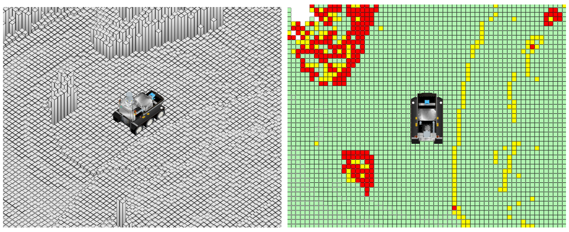


Figure 2: Left: A 2.5D height map acquired using a Velodyne. Right: The corresponding Velocity Grid. A car and the following person are marked as impassable (red), the sidewalk's curb is identified as slowly traversable (yellow).

To generate a Velocity Grid, the first step is to calculate a 2.5D map or height grid. A simple method to do this is to use a maximum-function on all measurements corresponding to one cell. Of course more sophisticated approaches could be used here, even including terrain classification (Vallespi and Stentz, 2008; Wellington et al., 2005), but all of these approaches require additional computation power and introduce additional delay. Since low latency and sufficient computing power are requirements for the following navigation approach, the simple mechanism was chosen over more intelligent algorithms. An example height grid is shown on the left in Figure 2. The information on the floor height now has to be transformed to a format which can easily be used for collision checks. For this purpose, we follow a line from the robot's center to every border cell of the Velocity Grid. Along this line the elevation changes are calculated, an appropriate maximum speed dependent on the robot's capabilities is chosen and assigned to the second of the two involved cells. This corresponds to extracting level curves in a star-like shape with the star's center in the center of the robot and categorizing the stepping of each curve into speed groups. Because of the Velocity Grid's geometry it is guaranteed that every cell is at least processed once. On the right of Figure 2 the outcome of this algorithm can be seen.

During the Velocity Grid calculation, it is assumed that the robot always moves in straight lines away from the center. For a robot that can turn on the spot this is a good estimate for the direct proximity, but a single Velocity Grid cannot be used for the planning and execution of a u-turn or similar. It has to be considered, that the Velocity Grid is continuously updated, several times per second, to ensure that new obstacle information reaches the motion planning in time. As soon as the planning module realizes that its previous path is invalid, it will generate a new trajectory with respect to the freshly detected obstacles.

4 USING MOTION PATTERNS FOR LOCAL NAVIGATION

4.1 Motion Patterns

The core of the overall approach is a local navigation planning component which directly controls the robot and steers it from its current position to a given destination on a collision-free path in configuration space. In our case a configuration $\mathbf{c}_t = (x,y,\theta,v)_t^\top$ of the robot at time t consists of the robot's position and heading $(x_t,y_t,\theta_t)^\top$ as well as its translational velocity v_t. In order to simplify the planning process, the rotational velocity is not considered here. Destination coordinates $\mathbf{d}$ of the local navigation are also defined as four dimensional vectors $\mathbf{d} = (x,y,\theta,e)^\top$, but instead of a velocity they contain a distance threshold e which defines a circle around the target coordinate. When the robot reaches this circle, the destination is considered reached, a technique which has also been used by Bruch et al. (Bruch et al., 2002). Since the robot is controlled by motor power commands instead of velocity commands, the outcome of a command depends on many factors and is far too expensive to compute.

Thus we introduce Motion Patterns to simplify the motion planning. The first component of each Motion Pattern is a series of robot control commands. These commands can be of any type; when used with a Longcross robot, they are motor power commands. This set of commands is static and not changed. The second component of a Motion Pattern is an array of oriented positions. It represents the trajectory on which the robot would probably move when the command series is sent to the robot. Together with the dimensions of the robot, the path the robot would take can be calculated and checked for collisions using the Velocity Grid described in section 3. This is a popular technique, because it "allows computing the cost of a motion without explicitly considering the motion itself" (Pivtoraiko et al., 2009). Notice that the number, shape, and complexity of Motion Patterns are not restricted, but definitely have an impact on the later described planning process. To combine Motion Patterns to a continuous path we have to make sure that the transitions between the chosen patterns are smooth. To accomplish this, the initial and final velocities are stored with every pattern. Furthermore, every Motion Pattern is assigned to a velocity group depending on its maximum speed. This property is used as criterion for exclusion.

4.2 Path Planning

Based on the model above, we can now build a collision-free tree of Motion Patterns $\mathcal{T} = (V,E)$ consisting of nodes V and possible transitions E between nodes. Every node V represents a Motion Pattern. The root node is defined by the final state of the currently applied Motion Pattern. New nodes are created in a breadth-first manner and are connected to their parent node if they meet the following three requirements:

1. The node's initial velocity matches the final velocity of its parent.
2. It is possible to apply the assigned Motion Pattern at the final position of the parent. In detail: the minimum speed of the cells, which the node traverses in the Velocity Grid, does not exceed the Motion Patterns maximum speed. This implies the absence of collisions.
3. The occurring roll and pitch angles are within the robot's safe operational parameters.

After adding a node, a weight-equivalent is assigned: Since the planning algorithm intends to find the fastest path to a given destination, distances between two configurations $\mathbf{c_i}$ and $\mathbf{c_j}$ are represented by the approximated travel time $h(\mathbf{c_i},\mathbf{c_j})$. This way it is possible to subsume the robot's current orientation, the spatial distance to, and the orientation of the destination into one scalar.

$$h(\mathbf{c_i},\mathbf{c_j}) = \frac{d(\mathbf{c_i},\mathbf{c_j})}{v_{\text{avg}}} + s\frac{(|\alpha(\mathbf{c_i},\mathbf{c_j})| + |\beta(\mathbf{c_i},\mathbf{c_j})|)}{\omega_{\text{avg}}} \quad (1)$$

Here $d(\mathbf{c_i},\mathbf{c_j})$ represents the line-of-sight distance to the target, v_{avg} denotes the robot's average translational speed, and ω_{avg} is the average rotational velocity. The angle $\alpha(\mathbf{c_i},\mathbf{c_j})$ describes the difference between the robot's heading in state $\mathbf{c_i}$ and the line-of-sight between $\mathbf{c_i}$ and $\mathbf{c_j}$. Similarly, $\beta(\mathbf{c_i},\mathbf{c_j})$ is defined as the difference between the heading of the target $\mathbf{c_j}$ and the line-of-sight. The idea behind this heuristic is to separate the motion from $\mathbf{c_i}$ to $\mathbf{c_j}$ into a rotation on the spot, followed by a straight line motion, followed again by a final rotation on the spot. Of course the robot generally translates and rotates simultaneously. The numerical constant s is introduced to account for the resulting speed advantage. If the destination's heading θ is undefined, β is set to zero. During the tree creation the node with the smallest h is marked. To ensure an effective movement, nodes which reach the destination threshold area should always be superior to other nodes, regardless of their h-value. Certainly the heuristic function does not guarantee that the destination pose is precisely reached but it always delivers a feasible path which at least moves the robot closer to the target area. Furthermore, the constant s defines how hard our approach tries to match the destination's orientation.

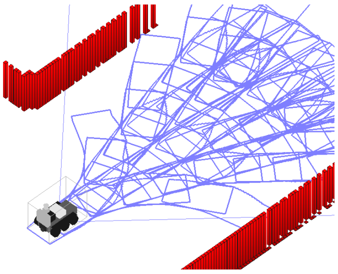


Figure 3: A tree of collision free paths that has been build using Motion Patterns in an 2-dimensional simulated environment. This tree would be used for planning.

The Velocity Grids are not used to sum up the costs of a potential path. Instead they are used as exclusion criterion for potentional Motion Patterns. The chosen Motion Patterns then define the cost.

The creation of new nodes is aborted when a sufficient tree depth or a time limit is reached. A potential new path is available after a tree is constructed. To be certain that a new path is available in time, the tree construction has to be finished before the Motion Pattern which is currently applied by the robot, is executed completely. To ensure this, the time limit for the tree construction process is equivalent to the time consumption of the quickest Motion Pattern available.

As mentioned before, the size of the Motion Pattern pool has a significant impact on the generated tree: While more available patterns enhance the quality of the resulting tree, they also decrease the tree depth that can be reached in the computation time window. A suitable quantity of Motion Patterns has to be choosen with respect to the capabilities of the used computer. An example tree from our current implementation can be seen in Figure 3. The used Motion Pattern pool consisted of five different basic maneuvers: accelerate, decelerate, turn left, turn right and move forward.

Using the previously marked node, a series of Motion Patterns representing a path towards the destination can be extracted from the tree.

5 MOTION LEARNING

A problem arising from our special kind of local navigation is its sensitivity to surface and traction changes. Motion Patterns are created for specific surfaces only. And it is unlikely that the surface or the surface's condition always remains constant, especially in outdoor

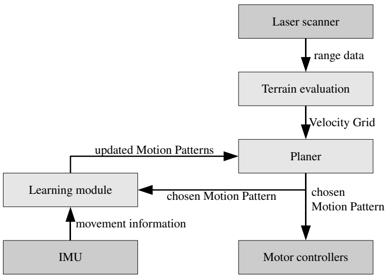


Figure 4: A flowchart showing the navigation process and the intregation of the learning mechanism.

scenarios. To compensate this, a basic learning mechanism has been added: While the command set of a Motion Pattern is executed, the robot's reactions are measured with the on-board IMU. Recorded are the relative x and y positions and the orientation. The measurements are integrated in the Motions Pattern's existing prediction using a component-by-component exponential smoothing function to allow continuous learning:

$$\bar{m}_{p,t} = (1-w)m_{p,t} + (w)\bar{m}_{p,t-1} \tag{2}$$

with $0 \leq w < 1$. Here $m_{p,t}$ is the measured trajectory of a Motion Pattern p at time t. $\bar{m}_{p,t-1}$ represents the existing prediction of the pattern and $\bar{m}_{p,t}$ the updated prediction. To limit the impact of new measurements, w should not be larger than $\frac{1}{2}$. Note that it is not possible to adapt the command sequence to match the desired trajectory, because the mapping from trajectories to commands is unknown and possibly not even computable. Figure 4 depicts how the learning mechanism is integrated in the navigation process.

A side effect of the Motion Pattern adaption is the possibility of pattern pool depletion: It occurs when an update causes one or more patterns to become best suited for a specific maneuver that was previously covered by another pattern. As the original Motion Pattern will not be used any more, it cannot be updated and remains unused. In order to counteract this effect, we delay pattern updates until new motion data for all Motion Patterns has been collected. Then, we update all patterns at once. The drawback of this method is that the trajectory construction will be less accurate due to outdated Motion Patterns.

Between the pattern updates some Motion Patterns will be executed more frequently than others. Therefore, we apply a secondary exponential smoothing with a much larger w which tracks the amount of change that is to be applied with the next update. This technique effectively balances the disproportionate impact that more frequently used Motion Patterns have on the pattern pool.

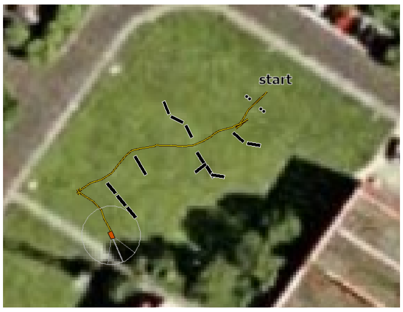


Figure 5: The obstacle course and the path autonomously driven by the robot.

6 EXPERIMENTAL RESULTS

6.1 Obstacle Course

To show the functionality of our navigation planning system, we set up an obstacle course with a length of about 50 m. Figure 5 shows the trajectory driven by the robot. The 50 m course was completed in 60 seconds. With an average Motion Pattern speed of 1.25 m/s this might sound surprising, but the robot executed a backward motion at the beginning and a turn-on-the-spot maneuver at the end of the course, which slowed it down. Notice that our system does not use global maps and thus was never meant to find optimal routes. To prevent such correction maneuvers, an additional global navigation is required.

6.2 Changing Surface

The second experiment demonstrates the adaption capabilities on a surface with varying characteristics. Therefor, a series of GPS-waypoints resulting in a total path lenght of approximately 260 m was given to the local navigation. The experiment was conducted with and without the learning mechanism. The average Motion Pattern speed was 1.25 m/s again.

To compare the performance, the absolute deviation from the predicted trajectory was measured for every executed Motion Pattern. The results are visualised using a rolling average on the left of Figure 6. The mean error without learning was 0.16 m and decreased by 18% (t-test significance >95.5%) with learning. Also, the total legth of the driven path was somewhat shorter. This seems to indicate a positive effect of the learning mechanism on the path planning. The two histrograms in Figure 6 show the error distributions for each setup. Both distributions are roughly normal and corroborate our findings.

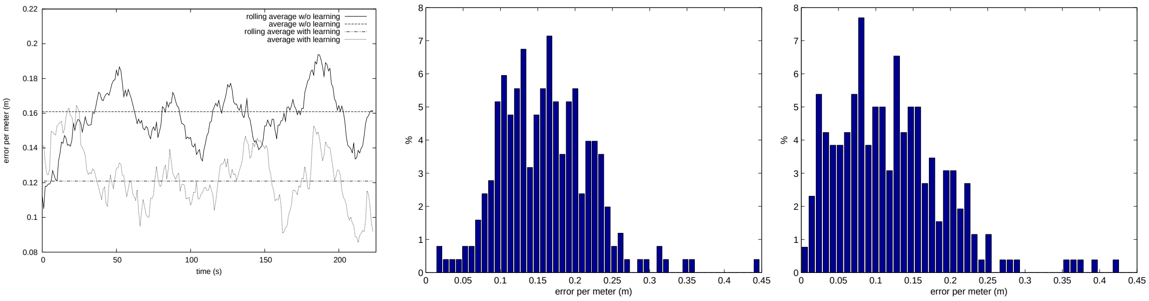


Figure 6: Experimental results: Left: The rolling and total average of the two setups. The learning variant outperforms the static version. Middle and right: Histogram showing the absolute error distribution without and with learning. Right: Large errors occur less frequent.

7 SUMMARY AND CONCLUSIONS

In this paper we presented a navigation system based on predefined motion templates which are combined with a tree-search technique to achieve efficient trajectories. We introduced Velocity Grids to represent difficult or impassable terrain by means of a maximum admissible velocity. The system has the ability to adjust the motion templates according to the actual robot movement, which is measured by an IMU, GPS, and lidar-based motion estimation. The soundness of our approach has been shown both in real-word navigation tasks. The system proved that it can navigate a robot through an obstacle course, and that it is able to adapt to different surfaces quickly.

Future work will focus on improving the performance of the motion learning and adapting mechanisms. The current countermeasure against pattern pool depletion causes temporarily outdated predictions. The most promising remedy would be to propagate the changes of one Motion Pattern to all others, provided that a sufficiently precise online approximation can be found. Another improvement worth investigating is the usage of the pattern deviation that is determined by the learning module as uncertainty measure in order to optimize the Motion Patterns' safety margins.

REFERENCES

Brock, O. and Khatib., O. (1999). High-speed navigation using the global dynamic window approach. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*.

Bruch, M., Gilbreath, G., and Muelhauser, J. (2002). Accurate waypoint navigation using non-differential gps. In *Proc. of AUVSI Unmanned Systems*.

Brunner, M., Schulz, D., and Cremers, A. B. (2010). Position estimation of mobile robots considering characteristic terrain model. In *submitted to the International Conference on Informatics in Control, Automation and Robotics (ICINCO)*.

Fox, D., Burgard, W., and Thrun, S. (1997). The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine*, 4(1):23–33.

Ko, J., Klein, D., Fox, D., and Haehnel, D. (2002). Vision-based Monte-Carlo self-localization for a mobile service robot acting as shopping assistant in a home store. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Neumann, G., Maass, W., and Peters, J. (2009). Learning complex motions by sequencing simpler motion templates. In *Proc. of the International Conference on Machine Learning (ICML)*.

Pivtoraiko, M., Knepper, R. A., and Kelly, A. (2009). Differentially constrained mobile robot motion planning in state lattices. *J. Field Robot.*, 26(3):308–333.

Seyr, M., Jakubek, S., Novak, G., and Dellaert, F. (2005). Neural network predictive trajectory tracking of an autonomous two-wheeled mobile robot. In *International Federation of Automatic Control (IFAC) World Congress*.

Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M., Hoffmann, G., Lau, K., Oakley, C., Palatucci, M., Pratt, V., Stang, P., and Strohband, S. (2006). Winning the darpa grand challenge. *Journal of Field Robotics*.

V. Hundelshausen, F., Himmelsbach, M., Hecker, F., Mueller, A., and Wuensche, H.-J. (2008). Driving with tentacles: Integral structures for sensing and motion. *J. Field Robot.*, 25(9):640–673.

Vallespi, C. and Stentz, A. (2008). Prior data and kernel conditional random fields for obstacle detection. In *Proceedings of the Robotics: Science and Systems (RSS)*.

Wellington, C., Courville, A., and Stentz, A. (2005). Interacting markov random fields for simultaneous terrain modeling and obstacle detection. In *Proceedings of the Robotics: Science and Systems (RSS)*.

EVALUATION OF FEEDBACK AND FEEDFORWARD LINEARIZATION STRATEGIES FOR AN ARTICULATED ROBOT

Roland Riepl, Hubert Gattringer and Hartmut Bremer
Institute for Robotics, Johannes Kepler University, Altenbergerstr. 69, 4040 Linz, Austria
{*roland.riepl, hubert.gattringer, hartmut.bremer*}*@jku.at*

Keywords: Robotics, Nonlinear control, Decentralized control, Linearization, Multi-body dynamics, Identification.

Abstract: The increasing demand of high performance applications in industrial environments calls for improved control strategies for nonlinear mechanical systems. Common flatness based approaches, effectively linearizing a highly nonlinear system, are available and ready for deployment. This contribution focuses on evaluating these strategies in a modern and widely used industrial setup.

1 INTRODUCTION

The objective to reduce cycle times while increasing the tool center point precision and dynamics, especially in robotic applications, requires sophisticated motion control.

Robots with a high number of degrees of freedom, for example articulated robots, are complex machines. Their system dynamics may be described by the equations of motion, a set of highly nonlinear differential equations. In contrary to the common approach, using Lagrange equations of second kind, this contribution utilizes the projection equation in subsystem representation, well described in (Bremer, 2008). Next to commonly used decentralized proportional and derivative (PD) control schemes, different model-based linearization techniques exist, as (Isidori, 1985; Slotine and Li, 1990; Fliess et al., 1995; Khalil and Dombre, 2004) suggest. These methods effectively linearize and decouple the equations of motion by using the inverse dynamic model in the feedback loop.

State of the art hardware with high processing power and low reaction times allows implementing these methods, not only under laboratory conditions, but also in industrial setups. With these tools at hand, the suggested linearization strategies are used to control a Stäubli RX130 industrial robot. Due to the fact that the author is unaware of any works which compare and investigate these approaches, this contribution tries to evaluate their performance. Focusing on measurements, the occurring lag errors with different control approaches are the basis for a final interpretation and conclusion.

2 SYSTEM DYNAMICS

In order to present detailed information on the multibody system under investigation,

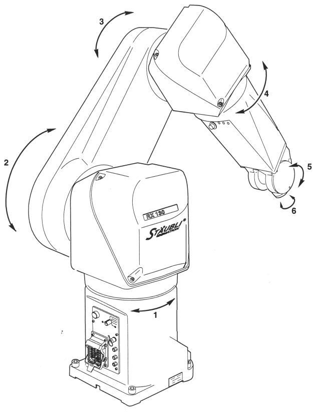


Figure 1: Sketch of the RX130L.

Figure 1 shows a sketch of the Stäubli industrial robot. It is a medium scale robot with six degrees of freedom, 1.6 m reach and a total mass of 235 kg.

Fundamentally, the basis for model based control approaches are the equations of motion

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{G}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{Q}(\mathbf{q},\dot{\mathbf{q}}) = \mathbf{Q}_m \in \mathbb{R}^n, \qquad (1)$$

where $\mathbf{q}$ denotes the vector of minimal coordinates, respectively the robots joint angles, $\mathbf{M}(\mathbf{q})$ is the configuration dependent, positive definite and symmetric mass matrix and $\mathbf{G}(\mathbf{q},\dot{\mathbf{q}})$ contains the velocity dependent nonlinearities as coriolis and centrifugal forces. The vector $\mathbf{Q}(\mathbf{q},\dot{\mathbf{q}})$ consists of all generalized forces, e.g. forces resulting from gravity or friction. The actuating motor torques are separated on the right side of the equations of motion and found in the vector $\mathbf{Q}_m$. The variable n represents the number of degrees of freedom which equals the number of joints, assuming a rigid multi body system.

The equations of motion, as stated in (1), are commonly derived using analytical methods, e.g. with the Lagrange equations. However, an articulated robot consists of several joints and may be regarded as an assembly of n arm/motor units.

Introducing such a unit, also called subsystem, as sketched in Figure 2,

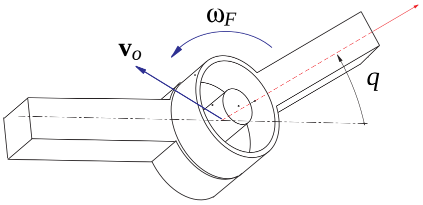


Figure 2: An arm/motor subsystem.

with the vector $\dot{\mathbf{y}}_k$

$$\dot{\mathbf{y}}_k = \begin{bmatrix} \mathbf{v}_0^T & \omega_F^T & q \end{bmatrix}_k^T \tag{2}$$

containing the translational and rotational velocities $\mathbf{v}_0$ and ω_F of the k-th subsystems reference frame and the corresponding joint angle q, leads with the projection equation

$$\sum_{k=1}^{N_{sub}} \left(\frac{\partial \dot{\mathbf{y}}_k}{\partial \dot{\mathbf{q}}}\right)^T [\mathbf{M}_k \ddot{\mathbf{y}}_k + \mathbf{G}_k \dot{\mathbf{y}}_k - \mathbf{Q}_k] = \mathbf{0} \tag{3}$$

in subsystem representation, see (Bremer, 2008), to the equations of motion. This synthetical method, based on projecting the linear and angular momentum in the free directions of movement, effectively makes use of the serial composition of several attached arm/motor units. The interested reader may find more information on the subsystems matrices and vectors, $\mathbf{M}_k$, $\mathbf{G}_k$ and $\mathbf{Q}_k$ and their detailed derivation in (Gattringer, 2006).

2.1 Model Verification

After an identification process as suggested in (Sciavicco and Siciliano, 2004; Khalil and Dombre, 2004), the model parameters are verified by comparing measured and simulated motor torques. The simulation is carried out by integrating the equations of motion. The reference trajectory for verification purposes strictly differs from the ones used during the identification process to assure the parameters' correctness in the whole task space.

Figure 3 presents the accordance of the first three joints. Joints four to six are of equivalent quality.

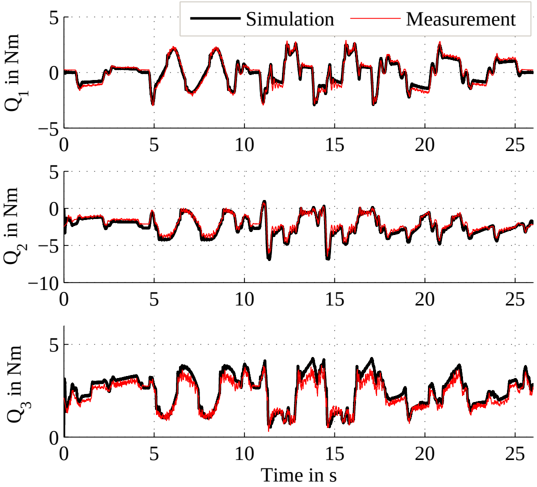


Figure 3: Measured versus simulated motor torques.

3 MOTION CONTROL

This section summarizes the control strategies for tracking the tool center point along a desired trajectory. All introduced methods are part of the evaluation process.

3.1 Decoupled PD-control

As the name suggests, the decoupled PD-control is a decentralized single joint control law. Each single axis compensates the nonlinearities, e.g. influences from the other joints, while tracking the desired reference trajectory in joint space. The control law is given by

$$\mathbf{Q}_m = \mathbf{K}_P(\mathbf{q}_d - \mathbf{q}) + \mathbf{K}_D(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}), \tag{4}$$

where the lower right index d denotes the desired reference values for joint positions and velocities. The positive definite diagonal matrices $\mathbf{K}_P$ and $\mathbf{K}_D$ are the proportional and derivative gains.

The major disadvantage of this common and simple control law is the lack of knowledge about the system which is to be controlled.

Asymptotic stability can only be proven if the system remains in steady state with the classical stability theorems of mechanical systems, summarized in (Bremer, 1988).

3.2 Decoupled PD-control with Feedforward

To negate the major drawback of the decoupled PD-control an additional feedforward term is inserted into control law (4), yielding

$$\mathbf{Q}_m = \mathbf{K}_P(\mathbf{q}_d - \mathbf{q}) + \mathbf{K}_D(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) + \mathbf{u}_{FF}. \qquad (5)$$

Effectively exploiting the system knowledge, obtained through the dynamic modeling and identification, the necessary reference motor torques which guide the tool center point along a desired trajectory, are given with the equations of motion (1) and are so intuitively suitable for a feedforward control

$$\mathbf{u}_{FF} = \mathbf{M}(\mathbf{q}_d)\ddot{\mathbf{q}}_d + \mathbf{G}(\mathbf{q}_d, \dot{\mathbf{q}}_d)\dot{\mathbf{q}}_d + \mathbf{Q}(\mathbf{q}_d, \dot{\mathbf{q}}_d). \qquad (6)$$

Obviously, if the reference trajectory is two times continuously differentiable with respect to time, the resulting feedforward torques will show a continuous progression. This set of feedforward control is also called exact feedforward linearization in literature, see (Hagenmeyer and Delaleau, 2003).

With this choice the PD-controller's task is reduced to compensate modeling inaccuracies, parameter variations and external or unknown disturbances.

Proving asymptotic stability, however, still leads to further challenges, because inserting the feedforward term and evaluating the equations of motion transforms the dynamics to a nonlinear time-variant system. The interested reader may find more information in (Kugi, 2008; Hagenmeyer and Delaleau, 2003).

3.3 Computed Torque

Introducing the computed torque control law

$$\mathbf{Q}_m = \mathbf{M}(\mathbf{q})\dot{\mathbf{v}} + \mathbf{G}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{Q}(\mathbf{q}, \dot{\mathbf{q}}) \qquad (7)$$

and inserting it into the equations of motion (1) yields the system

$$\ddot{\mathbf{q}} = \mathbf{v} \qquad (8)$$

of n double integrators, resulting in a solely linear interrelation between the new system input $\mathbf{v}$ and the joint angles $\mathbf{q}$. A possible choice for the new system input $\mathbf{v}$ is

$$\mathbf{v} = \ddot{\mathbf{q}}_d + \mathbf{K}_D(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) + \mathbf{K}_P(\mathbf{q}_d - \mathbf{q}) \qquad (9)$$

with the positive definite diagonal matrices $\mathbf{K}_P$ and $\mathbf{K}_D$ as stabilizing gains.

By defining the joint tracking error $\mathbf{e} = \mathbf{q}_d - \mathbf{q}$ and inserting Equation (9) in (8) the system dynamics of the tracking error $\mathbf{e}$ for the closed loop system

$$\begin{aligned} \ddot{\mathbf{q}}_d - \ddot{\mathbf{q}} + \mathbf{K}_D(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) + \mathbf{K}_P(\mathbf{q}_d - \mathbf{q}) &= \mathbf{0} \\ \ddot{\mathbf{e}} + \mathbf{K}_D\,\dot{\mathbf{e}} + \mathbf{K}_P\,\mathbf{e} &= \mathbf{0} \quad (10) \end{aligned}$$

is found. Equation (10) instantly allows to shape the error dynamics by choosing the gains $\mathbf{K}_P$ and $\mathbf{K}_D$, for example by pole placement.

Please note, in literature the computed torque method is also referred to as inverse dynamics, feedback linearization or flatness based control, as (Isidori, 1985; Slotine and Li, 1990; Fliess et al., 1995) describe.

3.4 Extended Linearization

Similar to Subsection 3.3, the nonlinearities in the equations of motion are eliminated with the control law

$$\begin{aligned} \mathbf{Q}_m = {} & \mathbf{M}(\mathbf{q})(\dot{\mathbf{v}} + \mathbf{K}_0\,\mathbf{q} + \mathbf{K}_1\,\dot{\mathbf{q}}) \\ & + \mathbf{G}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{Q}(\mathbf{q}, \dot{\mathbf{q}}). \qquad (11) \end{aligned}$$

Inserting the control law (11) into the equations of motion (1) yields

$$\ddot{\mathbf{q}} = \mathbf{v} - \mathbf{K}_0\,\mathbf{q} - \mathbf{K}_1\,\dot{\mathbf{q}} \qquad (12)$$

or equivalent in state space representation

$$\frac{d}{dt}\begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{E} \\ -\mathbf{K}_0 & -\mathbf{K}_1 \end{bmatrix}\begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{bmatrix} + \begin{bmatrix} \mathbf{0}, \\ \mathbf{E} \end{bmatrix}\mathbf{v}, \qquad (13)$$

which describes the dynamics of the closed loop system. Apparently, this MIMO-system is controllable with the whole repertory of tools available from linear system theory, for example linear quadratic regulators or pole placement, well summarized in (Chen, 1998).

Certainly, before designing a linear controller for system (13), the matrices $\mathbf{K}_0$ and $\mathbf{K}_1$ need to be chosen. It is suggested to pick them in a manner so that the closed loop system (13) resembles its physical counterpart. One suggestion is that the eigenvalues of the linearized equations of motion may be used as guideline for picking the eigenvalues of the closed loop system. However, due to stability issues positive eigenvalues in the closed loop system (13) need to be avoided.

Asymptotic stability of Equation (13) is derived with linear system theory.

3.5 Friction Feedback Issues

One common problem with feedback linearization methods occurs when friction models are part of the inverse dynamics. This is usually the case for any mechanical or robotic application. If the classical model for friction

$$Q_{fric} = d_v\,\dot{q} + d_c\text{signum}\,(\dot{q}) \qquad (14)$$

is part of the feedback loop, then the noisy measurement of the velocity signal $\dot{q}$ will be amplified and fed

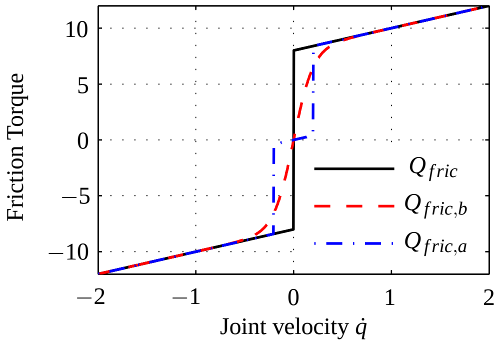


Figure 4: Friction models for feedback loop.

through to the actuators, directly resulting in a disturbing hum.

To counter this effect the friction characteristics in the feedback loop have to be adapted. Two simple solutions are given by

$$\begin{aligned} Q_{fric,a} &= d_v\,\dot{q} + d_c \text{ signum }(\dot{q}) && \text{for } \dot{q} \geq \varepsilon_1 \\ Q_{fric,a} &= 0 && \text{else} \\ &\text{or} \\ Q_{fric,b} &= d_v\,\dot{q} + d_c \tanh\,(\dot{q}/\varepsilon_2) \end{aligned} \tag{15}$$

with suitable values for ε_1 and ε_2. Figure 4 shows the original and adapted friction models.

4 EXPERIMENTS

The introduced control strategies are evaluated in a series of experiments. Additionally, details on the laboratory setup and the recorded data are the dominating topics of this section.

4.1 Setup

In the laboratory, the Stäubli RX130 industrial robot is interfaced with state-of-the-art motion hardware. All mechanical and electrical parts of the robot itself, like motors and resolvers, remain untouched. The servo drives, powering the synchronous motors, and the digital processing unit are of industrial standard but still offer the necessary development tools for implementing the proposed methods.

Figure 5 shows the main components and their interrelation to each other. The central computing unit is connected to six servo drives via a high-speed Powerlink bus. Each servo drive acts as an amplifier for the AC motor, evaluates resolver signals and may use an internal linear cascaded controller for position and velocity control of the attached motors.

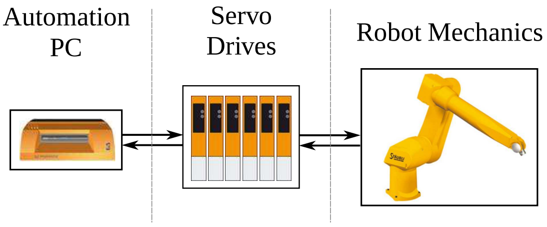


Figure 5: Setup.

4.2 Trajectory

A reference trajectory with significant characteristics is necessary for the process of evaluation. It needs to cover a reasonable workspace while containing high dynamics, even in disadvantageous poses and configurations.

A suitable choice, meeting this prerequisites, is found in (ISO NORM 9283, 1998) which originally defines a standardized trajectory for tool center point measurements and evaluations. By sharing a common objective, the proposed trajectory is also the reference in the following experiments.

The velocity and acceleration of the tool center point is given with 1 m/s and 5 m/s^2. These settings represent the highest values without exceeding the available actuating torques. The trajectory itself is a composition of straight lines, circles and squares in space.

The inverse kinematics is computed numerically, yielding the according joint values and their derivatives with respect to time. It is also guaranteed that all reference joint angles are two times continuously differentiable with respect to time.

4.3 Centralized / Decentralized Control

When realizing the introduced control structures on the industrial hardware, the feedforward and feedback approach show major differences. Basically, the model based control methods – feedforward as well as feedback – evaluate the equations of motion each sample step in order to linearize the mechanical system. Furthermore some stabilizing gains guarantee stability.

However, in the case of feedforward control, which is a decoupled and decentralized method, the stabilizing PD feedback loop may be implemented directly on the servo drives which profit from very short response times and a fast sampling rate. On the other hand, linearizing the system only by using reference values instead of actual ones, is not as exact as with feedback linearization.

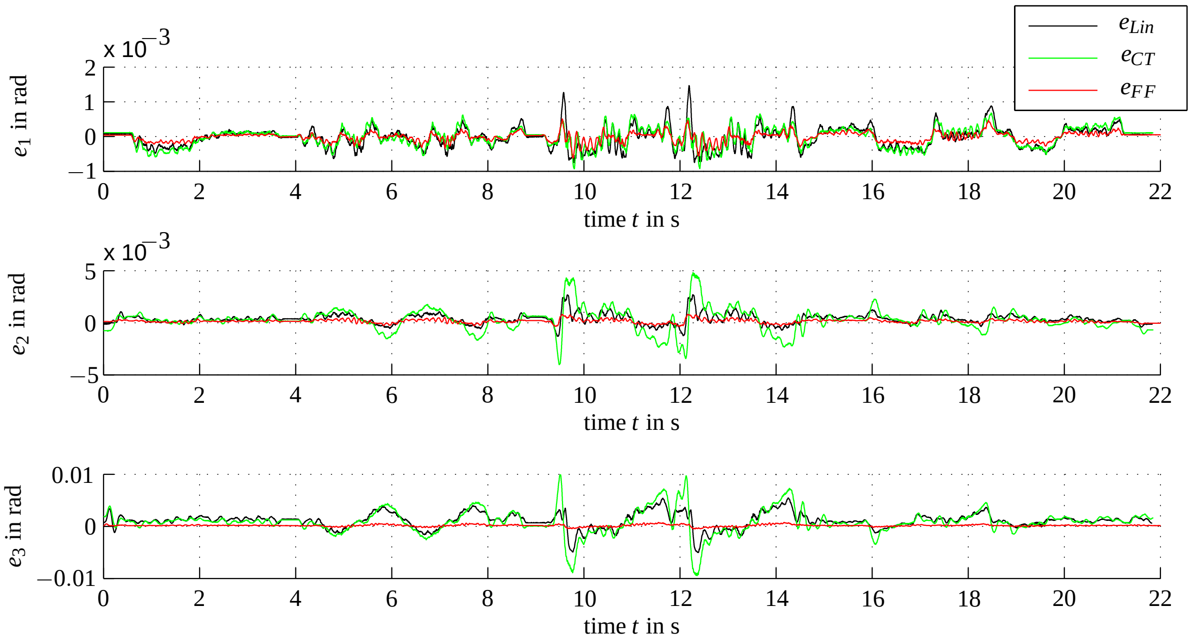


Figure 6: Lag errors with various control strategies.

4.4 Results

During a series of experiments the lag errors of the joints are recorded. Due to the rigid construction and the high stiffness of the gears, the joints' lag errors may be used to give estimations for the tool center accuracy. Also, all design parameters are chosen with highest possible gains which were evaluated in another set of preceding investigations.

In Figure 6, the lag errors for joint one, two and three are revealed. The lag errors resulting with extended linearization are denoted with e_{Lin}, respectively the ones with computed torque and feedforward with e_{CT} and e_{FF}.

While the performance of the control methods seems to be nearly equal for joint one, the joints two and three show different results. Especially axes with high loads resulting from gravity are benefiting from their reduced lag errors with the feedforward approach.

The axes four, five and six are of similar behavior and thus omitted. All values presented are joint angles, transformed to the gears' output sides.

5 INTERPRETATION

It is clearly not immediately evident that the feedforward linearization excels in performance in case of this robotic application. The reason lies in the structure of control. If the feedforward linearization is used in combination with the decoupled PD - control, the servo drives assume the task of decentralized controllers. These decentralized controllers profit from very short cycle times, still supported by the feedforward variables.

The centralized control methods, which linearize the system by feedback, are mathematically outstanding and superior. However, due to the delays which arise from closing the feedback loop to the central processing unit, the maximum gains are drastically reduced. Thus, deviations from the model and external disturbances are eliminated in a slower manner.

The effort for preparation – originating from obtaining the dynamic model with a valid and well identified set of parameters – is equal for all proposed model-based control strategies because they share a common origin, the equations of motion.

To sum up, the method of choice to control the motion of an articulated robot with six axes is the feedforward linearization approach. Model based system knowledge can effectively be exploited to support the servo drives internal, fast sampling controllers which are technically mature and professional products.

6 CONCLUSIONS

With the interpretation at hand, the feed forward linearization technique is investigated in more detail. From a practical viewpoint, the endeffector error along a given trajectory is a characteristic of interest. Assuming a rigid model and high gear stiffnesses, the forward kinematics is evaluated to compute the deviation between the desired trajectory and the actual trajectory.

To analyze the effects of the feedforward linearization, the experiment is conducted with and without superimposed feed forward loop. Figure 7 shows the tool center point (short TCP) error in inertial coordinates. The x- and y-axes are parallel to the robots mounting surface whereas the z-axis points in the same direction as the gravity vector.

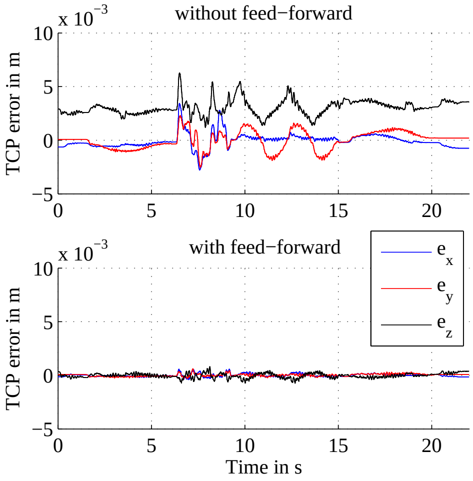


Figure 7: Evolution of TCP errors.

Interpreting the plot shows, that the robots manipulator has nearly an average of 4 mm deviation in direction of the z-axis. This static payload of the robot's own mass is instantly compensated with the feed forward approach. During the phases with high dynamics, the result with feed-forward control is also considerably improved.

ACKNOWLEDGEMENTS

The authors acknowledge the support and cooperation with Bernecker & Rainer, especially the group of Alois Holzleitner and Gernot Bachler. The motion control system provided allows quick implementation and offers technical perfection which made this research possible.

REFERENCES

Bremer, H. (1988). *Dynamik und Regelung mechanischer Systeme*. Teubner Studienbücher, Stuttgart.

Bremer, H. (2008). *Elastic Multibody Dynamics: A Direct Ritz Approach*, volume 35 of *Intelligent Systems, Control, and Automation: Science and Engineering*. Springer-Verlag GmbH.

Chen, C.-T. (1998). *Linear System Theory and Design (Oxford Series in Electrical and Computer Engineering)*. Oxford University Press, third. edition.

Fliess, M., Lévine, J., and Rouchon, P. (1995). Flatness and defect of nonlinear systems: Introductory theory and examples. *International Journal of Control*, 61:1327–1361.

Gattringer, H. (2006). *Realisierung, Modellbildung und Regelung einer zweibeinigen Laufmaschine*. PhD thesis, Johannes Kepler Universität Linz.

Hagenmeyer, V. and Delaleau, E. (2003). Robustness analysis of exact feedforward linearization based on differential flatness. *Automatica*, 39(11):1941 – 1946.

Isidori, A. (1985). *Nonlinear Control Systems*. Springer-Verlag.

ISO NORM 9283 (1998). Manipulating industrial robots - performance and criteria. Norm, EN ISO 9283.

Khalil, W. and Dombre, E. (2004). *Modeling, Identification and Control of Robots*. Kogan Page Science, London.

Kugi, A. (2008). Introduction to tracking control of finite- and infinite-dimensional systems. *Stability. Identification and Control in Nonlinear Structural Dynamics (SICON)*.

Sciavicco, L. and Siciliano, B. (2004). *Modelling and Control of Robot Manipulators*. Springer, United Kingdom.

Slotine, J.-J. and Li, W. (1990). *Applied Nonlinear Control*. Prentice Hall.

EXPERIMENTS WITH A CONTINUUM ROBOT STRUCTURE

Dorian Cojocaru, Sorin Dumitru, Florin Manta, Giuseppe Boccolato
Faculty of Automation, Computers and Electronic, University of Craiova, Bld. Decebal, Craiova, Romania
cojocaru@robotics.ucv.ro

Ion Manea
Faculty of Mechanics, University of Craiova, Calea Bucuresti Str., Craiova, Romania
Manea53@yahoo.co.uk

Keywords: Continuum Robots, Kinematics, Mechanical Experiments.

Abstract: A continuum arm prototype was designed and implemented using two different shapes: cylinder like and cone like. This new robot is actuated by stepper motors. The rotation of these motors rotates the cables which by correlated screwing and unscrewing of their ends determine their shortening or prolonging, and by consequence, the tentacle curvature. The kinematics and dynamics models, as well as the different control methods developed by the research group were tested on these robots. The lack of no discrete joints is a serious and difficult issue in the determination of the robot's shape. A solution for this problem is the vision based control of the robot, kinematics and dynamics. An image – based visual servo control where the error control signal is defined directly in terms of image feature parameters was designed and implemented. In this paper, we analyze some mechanical capabilities of this type of continuous robots. We present the structure of the tentacle robot, it's kinematics model and the results for a series of tests regarding the mechanical behaviour of the robotic structure.

1 INTRODUCTION

Modern industrial robots are mostly (human) arm-inspired mechanisms with serially arranged discrete links. When it comes to industrial environment where the workspace is structured and predefined this kind of structure is fine. This type of robots are placed in carefully controlled environments and kept away from human and their world.

When it comes to robots that must interact with the natural world, it needs to be able to solve the same problems that animals do. The rigid structure of traditional robots limit their ability to maneuver and in small spaces and congested environments, and to adapt to variations in their environmental contact conditions (Suzumori et al., 1991), (Robinson and Davies, 1999). For improving the adaptability and versatility of robots, recently there has been interest and research in "soft" robots (Cowan and Walker, 2008). In particular, several research groups are investigating robots based on continuous body "continuum" structure. If a robot's body is soft and/or continuously bendable it might emulate a snake or an eel with an undulating locomotion (Crespi and Ijspeert, 2006).

The continuum or hyper-redundant robot manipulators behaviour is similar to biological trunks, tentacles or snakes. The movement of the continuum robot mechanisms is generated by ending continuously along their length to produce a sequence of smooth curves (Blessing and Walker, 2004). This contrasts with discrete robot devices, which generate movement at independent joints separated by supporting links.

We can also describe as continuum robots snake-like robots and elephant's trunk robots, although these descriptions are restrictive in their definitions and cannot be applied to all snake-arm robots. A continuum robot is a continuously curving manipulator, much like the arm of an octopus (Davies, J.B.C., 1998). An elephant's trunk robot is a good descriptor of a continuum robot. The elephant's trunk robot has been generally associated with an arm manipulation – an entire arm used to grasp and manipulate objects, the same way that an elephant would pick up an apple. Snake-arm robots are often used in association with another device meant to introduce the snake-arm into the confined space. However, the development of high-performance control algorithms for these manipulators is quite a challenge, due to their unique design and the high degree of uncertainty in their dynamic models. The great number of parameters, theoretically an infinite one, makes very difficult the use of classical control and the conventional transducers for position and orientation.

2 THE ROBOTIC STRUCTURE

A research group from the Faculty of Automation, Computers and Electronics, University of Craiova, Romania, started working in research field of hyper redundant robots over 20 years ago. The experiments started on a family of TEROB robots which used cables and DC motors. The kinematics and dynamics models, as well as the different control methods developed by the research group were tested on these robots. Since 2008, the research group designed a new experimental platform for hyper redundant robots.

Our robotic system is composed from two units, one with a flexible structure with kinematic possibilities similar with the snake's locomotion and another one with driving. The flexible unit is composed from three modules with independent driving, that confers a complex 3D configuration, with multiple kinematic possibilities for the working space. The flexible structure is conceived in modular systems with decoupling possibilities for a controlled optimization of the working space.

The flexible structure as an integrate system, or as independent modules, is conceived to allow driving in two modes, respective:

- one with wires and a flexible central column;

- one with flexible vertebrates and a flexible central column;

In the case where the driving is made by wires, a module has two freedom degrees and in the case of driving with flexible columns each module has three degrees of freedom. As it can be observed the flexible unit structured in three modules for controlling a complex workspace, with multiple kinematic possibilities. The machine structure for each module is based upon thread transmissions with self decelerations possibilities and adjust of the axial-radial clearances.The flexible unit with the snake-like design is composed from a base flange, some intermediary flanges, and four flexible shafts, with high elasticity, which will be called vertebral spines. The central shaft is mounted rigidly to all the intermediary flanges, fig. 1.

The three super elastic spines are mounted equidistantly upon the central spine. The entire vertebrates are connected only to the end flange. The intermediary flanges maintain constant the radial distance between the secondary tubes and the central vertebrate. Changing in an active way the length for two of the vertebrate spines, the final flange can be manipulated with two degrees of freedom in any direction.

The three actuating spines are rigidly jointed only to the end flange, the joint between them and the intermediary flanges is like one translational joint.

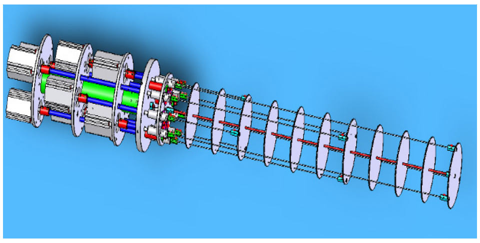


Figure 1: The virtual model with the driving and vertebral unit.

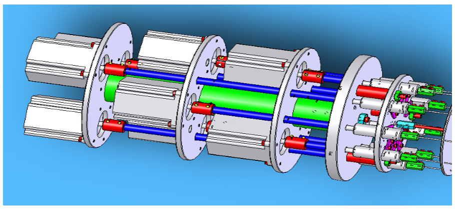


Figure 2: Redundant driving unit.

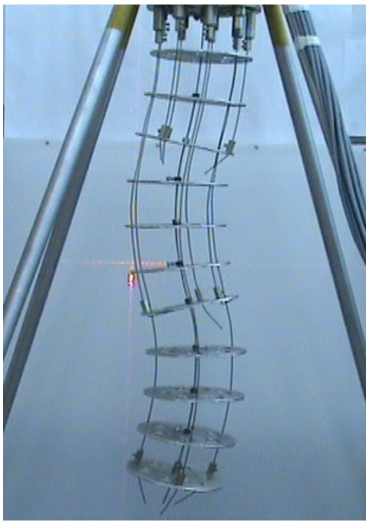


Figure 3: The assembly of the robotic system with the baggy vertebral unit.

2.1 Experimental Modelling

As it was presented above, the robot is designed in a modular structure, with three modules, independently actuated through thread transmissions with possibility to adjust the radial and axial clearances, in the aim to assure the imposed kinematics precision. The vertebrate unit is designed to work in two ways, respectively with two or three degrees of freedom on each module.

2.2 Modelling with Finite Elements

For demonstrating the viability of mechanical system it is proposed the modelling and simulation functionality with finite element method. Faithfully are respected the shape conditions and loadings, so that through a Virtual Prototyping attempt it is obtained a parameterized virtual system, which can be loaded so that you get various types of deformed shapes, obviously controlled for the analyzed mechanic system (Dumitru et al., 2009).

Parameterized modelling of the flexible system allows 3D simulation for different variants with one, two or three modules, with different size dimensions and types of materials with circular vertebras full or tubular, in this way assuring a wider area of skills for the proposed and analyzed system.

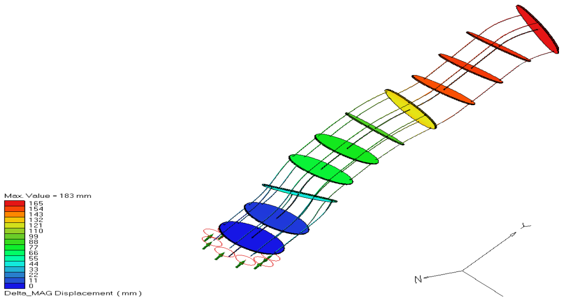


Figure 4: Elastic displacement distribution for a kinematic chain with 3 modules.

3 THE KINEMATIC MODEL OF THE TENTACLE ARM

In order to control a hyper-redundant robot we have to develop a method to compute the positions for each one of his segments (Ivanescu et al., 2006). By consequence, given a desired curvature S*(x, tf) as sequence of semi circles, identify how to move the structure, to obtain s(x, t) such that:

$$\lim_{t \to t_f} s(x,t) = S^*(x,t_f) \qquad (1)$$

where x is the column vector of the shape description and tf is the final time (see Fig. 6).

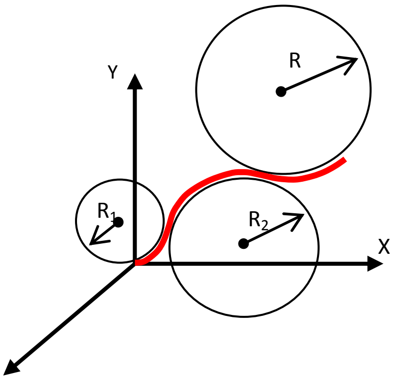


Figure 5: The description of the desired shape.

3.1 Tentacle Shape Description

The tentacle's shape will be described considering two angles (θ, α) for each segment, where θ is the rotation angle around Z-axis and is the rotation angle around the Y-axis (see Figure 5). In order to describe the movement we can use the roto-translation matrix as shown in Figure 6. Figure 7 shows the relation between the orientation of one projection on the main plane segment and the curvature angle of the previous segment.

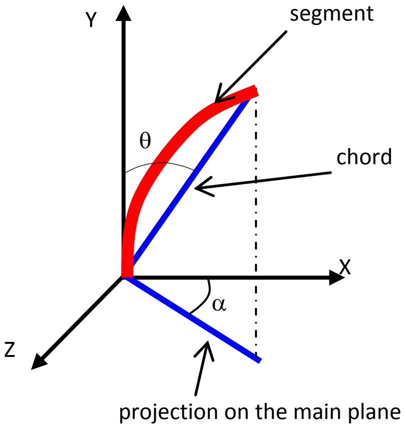


Figure 6: The description of a 3D curvature.

$$\begin{bmatrix} cos(2\cdot\beta) & sin(2\cdot\beta) & L\cdot sin(\beta) \\ -sin(2\cdot\beta) & cos(2\cdot\beta) & L\cdot cos(\beta) \\ 0 & 0 & 1 \end{bmatrix} \qquad (2)$$

The generic matrix in 2D that expresses the coordinate of the next segment related to the previous reference system can be written as follow:

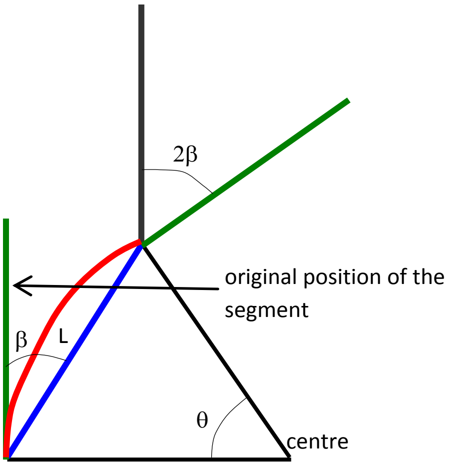


Figure 7: Curvature and relation between θ and β.

In 3D space we cannot write immediately the dependence that exists between segments. This relation can be obtained through the pre-multiplication of generic roto-translation matrix (Cojocaru et al., 2008). One of the possible combinations to express the coordinate of the next segment related to the frame coordinate of the previous segment is the following:

$$R^i{}_{generic} := R^i_z(\theta^i)\cdot Tr_y(V^i)\cdot R^i_y(\alpha^i)\cdot R^i_z(\theta^i) \tag{3}$$

where $R^i_z(\theta^i)$ and $R^i_y(\alpha^i)$ are the fundamental roto-translation matrix having 4x4 elements in 3-D space, and Try(V) is a 4x4 elements matrix of pure translation in 3-D space and where Vi is the vector describing the translation between two segments expressed in coordinate of i-th reference system. The main problem remains to obtain an imposed shape for the tentacle arm. In order to control the robot, we need to obtain the relation between the position of the wires and the position of the segment.

3.2 Curvature of One Segment

In the current stage of our research, a decoupled approach is used for the robot control scheme, thus the three segments are controlled separately, without considering the interaction between them. This section presents the way direct kinematics of one segment was obtained. The geometry of one segment for the 2D case is described in Fig. 10. The curvature angle θ of the segment is considered as the input parameter, while the lengths L1 and L2 of the control wires are the outputs (Boccolato et all., 2009).

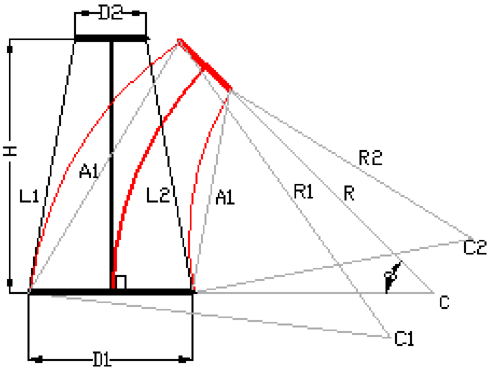


Figure 8: The geometry of one segment.

The radius R of the segment curvature is obtained using equation (4):

$$R = \frac{H}{\theta} \tag{4}$$

where H is the height of the segment. The following lengths are obtained from Fig. 10, based on the segment curvature:

$$\begin{aligned} L_{11} &= R + \frac{D_1}{2} & L_{12} &= R + \frac{D_2}{2} \\ L_{21} &= R - \frac{D_1}{2} & L_{22} &= R - \frac{D_2}{2} \end{aligned} \tag{5}$$

where D1 and D2 are the diameters of the segment end discs.

Based on the Carnot theorem, the lengths A1 and A2 are then obtained:

$$\begin{aligned} A_1 &= \sqrt{L_{11}^2 + L_{12}^2 - 2\cdot L_{11}^2 \cdot L_{12}^2 \cdot \cos\theta} \\ A_2 &= \sqrt{L_{21}^2 + L_{22}^2 - 2\cdot L_{21}^2 \cdot L_{22}^2 \cdot \cos\theta} \end{aligned} \tag{6}$$

The control wires curvature radius R1 and R2 are given by the relations (7):

$$R_1 = \frac{A_1}{2\cdot \sin\frac{\theta}{2}} \qquad R_2 = \frac{A_2}{2\cdot \sin\frac{\theta}{2}} \tag{7}$$

Finally, the lengths of the control wires are obtained as in (8):

$$\begin{aligned} L_{w1} &= R_1 \cdot \theta = \frac{A_1 \cdot \theta}{2\cdot \sin\frac{\theta}{2}} \\ L_{w2} &= R_2 \cdot \theta = \frac{A_2 \cdot \theta}{2\cdot \sin\frac{\theta}{2}} \end{aligned} \tag{8}$$

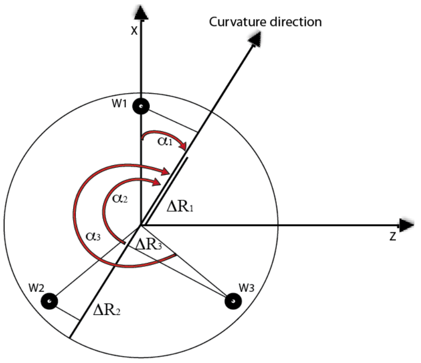


Figure 9: Projection of the wire to get the α direction.

For the 3D case, a virtual wire is considered, which gives the α direction of the curvature. Considering one virtual wire in the direction of the desired curvature having length calculated as below. First, the following lengths are calculated:

$$
\begin{aligned}
L_{11}&=R+\frac{D_1}{2}\cdot cos(\alpha_1) \quad L_{12}=R+\frac{D_2}{2}\cdot cos(\alpha_1)\\
L_{21}&=R+\frac{D_1}{2}\cdot cos(\alpha_2) \quad L_{22}=R+\frac{D_2}{2}\cdot cos(\alpha_2) \qquad (9)\\
L_{31}&=R+\frac{D_1}{2}\cdot cos(\alpha_3) \quad L_{22}=R+\frac{D_2}{2}\cdot cos(\alpha_3)
\end{aligned}
$$

$$
\begin{cases}
\alpha_1 = -\alpha\\
\alpha_2 = 120° - \alpha\\
\alpha_3 = 240° - \alpha
\end{cases} \qquad (10)
$$

Based on (6) and (9) the curvature radius R1, R2 and R3 of the three control wires are then obtained. Finally the lengths of the control wires are calculated with relation (11):

$$
\begin{aligned}
L_{w1} &= R_1 \cdot \theta\\
L_{w2} &= R_2 \cdot \theta \qquad (11)\\
L_{w3} &= R_3 \cdot \theta
\end{aligned}
$$

Besides, for the system presented we can obtain two useful relations:

$$
\begin{cases}
\sum_{i=1}^{3} \cos(\alpha_i) = 0\\
\frac{1}{3}\sum_{i=1}^{3} Lw_i = L
\end{cases} \qquad (12)
$$

The second equation can be utilized to estimate the compression or the extension of the central bone.

4 EXPERIMENTS

We conducted a series of experiments, having as purpose the analysis of the robotic structure, including the behavior of the arm and the stepper motors used for its actuation.

For these tests, we used the following equipment:

- Spider 8 acquisition system, with eight input channels and 12 bit resolution per channel
- Nexus 2692-A-0I4 signal conditioner with load compensation, four input/output channels, frequency domain 0.1Hz ÷ 100KHz and linearity error ≤0.05%;
- Bruel&Kjaer model 4391 accelerometers, frequency domain 0.3 ÷ 10000Hz, load 1 pC/ms-2, linearity error ≤1%;

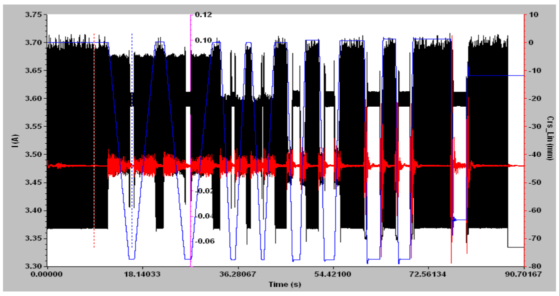


Figure 10: Recorded dataset. Marked with black - the current, with red - end effector oscillation and with blue - the displacement.

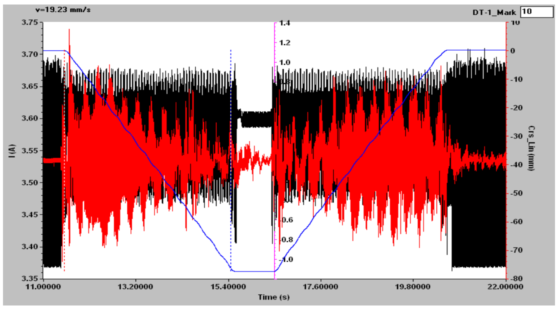


Figure 11: Recorded data for a movement executed at 19.23 mm/s.

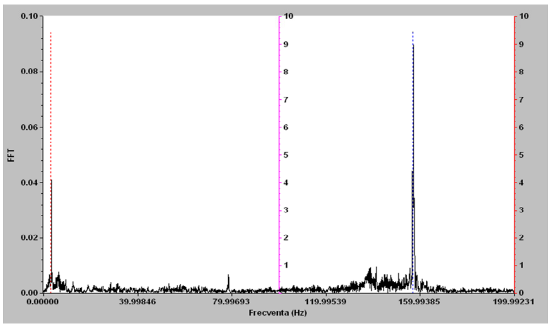


Figure 12: Frequency analysis for oscillation. Slow speed case.

- WA300 inductive transducer for linear displacement, displacement range 0÷300mm, linearity error ≤1%;
- MicroSwitch model CSLA 20KI current transducer, current domain ±20A, linearity error ≤1%
- The data was recorded on a IBM notebook with Testpoint V5.1 soft-ware installed

We placed the sensors as follows:

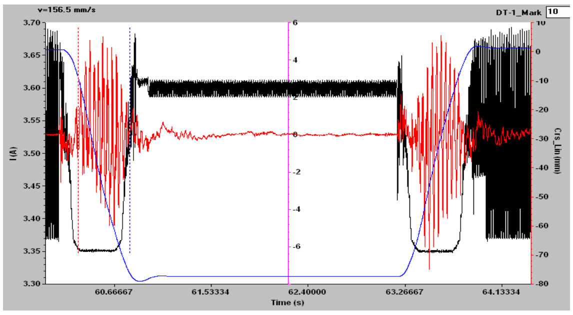


Figure 13: Medium speed movement (156.99 mm/s).

- The displacement sensor was connected at the robot's arm free end. The objective is to measure the total displacement realized by the end effector. Because of the sensor's intrinsic structure, and for obtaining a linear relation between sensor measurement and robot displacement in the 3D space, we restricted the robot's movement to a movement in a plane formed by the robot's backbone and the sensor's extension axis
- The accelerometer was placed in the same point as the displacement sensor. The objective is to determine the vibration induced in the robot arm structure by the actuation system and the control algorithm.
- The ammeter was connected in a serial mode with one of the stepper motor phases (the stepper motors are tree phases ones). The objective is to determine the stepper motor current need during the movements, and also to detect if and when the losing steps phenomenon occurs, based on the fact that a losing step process can be noticed by observing a high current peak.

In the figure 10 can be observed a recorded dataset. The measured current is represented with black color, with blue is represented the arm displacement, and with red is marked the oscillation recorded at the end of the robot arm. The dataset recorded a same trajectory described by the robot, but at different speeds. In the left side can be observed the movement execution at a lowest speed (19.23 mm/s), and in the right side at high speed (330 mm/s). In the following paragraphs we will analyze the robot's behavior for these two extreme movements. The first move observed was executed with a speed of 19.23 mm/s, going forward from the origin to the end point and, after a short break, back to the original. The detailed recorded data can be seen in the Figure 11.

The oscillation was amplified 100 times, for having a better view for its evolution. As expected for slow moves case, when the effort needed for robot arm displacement is low, the current evolution indicates that the stepper motor doesn't lose any steps and the movement there is no position error. Regarding oscillation, one can observe that near the origin position the robot exhibits a higher oscillation of its end effector. We consider that this phenomenon appeared due to the fact that near origin the tension in the actuation cables is lower, because the potential energy due gravity have the lowest value, and at end point the potential energy have a considerable higher value, conducting to a higher tension in cables which prevents the robot arm to oscillate.

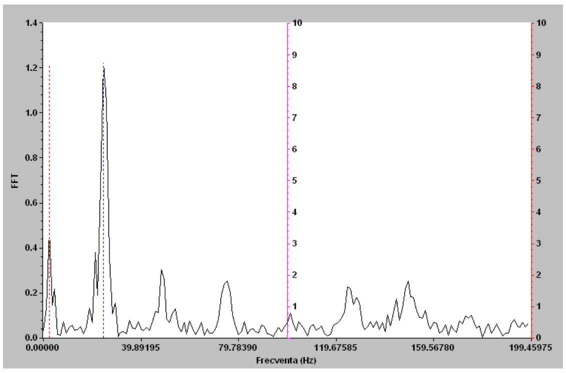


Figure 14: Frequency analysis for oscillation. Medium speed case.

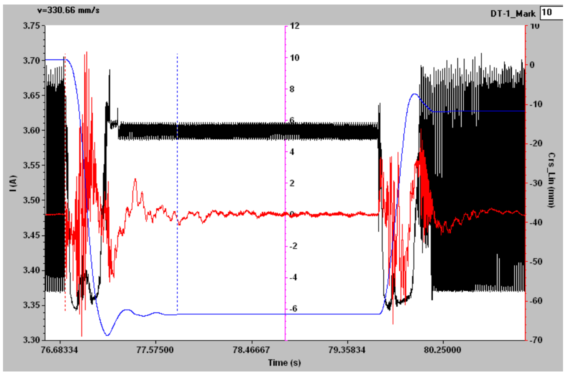


Figure 15: High speed movement.

If we take a closer look at the oscillation, and make a frequency analysis (figure 12), it can be observed that there are two spikes in frequency, one at 2.9 Hz, and one at 159.99 Hz.

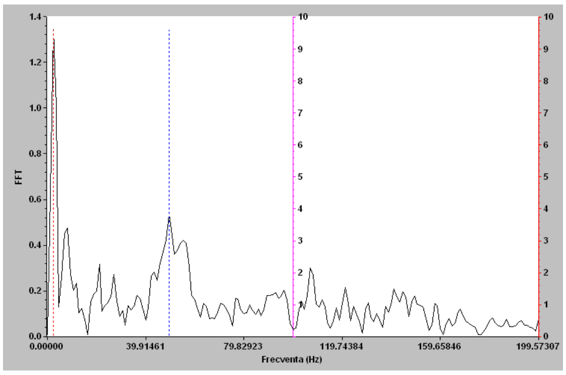


Figure 16: Frequency analysis for oscillation. High speed case.

As already identified, where an artificial vision system was used (Tanasie et al., 2009), the first spike at 2.9 Hz represents the natural resonance frequency for the robotic arm. By observing the oscillation analysis for faster movements, we determined that the second spike, with a higher level, placed at 159.99 Hz, is induced by the actuation system – the vibrations frequency generated by the actuation system and transmitted in the cables, also being responsible for the resonance spike at 2.9 Hz.

Considering a medium speed movement (156.99 mm/s), the following dataset was recorded (Figure 13). Analyzing the current evolution, a singular spike can be observed at 63.8 second on the timescale, which indicates that the motor loses a single step, this situation being caused by the fast stop of the robotic arm. Correlated with this fact, the analysis of the trajectory evolution shows that the end of the robot arm don't stop directly at the desired position, and under the effect of inertia determined by its own mass it passes that position. While the cables exhibit neither compression nor extension, the position error is determined exclusively by a shear effect. Shear is an elastic deformation, thus generating a force that brings the robot arm back at the desired position. This transitory effect takes 0.2 seconds and the maximum position error is 2 millimeters.

The oscillation graph shows a similar evolution as described for the previous case, but the frequency analysis gives a different characteristic (Figure 14). The natural oscillation frequency can be identified as a spike at the same frequency 2.9 Hz, but the vibration induced by the actuation system may be found now at 24.6 Hz, as a result of the motor's speed variation.

The last observed evolution is for high speed movement (330 mm/s). The recorded data is represented in the Figure 15. The current evolution is more dramatic and irregular, as can be observed in the figure 15 (marked with black line). The current profile isn't anymore smooth, as in previous cases, and some major fluctuations can be seen at 76.9 sec and at 79.7 sec. Those fluctuations correspond to a large number of lost steps, the effort required for moving the robot being very large. This conclusion is sustained by the conclusions of analysis conducted for displacement and vibration. Regarding the displacement, the robot was programmed to execute the very same movement, going from origin to a desired position and then back to the origin, but at different speeds. This means that in all recorded data, the robot must achieve the same desired position, and the position at the end of the movement must be the same with the position from where the movement started (the origin). Observing the displacement from figure 15 (marked with blue line), it can be determined a difference of 12 mm between the beginning position and final position. Also, if one compares the desired position achieved in figure 11, 13 and 15, can determine that the desired position is achieved successfully in figure 11 and 13, but in figure 15 the robot fails to reach the desired position, introducing a position error of 6 mm. The inertial effect is very high, the robot presenting a large oscillation before remaining steady in the desired position. Previous conclusions are also sustained by the recorded vibrations, marked with red line in figure 15. On the same timeline with the current fluctuations can be noticed a large amplitude of the vibration, as an effect of the discontinuity in movement determined by lost steps. Moreover, as the frequency analysis shown in the figure 16 denotes, this high speed actuation determines a resonance with the natural frequency of the robot arm, affecting the robot arm evolution.

5 CONCLUSIONS

As conclusions, for the slow speed moves, there are no lost steps in stepper motors, meaning that an open loop control can be successfully applied to the system. There is a low amplitude oscillation of the robot's arm, mainly at 159.99 Hz, induced by the actuation system, but it induces only a small amplitude natural frequency resonance, so the robot arm doesn't resonate with actuation system. Also, considering the displacement (marked with blue in the figure), it can be observed that the position error is zero. The overall behavior of the structure is a very good one.

For the medium speed moves, the stepper motors may lose steps, but this phenomenon occurs occasionally and can be considered insignificant if is evaluated as a ratio lost steps / total number of steps, meaning that an open loop control can still be successfully applied to the system. There is a low amplitude oscillation of the robot's arm, mainly at 24.6 Hz, induced by the actuation system, but it induces only a small amplitude natural frequency resonance, so the robot arm doesn't resonate with actuation system. The displacement record reveals a transient evolution around the final position, introducing a transient position error, having 2 mm maximum amplitude and with a short duration, 0.2 seconds. At the end of the transient evolution, the robot stops on the desired position and the position error is zero. The overall behavior of the structure is a good one.

ACKNOWLEDGEMENTS

The research presented in this paper was supported by the Romanian National University Research Council CNCSIS through the IDEI Research Grant ID93 and by FP6 MARTN through FREESUBNET Project no. 36186.

REFERENCES

Blessing, M., Walker, I. D., 2004. Novel Continuum Robots with Variable- Length Sections, Proceedings 3rd IFAC Symposium on Mechatronic Systems, Sydney, Australia, September 2004, pp. 55-60.

Boccolato, G., Manta, F., Dumitru, S., Cojocaru, D., 2009. 3D Control for a Tentacle Robot, 3rd International Conference on Applied Mathematics, Simulation, Modelling (ASM'09), Vouliagmeni Beach, Athens, Greece.

Cojocaru, D., Tanasie, R. T., Marghitu, D. B., 2008. A Complex Mathematical Expression Operations Tool, Annals of The Univ. of Craiova, Series: Automation, Computers, Electronics And Mechatronics, ISSN: 1841-0626, Volume 5(32), no. 1, pp. 7-12.

Cowan, L. S. and Walker, I. D., 2008. "Soft" Continuum Robots: the Interaction of Continuous and Discrete Elements, Artificial Life XI.

Crespi, A. and Ijspeert, A. J., 2006. An amphibious snake robot that crawls and swims using a central pattern generator. 9th International Conference on Climbing and Walking Robots (CLAWAR 2006), pp19-27.

Dumitru, S., Cojocaru, D., Dumitru, N., Ciupitu, I., Geonea, I., Dumitru, V., 2009. Finite Element Modeling of a Polyarticulated Robotic System, Annals of DAAAM, ISSN 1726-9679, pp 969.

Ivanescu, M., Cojocaru, D., Popescu, N., Popescu, D., Tănasie, R.T., 2006. Hyperredundant Robot Control by Visual Servoing, Studies in Informatics and Control Journal, Volume 15, Number 1, pp93-102, ISSN 1220-1766.

Robinson, G., Davies, J. B. C., 1999. Continuum robots—A state of the art. IEEE International Conference on Robotics and Automation, pp2849–2854. Detroit, MI.

Suzumori, K., Iikura, S., Tanaka, H., 1991. Development of Flexible Microactuator and its application to Robotic Mechanisms, Proceedings of the IEEE International Conference on Robotics and Automation, pp. 1622-1627.

Tanasie, R. T., Ivanescu, M., Cojocaru, D., 2009. Camera Positioning and Orienting for Hyperredundant Robots Visual Servoing Applications, Journal of Control Engineering and Applied Informatics, Vol 11, No 1, p19-26, ISSN 1454-8658.

A ROBUST MOSAICING METHOD FOR ROBOTIC ASSISTED MINIMALLY INVASIVE SURGERY

Mingxing Hu, David J. Hawkes
Centre for Medical Image Computing, University College London, London, U.K.
{mingxing.hu, d.hawkes}@ucl.ac.uk

Graeme P. Penney
Department of Imaging Sciences, King's College London, U.K.
graeme.penney@kcl.ac.uk

Daniel Rueckert, Philip J. Edwards, Fernando Bello, Michael Figl
Department of Computing, Imperial College, London, U.K.
{d.rueckert, eddie.edwards, f.bello, m.figl}@imperial.ac.uk

Roberto Casula
Cardiothoracic Surgery, St. Mary's Hospital, London, U.K.
roberto.casula@st-mary.nhs.uk

Keywords: Video Mosaicing, Robotic Assisted Minimally Invasive Surgery, Homography, Trifocal Tensor, Bundle Adjustment.

Abstract: Constructing a mosaicing image with broader field-of-view has become an interesting topic in image guided diagnosis and treatment. In this paper, we present a robust method for video mosaicing in order to provide more guiding information for robotic assisted minimally invasive surgery. Outliers involved in the feature dataset are removed using trifocal constraints, homographies between images are estimated with L_∞-norm optimization and chained together in a practical way. Finally refinement based on bundle adjustment is applied to minimize the error between reprojection and feature measurement. The proposed method has been tested with endoscopic images from Totally Endoscopic Coronary Artery Bypass (TECAB) surgery. The results showed our method performs better than other typical methods in terms of accuracy and robustness to deformation.

1 INTRODUCTION

The past decade has witnessed significant advances on robotic assisted Minimally Invasive Surgery (MIS) evolving from early laboratory experiments to an indispensable tool for many surgeries. MIS offers great benefits to patients: the incisions and trauma are reduced and hospitalisation time is shorter. Robotic assisted techniques further enhance the manual dexterity of the surgeon and enable him to concentrate on the surgical procedure. Despite of all these advantages, MIS using an endoscope still suffers from a fundamental problem: the narrow field-of-view. As a result, the restricted vision impedes the surgeon's ability to collect visual information from the scenes and his/her awareness of peripheral sites.

A straightforward solution to overcome the difficulty is video mosaicing, creating a 2D image with wider field-of-view by aligning and properly blending a number of partly overlapped images acquired at different positions. A lot of research work about video mosaicing has been done in both computer vision and medical imaging communities. 1n 1975, Milgram (Milgram, 1975) proposed the first photomosaics method by minimizing the visual impact of the introduced seam. Geometric and greyscale information was used to combine the images on a line-by-line basis and to choose a best seam point for each line. After that, this area has attracted great attention from researchers in computer vision community. For example, Zoghlami *et al.* (Zoghlami *et al.*, 1997) proposed a feature-based algorithm to compute the homography between images with relatively small overlap and experimental results showed that it could deal with large rotation around optical axis and zooming factor. Alternatively, Capel (Capel, 2001) focused on the global registration for the video mosaicing, the alignment of the image frames, taking into account all the overlapped images, and not just the consecutive ones. Maximum likelihood estimate was used to build the chain of consisted homographies using all the available feature points. Most recently, Brown and Lowe (Brown and Lowe, 2007) introduced an automatic mosaicing method based on the invariance features. The features are detected and matched together between images using SIFT (Lowe, 2004). This method is robust to orientation, scale and illumination of the input images and can recognize multiple panoramas in an unordered image dataset. These methods work well for static scene without any deformable objects in it. However, medical image usually involves some deformation from organs and soft tissues, which often lead to the failure of these methods.

In medical imaging community, Seshamani *et al.* (Seshamani *et al.*, 2006) presented an endoscopic mosaicing technique to display a wider field-of-view of the surgical scene by stitching together images. This method, which was evaluated using microscopic retinal and catadioptric endometrial images, can perform online image registration and provide warping models to handle tubular organ structure. Vercauteren *et al.* (Vercauteren *et al.*, 2006) also proposed a similar mosaicing method but they applied statistics for Riemannian manifolds to pairwise registration. Their method is able to produce a globally consistent mapping of input frames which is also aligned to a reference plane. It also considers non-rigid deformations of soft tissue, and the irregular sampling present in fibered confocal microscopy. Recently Miranda-Luna *et al.* (Miranda-Luna *et al.*, 2008) also proposed a method of mosaicing of bladder endoscopic images by mutual information-based similarity measure and stochastic gradient optimization. Besides, an undistortion method is used to preprocess the endoscopic images in order to improve the robustness of the registration. Unfortunately, a common trait shared by these methods is the requirement of large overlap to guarantee the convergence and accuracy of the local and global alignment.

So in this paper, we propose a robust method to mosaic medical images for robotic assisted minimally invasive surgery. Good features are detected and tracked based on the optical flow and then the potential outliers are removed from the feature dataset using the trifocal tensor. Homographies between images are estimated using Second-Order Cone Programming (SOCP) under L_∞-norm. Then they are chained together under a common and global reference system, followed by bundle adjustment refinement to minimize the total misalignment. The contributions of the proposed method are as follows: (1) Mosaicing image with a broader field-of-view can be constructed from the input images containing deformable organs and soft tissues. Thus it can be used for 2D-3D registration of the anatomy to the preoperative CT/MRI data in order to provide more information for image guided diagnosis or surgery. (2) A robust strategy based on the trifocal tensor and bundle adjustment is used to remove outliers obtained from incorrect locations and incorrect tracking and to obtain the global alignment by minimizing the reprojection error.

2 ROBUST ESTIMATION FOR VIDEO MOSAICING

Given a set of images I^i ($i = 1, \cdots, m$), and some image point $\mathbf{x}_k^i = (x_k^i, y_k^i, 1)^T$ detected on each frame i. If two images I^i and I^j can be related by a linear transformation of the projective plane, we have

$$\mathbf{x}^i = \mathbf{H}^{i,j}\mathbf{x}^j \tag{1}$$

where $\mathbf{H}$ is a 3×3 matrix, representing the 2D-2D transformation via a projective plane, also called a homography.

2.1 Feature Detection and Tracking

The first step to construct mosaicing image is to track image features as the camera moves. One of the well-known tracking methods is the Lucas-Kanade (LK) tracker (Tomasi and Kanade, 1992). The LK tracker minimizes the sum of squared errors between two images I^k and I^{k+1} by altering the warping parameters $\mathbf{p}$ which are used to warp I^{k+1} to the coordinate frame of I^k. For a general motion model with transformation function $W(\mathbf{x}, \mathbf{p})$, the objective function is

$$\min \sum_{\mathbf{x}} \left[I^{k+1}(W(\mathbf{x}; \mathbf{p} + \Delta\mathbf{p})) - I^k(\mathbf{x}) \right]^2 \tag{2}$$

This expression is linearized by a first order Taylor expansion on $I^{k+1}(W(\mathbf{x};\mathbf{p}+\Delta\mathbf{p}))$

$$\min \sum_{\mathbf{x}} \left[I^{k+1}(W(\mathbf{x};\mathbf{p})) + \nabla I^{k+1} \frac{\partial W}{\partial \mathbf{p}} - I^{k}(\mathbf{x}) \right]^2 \qquad (3)$$

Where ∇I^{k+1} is the image gradient vector and $\partial W / \partial \mathbf{p}$ is the Jacobian of the transformation function.

2.2 Outlier Removal

Usually there are some outliers in the feature dataset after the detection and tracking, and they are in gross disagreement with a specific postulated model and must be handled by robust approaches. More importantly, the L_∞ optimization, which will be discussed in the next section, is very vulnerable to outliers. So the outlier removal is crucial to the success of the whole mosaicing process.

Given three cameras characterized by projective matrices $\mathbf{P}=[\mathbf{I}|\mathbf{0}]$, $\mathbf{P}'=[\mathbf{A}|\mathbf{V}']$, $\mathbf{P}''=[\mathbf{B}|\mathbf{V}'']$, the images of a 3D point in each view can be denoted as $\mathbf{x}=(x,y,1)^T$, $\mathbf{x}'=(x',y',1)^T$, $\mathbf{x}''=(x'',y'',1)^T$ in homogeneous coordinates. It can be noted that matrices $\mathbf{A}$ and $\mathbf{B}$ are 2D homograph matrices, where $\mathbf{x}'=\mathbf{A}\mathbf{x}$ and $\mathbf{x}''=\mathbf{B}\mathbf{x}$, and $\mathbf{V}'$ and $\mathbf{V}''$ are the projection of the first camera centre into the second and third images. Then the trilinear constraints across the three views can be compactly expressed in terms of trifocal tensor, $\mathbf{T}_i^{jk}$, which is a $3\times3\times3$ matrix with 27 entries. And the relation $\mathbf{x}\leftrightarrow\mathbf{x}'\leftrightarrow\mathbf{x}''$ can be described as (Shashua, 1995)

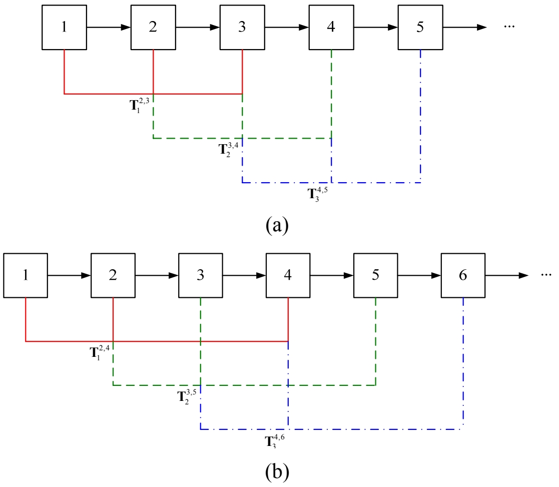
$$\mathbf{T}_i^{jk} = \mathbf{v}'^{j}\mathbf{b}_i^k - \mathbf{v}''^{k}\mathbf{a}_i^j, \qquad i,\ j,\ k=1,\ 2,\ 3 \qquad (4)$$

Since every corresponding triplet $\mathbf{x}$, $\mathbf{x}'$, $\mathbf{x}''$ contributes four linearly independent equations, then seven point correspondences uniquely determine (up to scale) the tensor $\mathbf{T}$. In fact the trifocal tensor can be estimated from a minimum of six point correspondences since it has only 18 degrees of freedom. However, the six-point estimation involves the solution of a cubic and a complicated parameterization (Quan, 1994), and so for simplicity, we use the seven-point method to compute a possible solution and employ the RANSAC strategy to detect the outliers based on the geometric error.

$$R=\sum_{i=1}^{n} R_i = \sum_{i=1}^{n} d(\mathbf{x}_i,\ \hat{\mathbf{x}}_i)^2 + d(\mathbf{x}'_i,\ \hat{\mathbf{x}}'_i)^2 + d(\mathbf{x}''_i,\ \hat{\mathbf{x}}''_i)^2 \qquad (5)$$

This error measures the sum-of-squares of the geometric distances between the image points $\mathbf{x}_i\leftrightarrow\mathbf{x}'_i\leftrightarrow\mathbf{x}''_i$ and the corrected data points $\hat{\mathbf{x}}_i\leftrightarrow\hat{\mathbf{x}}'_i\leftrightarrow\hat{\mathbf{x}}''_i$, with the latter obeying the trilinear constraint Eq. (4) for the estimated tensor $\mathbf{T}$. Thus, given three images with overlap, we can estimate the trifocal tensor among them and use the above error measure to detect outliers accordingly.

The above method is only applicable to three images, we require a method to process an entire image sequence and remove the outliers. The simplest way is to compute the tensor among three consecutive images, $(i,i+1,i+2)$, e.g., image triplet, $(1,2,3)$, $(2,3,4)$, etc, as shown in Fig.1 (a), and delete feature points if they are considered an outlier from any independent tensor estimation. Besides, we also employ additional image triplets for computation, e.g., $(i,i+1,i+3)$, as shown in Fig.1 (b). However, the more image triplets that are used, the more feature points will be removed if a previous decision rule is applied (e.g. once an outlier, always an outlier). Our method carries out a number of independent tests (each time using a unique combination of three images) on each feature point. Feature points are only removed if they are determined to be outliers more than 50% of the times.



(a)

(b)

Figure 1: Strategy of outlier removal based on trifocal tensor. (a) The three consecutive images $(i,i+1,i+2)$ are used to compute the trifocal tensor. (b) More nearby images $(i,i+1,i+3)$ are used to remove the outliers from the image sequence.

2.3 Image Alignment

Today L_∞-norm optimization has been widely used in various multiple-view geometry problems (Kahl and Hartley, 2008). One of the main advantages of L_∞ is that: problems formulated by L_∞ often possess a single, hence global, optimum. Besides, it usually leads to a simpler formulation for the same problem compared with L_2.

Without loss of generality, we set the last element of the homography $\mathbf{H}$, h_{33}, to 1 and have

$$\mathbf{x}_k^i = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \mathbf{x}_k^j$$

So the residual of homography estimation between image i and j can be expressed as

$$d\left(\mathbf{x}_k^i, \mathbf{x}_k^j\right) = \left\| \frac{\mathbf{h}_1^T \mathbf{x}_k^j}{\mathbf{h}_3^T \mathbf{x}_k^j} - x_k^i, \frac{\mathbf{h}_2^T \mathbf{x}_k^j}{\mathbf{h}_3^T \mathbf{x}_k^j} - y_k^i \right\|$$

$$= \frac{\sqrt{\left(\left(\mathbf{h}_1^T - x_k^i \mathbf{h}_3^T\right)\mathbf{x}_k^j\right)^2 + \left(\left(\mathbf{h}_2^T - y_k^i \mathbf{h}_3^T\right)\mathbf{x}_k^j\right)^2}}{\mathbf{h}_3^T \mathbf{x}_k^j} = \frac{\sqrt{f_{k1}(\mathbf{s})^2 + f_{k2}(\mathbf{s})^2}}{\lambda_k(\mathbf{s})} \qquad (6)$$

where $\mathbf{h}_l^T$ represents the l-th row of the matrix $\mathbf{H}$. So our aim is to solve the following optimization problem by minimizing the residual

$$\min \sum_{k=1}^{m} d\left(\mathbf{x}_k^i, \mathbf{x}_k^j\right)^2 \text{ , s.t. } \lambda_k(\mathbf{s}) > 0$$

Suppose each residual has an upper bound γ_k, that is, $\left(f_{k1}(\mathbf{s})^2 + f_{k2}(\mathbf{s})^2\right) / \lambda_k(\mathbf{s})^2 \leq \gamma_k$. Then the formulation in (6) is equivalent to

$$\min \gamma_1 + \gamma_2 + \cdots + \gamma_m$$

$$\text{s.t. } f_{k1}(\mathbf{s})^2 + f_{k2}(\mathbf{s})^2 \leq \gamma_k \lambda_k(\mathbf{s})^2 \text{ , } k = 1, \cdots, m$$

$$\lambda_k(\mathbf{s}) > 0$$

Then we can use Second-Order Cone Programming (Alizadeh and Goldfarb, 2003) to solve this problem. Readers can refer to Kahl's paper for more details (Kahl and Hartley, 2008).

Ideally, after the alignment of all consecutive images, we can chain all the images together and wrap them onto a reference plane

$$\mathbf{H}^{r,i} = \begin{cases} \mathbf{I} & \text{if } i = r \\ \mathbf{H}^{r,i-1}\mathbf{H}^{i-1,i} & \text{if } i > r \\ \mathbf{H}^{r,i+1}\mathbf{H}^{i+1,i} & \text{if } i < r \end{cases}$$

Here image r is the reference frame. For simplicity, it can be the middle image of the whole video sequence.

However, the misalignment error usually accumulates by concatenating homographies. This is especially evident when the camera goes back to the scene previously seen in a long image sequence. The accumulation of error may be so great that the first and last images are very poorly registered. In other words, the homographies are not consistent with alignment to a common frame. So we use a strategy to minimize the number of good homographies to link image i with reference frame r:

(1) Find image j, which is the furthest to image r but with enough overlap. Here the overlap can be the number of feature correspondences between image j and r

$$n^{j,r} \geq \delta_{overlap}$$

(2) Compute the homography between frame k and i, and calculate the mean of the residual error

$$D_{j,r} = \frac{1}{n^{j,r}} \sum_k d\left(\mathbf{x}_k^i, \mathbf{x}_k^j\right) \qquad (7)$$

(3) If $D_{j,r}$ is small enough, $D_{j,r} \leq \delta_{residual}$, this homography $\mathbf{H}^{j,r}$ is accepted. Then we start from image j, $r = j$, and find the next acceptable homography $\mathbf{H}^{j,r}$ using step (1) and (2). If $D_{j,r} > \delta_{residual}$, we select the image next to j

$$j = \begin{cases} j+1 & \text{if } r > j \\ j-1 & \text{if } r > j \end{cases}$$

and repeat (2) and (3).

(4) The process will halt until the whole homography chain is built.

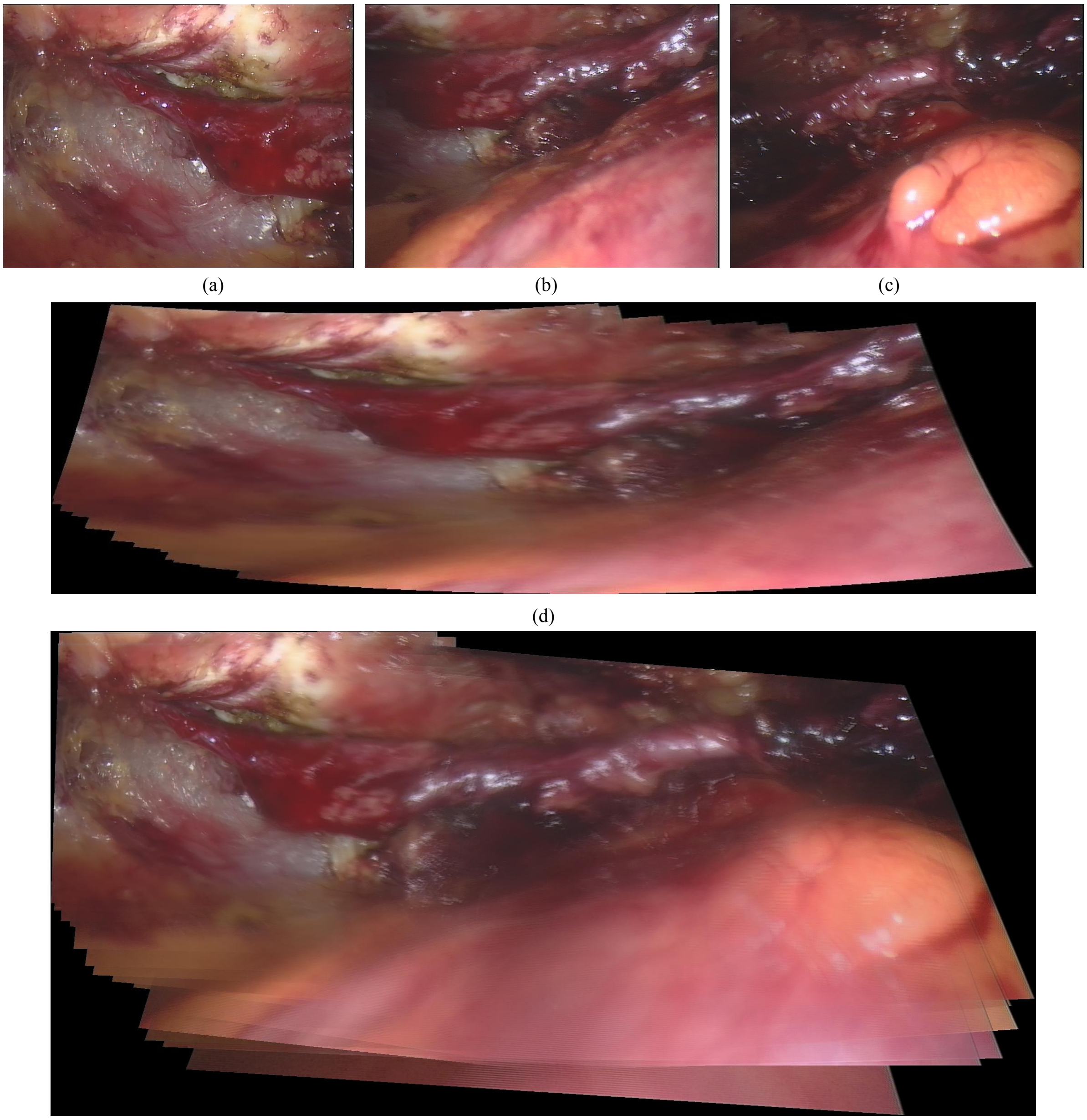
Thus, alignment can take advantage of homographies linking non-consecutive frames and reduces the global registration error.

2.4 Refinement based on Bundle Adjustment

The bundle adjustment (BA, Triggs *et al.*, 1999) we used is different from the ones addressed in McLauchlan's (McLauchlan, and Jaenicke, 2002) and Brown's paper (Brown and Lowe, 2007). In their papers, BA was used to solve the rotation parameters and focal lengths of all cameras. In this paper, BA was performed to find the best homography set $\left\{\mathbf{H}^{i,r}\right\}$, $i = 1, \cdots, m$, that minimize the misalignment error.

$$\min_{\left\{\mathbf{H}^{i,r}\right\}} \sum_{i=1,\cdots,m}^{i \neq r} \sum \left\| \mathbf{x}_k^i - \mathbf{H}^{i,r} \tilde{\mathbf{x}}_k^r \right\|^2 \qquad (8)$$

where $\tilde{\mathbf{x}}^r$ is the reprojecion of all the feature points onto frame r. It can be easily computed using Least Square method with all the available homographies. Then Levenberg–Marquardt algorithm is used to sol-



(e)

Figure 2: The experimental result of endoscopic images from Totally Endoscopic Coronary Artery Bypass surgery. (a), (b) and (c) show the first, middle and last images of the sequence, respectively. (d) displays the mosaicing result of Brown's Method. (e) displays the mosaicing result of the proposed method.

ve Eq. (8). The C++ code about the generic sparse bundle adjustment is available online by courtesy of Manolis Lourakis.

3 EXPERIMENTAL RESULTS

In this section, the performance of the proposed method was evaluated using endoscopic images from Totally Endoscopic Coronary Artery Bypass (TECAB) surgery and compared with Brown's method (Brown and Lowe, 2007).

The *da Vinci*^{TM} robotic surgical system **(**Intuitive Surgical, Inc., Sunnyvale**,** CA, USA) was used to obtain images of the heart surface. The video endoscopic images were digitized at 25 frames per second (fps) using a frame grabber (LFG4 PCI64, Active Silicon, Uxbridge, U.K.). Although *da Vinci* system provides stereo vision, we only use the image sequence from left camera to perform the mosaicing in order to compare with other methods. 150 images were captured from the endoscope but we use only 30 frames (every 5 frame from the sequence) for the mosaicing. Our aim is to create a mosaicing image which includes the whole structure of the coronary artery. The main challenge is the large complicated non-rigid motion introduced by the beating heart surface, which is shown in the right bottom of Fig. 2 (b) and (c).

Fig. 2 (d) displays the mosaicing result of the proposed method. We can notice that the whole vessel structure has been built correctly. So the surgeon can realize the environment outside the current scene when he views a part of the vessel. More importantly, the mosaicing image can help him link the endoscopic video with the preoperative information from CT/MRI scan. Brown's method was also tested using this image sequence and the mosaicing result was displayed in Fig. 2 (e). It is noticed that only part (around three quarters) of the whole vessel had been constructed and the images affected badly by the beating heart surface could not be used by Brown's method. The possible reason is that SIFT feature descriptor could not find enough reliable features from the images with severe deformation from the internal organ or soft tissue.

4 CONCLUSIONS

In this paper, we proposed a robust video mosaicing method for robotic assisted minimally invasive surgery. The mosaicing image displays a much wider field-of-view of the operation scene and helps the surgeon realize the surrounding environment outside the current view. Experiments with TECAB endoscopic images and FCM images show that the proposed method performs better than other typical methods. It is robust to deformation caused by organs and soft tissues and can even deal with artefacts involved in the images.

Effort in the near future will focus on future improvement of robustness to deformation and artefacts. Our long term goal is to automatically construct mosaicing image of the surgical scene, reconstruct the internal organ surfaces and register these with the preoperative data (CT or MRI) to provide more information for image guided diagnosis and treatment.

REFERENCES

Alizadeh, F. and Goldfarb, D, 2003. Second-order cone programming, *Mathematical Programming*, 95 (1), 3-51.

Brown, M. and Lowe, D. G, 2007. Automatic Panoramic Image Stitching using Invariant Features, *International Journal of Computer Vision*, 74, 59–73.

Capel D. P. , 2001. Image Mosaicing and Super-Resolution, Ph.D thesis, Dept. of Eng. Science, Univ. of Oxford.

Kahl, F. and Hartley, R, 2008. Multiple-View Geometry Under the Linfinity-Norm, *IEEE Trans. Pattern Anal. Mach. Intell.* 30(9): 1603-1617

Lowe, D. G., 2004: Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60, 91–110.

McLauchlan, P. and Jaenicke, A, 2002. Image mosaicing using sequential bundle adjustment. *Image and Vision Computing*, 20(9–10):751–759.

Milgram D. L., 1975. Computer Methods for Creating Photomosaics, *IEEE Trans. Computers*, 24(11), 1113-1119.

Miranda-Luna, R., Daul, C., Blondel, W.C.P.M., Hernandez-Mier, Y., Wolf, D., Guillemin, F., 2008. Mosaicing of Bladder Endoscopic Image Sequences: Distortion Calibration and Registration Algorithm. *IEEE Trans. on Biomedical Engineering* 55, 541–553.

Quan, L., 1994. Invariants of 6 points from 3 uncalibrated images. *In: Proc. ECCV* , 2, 459-470.

Shashua, A., 1995. Algebraic functions for recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 17 (8), 779-789.

Seshamani, S., Lau W., and Hager, G., 2006. Real-Time Endoscopic Mosaicking. *In: Proc. MICCAI*, 355-363.

Tomasi, C., and Kanade, T., 1992. Shape and Motion from Image Streams under Orthography: a Factorization Method. *Int. J. Computer Vision*, 9(2), 137-54.

Triggs, W., McLauchlan, P., Hartley, R., and Fitzgibbon, A. 1999. Bundle adjustment: A modern synthesis. *In Vision Algorithms: Theory and Practice*, number 1883 in LNCS. Springer-Verlag. Corfu, Greece, pp. 298–373.

Vercauteren, T., Perchant, A., Malandain, G., Pennec, X., Ayache, N., 2006. Robust mosaicing with correction of motion distortions and tissue deformation for in vivo fibered microscopy. *Medical Image Analysis*, 10 (5), 673-692.

Zoghlami, I., Faugeras, O., and Deriche, R. 1997. Using geometric corners to build a 2D mosaic from a set of images. In Proc. CVPR, 420-425.

http://www.ics.forth.gr/~lourakis/sba/

ROBOT SKILL SYNTHESIS THROUGH HUMAN VISUO-MOTOR LEARNING

Humanoid Robot Statically-stable Reaching and In-place Stepping

Jan Babič, Blaž Hajdinjak
Jožef Stefan Institute, Ljubljana, Slovenia
{jan.babic, blaz.hajdinjak}@ijs.si

Erhan Oztop
ATR Computational Neuroscience Laboratories, JST-ICORP Computational Brain Project
NICT Biological ICT Group, Kyoto, Japan
erhan@atr.jp

Keywords: Humanoid robot, Skill synthesis, Visuo-motor learning, Radial basis functions.

Abstract: To achieve a desirable motion of the humanoid robots we propose a framework for robot skill-synthesis that is based on human visuo-motor learning capacity. The basic idea is to consider the humanoid robot as a tool that is intuitively controlled by a human demonstrator. Once the effortless control of the humanoid robot has been achieved, the desired behavior of the humanoid robot is obtained through practice. The successful execution of the desired motion by the human demonstrator is afterwards used for the design of motion controllers that operate autonomously. In the paper we describe our idea by presenting a couple of robot skills obtained by the proposed framework.

1 INTRODUCTION

If robots could be able to imitate human motion demonstrated to them, acquiring complex robot motions and skills would become very straightforward. One can capture the desired motion of a human subject and map this motion to the kinematical structure of the robot. Due to the different dynamical properties of the humanoid robot and the human demonstrator, the success of this approach with regard to the stability of the humanoid robot depends on the ad-hoc mapping implemented by the researcher (Schaal, 1999). Here we propose a very different approach where we use the human demonstrators real-time action to control the humanoid robot and to consecutively build an appropriate mapping between the human and the humanoid robot. This effectively creates a closed loop system where the human subject actively controls the humanoid robot motion in real time with the requirement that the robot stays stable. This requirement can be easily satisfied by the human subject because of the human brain ability to control novel tools (Oztop et al., 2006; Goldenberg and Hagmann, 1998). The robot that is controlled by the demonstrator can be considered as a tool such as a car or a snowboard when one uses it for the first time. This setup requires the humanoid robots state to be transferred to the human as the feedback information.

The proposed closed-loop approach exploits the human capability of learning to use novel tools in order to obtain a motor controller for complex motor tasks. The construction of the motor controller has two phases. In the first phase a human demonstrator performs the desired task on the humanoid robot via an intuitive interface. Subsequently in the second phase the obtained motions are acquired through machine learning to yield an independent motor controller. The two phases are shown on Figure 1.

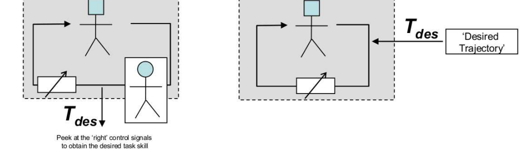

A
B
T_{des}
Peek at the 'right' control signals to obtain the desired task skill
T_{des}
'Desired Trajectory'

Figure 1: (A) Human demonstrator controls the robot in closed loop and produces the desired trajectories for the target task. (B) These signals are used to synthesize a controller for the robot to perform this task autonomously.

In the following sections, we present two example skills that were obtained by the described framework.

2 STATICALLY STABLE REACHING

The proposed approach can be considered as a closed loop approach where the human demonstrator is actively included in the main control loop as shown on Figure 2. The motion of the human demonstrator was acquired by the contact-less motion capture system. The joint angles of the demonstrator were fed forward to the humanoid robot in real-time. In effect, the human acted as an adaptive component of the control system. During such control, a partial state of the robot needs to be fed back to the human subject. For statically balanced reaching skill, the feedback we used was the rendering of the position of the robot's centre of mass superimposed on the support polygon of the robot which was presented to the demonstrator by means of a graphical display. During the experiment the demonstrator did not see the humanoid robot.

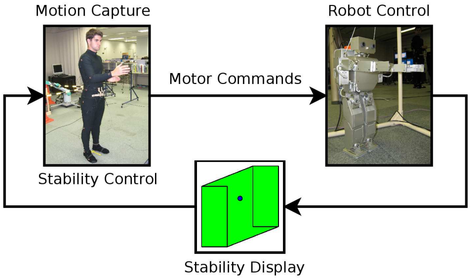


Figure 2: Closed-loop control of the humanoid robot. Motion of the human is transfered to the robot while the robot's stability is presented to the human by a visual feedback.

The demonstrator's task was to keep the center of mass of the humanoid robot within the support polygon while performing the reaching movements as directed by the experimenter. With a short practice session the demonstrator was able to move his body and limbs with the constraint that the robot's center of mass was within the support polygon. Hence the robot was statically stable when the demonstrator generated motions were either imitated by the robot in real-time or played back later on the robot. The robot used in the study was Fujitsu HOAP-II small humanoid robot.

The motion of the humanoid robot was constrained to the two dimensions; only the vertical axis and the axis normal to the trunk were considered. The light wiggly curve on Figure 3 shows the robot end-effector position data which was generated by the demonstrator. One can imagine the humanoid robot from its left side standing with the tips of the feet at the centre of the coordinate frame and reaching out outwards with its right hand gliding over the curve. The long straight segment of the curve connects the beginning and the end of the reaching motion.

For each data point of the obtained end-effector trajectory, the robot joint angles were recorded. Assuming rows of the humanoid robot end-effector position $\mathbf{X}$ is formed by the data points taken from the obtained end-effector trajectory and the robot joint angles $\mathbf{Q}$ is formed by the corresponding joint angles we get a non-linear relation of the form

$$\mathbf{Q} = \Gamma(\mathbf{X})\,\mathbf{W}. \tag{1}$$

By performing a non-linear data fit and solving for $\mathbf{W}$ we can afterwards make prediction with

$$\boldsymbol{q}_{pred} = \Gamma(\boldsymbol{x}_{des})\,\mathbf{W} \tag{2}$$

where $\boldsymbol{q}_{pred}$ is a vector of the predicted joint angles and $\boldsymbol{x}_{des}$ is a vector of the desired end-effector position. Using the prediction we can afterwards ask the humanoid robot to reach out for a desired position without falling over.

For non-linear data fitting the recorded positions $\mathbf{X}$ are mapped into an N dimensional space using the Gaussian basis functions given by

$$\varphi_i(\boldsymbol{x}) = e^{\frac{\boldsymbol{x}-\boldsymbol{\mu}_i}{\sigma^2}} \tag{3}$$

where μ_i and σ^2 are open parameters to be determined. Each row of $\mathbf{X}$ in converted into an N dimensional vector forming a data matrix

$$\mathbf{Z} = \Gamma(\mathbf{X}) = \begin{bmatrix} \varphi_1(\boldsymbol{x}_1) & \varphi_2(\boldsymbol{x}_1) & \dots & \varphi_N(\boldsymbol{x}_1) \\ \varphi_1(\boldsymbol{x}_2) & \varphi_2(\boldsymbol{x}_2) & \dots & \varphi_N(\boldsymbol{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_1(\boldsymbol{x}_m) & \varphi_2(\boldsymbol{x}_m) & \dots & \varphi_N(\boldsymbol{x}_m) \end{bmatrix}. \tag{4}$$

Assuming we have a linear relation between the rows of $\mathbf{Z}$ and $\mathbf{Q}$, we can solve Eq. (2) for $\mathbf{W}$ in the sense of the minimum least squares by

$$\mathbf{W} = \mathbf{Z}^{+}\,\mathbf{Q} \tag{5}$$

where $\mathbf{X}^{+}$ represents the pseudo-inverse of $\mathbf{X}$. The residual error is given by

$$tr\left((\mathbf{XW}-\mathbf{Q})\,(\mathbf{XW}-\mathbf{Q})^{T}\right). \tag{6}$$

In effect, this establishes a non-linear data fit; given a desired end-effector position $\boldsymbol{x}$, the joint angles that would achieve this position are given by

$$\boldsymbol{q}_{pred} = \begin{pmatrix} \varphi_1(\boldsymbol{x}_{des}) & \varphi_2(\boldsymbol{x}_{des}) & \dots & \varphi_N(\boldsymbol{x}_{des}) \end{pmatrix}\mathbf{W}. \tag{7}$$

The open parameters are N as the number of basis functions which implicitly determines $\boldsymbol{\mu}_i$ and the variance σ^2. They were determined using cross-validation. We prepared a Cartesian desired trajectory that was not a part of the recording data set and converted it into a joint trajectory with the current set values of (N, σ^2). The joint trajectory was simulated on a kinematical model of the humanoid robot producing an end-effector trajectory. The deviation of the resultant trajectory from the desired trajectory was used as a measure to choose the values of the open parameters.

Figure 3 shows the desired end-effector trajectory and the generated end-effector trajectory obtained by playing back the predicted joint angle trajectories on the humanoid robot. The light wiggly curve on Figure 3 represents the end-effector trajectory that was generated by the human demonstrator in the first phase and subsequently used to determine the mapping $\mathbf{W}$ between the joint angles and the end-effector position.

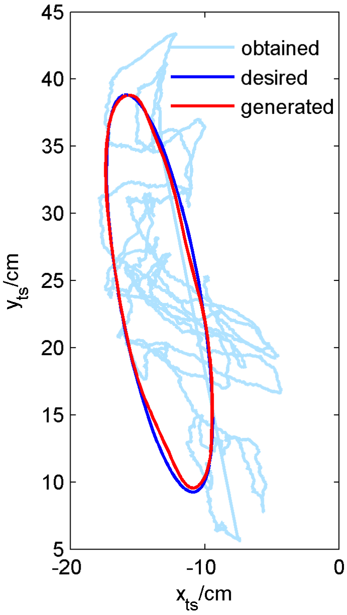


Figure 3: The obtained end-effector trajectory generated by the demonstrator (light wiggly curve) with the desired end-effector trajectory that was used as the input for the joint angle prediction and the generated end-effector trajectory obtained by playing back the predicted joint angle trajectories on the humanoid robot.

The reaching skill of the humanoid robot we obtained was statically stable which means that the robot's centre of mass was inside the robot's support polygon. A sequence of video frames representing the statically stable autonomous trajectory tracking obtained with our method is shown on Figure 4.

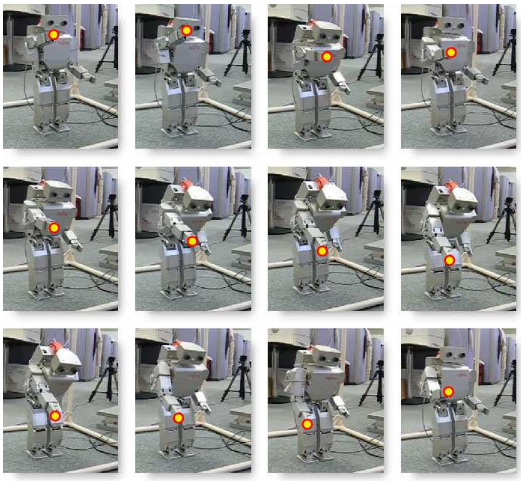


Figure 4: Video frames representing the statically stable reaching motion of the humanoid robot obtained with the proposed approach.

3 IN-PLACE STEPPING

In this section, we present our preliminary work on performing a statically stable in-place stepping of the humanoid robot. In-place stepping is a task that requires an even stricter balance control than the reaching experiment described in the previous section. In order for the humanoid robot to lift one of its feet during the statically stable in-place stepping, the robot's centre of mass needs to be shifted to the opposite leg before the lifting action occurs. As the robot's centre of mass is relatively high and the foot is relatively small, it is crucial that the position of the centre of mass of the robot can be precisely controlled. For humans, to maintain the postural stability is a very intuitive task. If one perturbs the posture of a human, he/she can easily and without any concious effort move the body to counteract the posture perturbations and to stay in a balanced posture. The main principle of our approach is to use this natural capability of humans to maintain the postural stability of the humanoid robots. In order to do so, we designed and manufactured an inclining parallel platform on which a human demonstrator is standing during the closed-loop motion transfer (Figure 5).

Instead of using visual information for the robot's stability as previously explained in the reaching experiment, the state of the humanoid robot's postural stability is feed-back to the human demonstrator by the inclining parallel platform. When the humanoid robot is statically stable, the platform stays in a horizontal position. On the contrary, when the centre of mass of the robot leaves its support polygon and therefore becomes statically unstable, the platform moves in a way that puts the human demonstrator standing on the platform in an unstable state that is directly comparable to the instability of the humanoid robot. The human demonstrator is forced to correct his/her balance by moving the body. Consecutively, as the motion of the human demonstrator is fed-forward to the humanoid robot in real-time, the humanoid robot gets back to the stable posture together with the demonstrator. Using some practice, human demonstrators easily learned how to perform in-place stepping on the humanoid robot. The obtained trajectories can afterwards be used to autonomously control the in-place stepping of the humanoid robot. Our future plans are to extend this approach and use it for acquiring walking of the humanoid robots. Figure 6 shows the human demonstrator and Fujitsu Hoap-3 humanoid robot during the in-place stepping experiment.

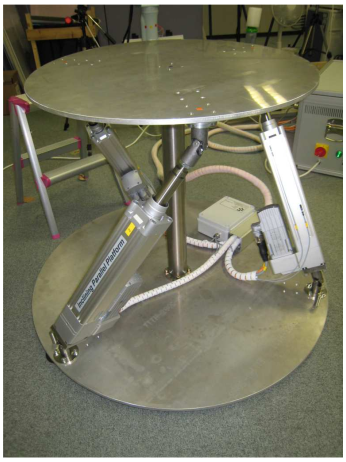


Figure 5: Inclining parallel platform that can rotate around all three axes. The diameter of the platform is 0.7*m* and is able to carry an adult human.

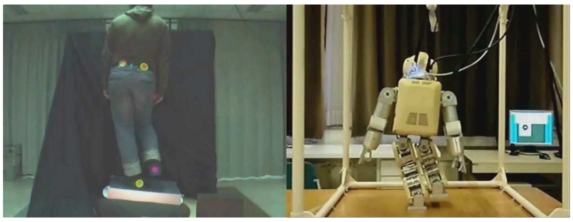


Figure 6: The human demonstrator and Fujitsu Hoap-3 humanoid robot are shown during the in-place stepping experiment. The video frame on the left side shows the human demonstrator performing in-place stepping on the inclining parallel platform. The right side frame shows the humanoid robot during the one foot posture.

4 CONCLUSIONS

A goal of imitation of motion from demonstration is to remove the burden of robot programming from the experts by letting non-experts to teach robots. The most basic method to transfer a certain motion from a demonstrator to a robot would be to directly copy the motor commands of the demonstrator to the robot (Atkeson et al., 2000) and to modify the motor commands accordingly to the robot using a sort of a local controller. Our approach is different in the sense that the correct motor commands for the robot are produced by the human demonstrator. For this convenience, the price one has to pay is the necessity of training to control the robot to achieve the desired action. Basically, instead of expert robot programming our method relies on human visuo-motor learning ability to produce the appropriate motor commands on the robot, which can be played back later or used to obtain controllers through machine learning methods as in our case of reaching.

The main result of our study is the establishment of the methods to synthesize the robot motion using human visuo-motor learning. To demonstrate the effectiveness of the proposed approach, statically stable reaching and in-place stepping was implemented on a humanoid robot using the introduced paradigm.

ACKNOWLEDGEMENTS

The research work reported here was made possible by Japanese Society for promotion of Science and Slovenian Ministry of Higher Education, Science and Technology.

REFERENCES

Atkeson, C., Hale, J., Pollick, F., Riley, M., Kotosaka, S., Schaal, S., Shibata, S., Tevatia, T., Ude, A., Vijayakumar, S., and Kawato, M. (2000). Using humanoid robots to study human behavior. *IEEE Intelligent Systems*, 15:45–56.

Goldenberg, G. and Hagmann, S. (1998). Tool use and mechanical problem solving in apraxia. *Neuropsychology*, 36:581–589.

Oztop, E., Lin, L.-H., Kawato, M., and Cheng, G. (2006). Dexterous skills transfer by extending human body schema to a robotic hand. In *IEEE-RAS International conference on humanoid robotics*.

Schaal, S. (1999). Is imitation learning the route to humanoid robots? *Trends Cogn Sci*, 3:233–242.

DYNAMIC MODELING OF A MOMENT EXCHANGE UNICYCLE ROBOT

S. Langius
Department of Mechanical Engineering, University of Twente, Enschede, 7500 AE, The Netherlands
s.langius@student.utwente.nl

R. A. de Callafon
Department of Mechanical and Aerospace Engineering, UCSD, 9500 Gilman Drive, La Jolla, CA 92093-0411, U.S.A.
callafon@ucsd.edu

Keywords: Unicycle, Robotics, Modeling.

Abstract: In this paper a three dimensional non-linear model has been derived to describe the dynamics of an unstable moment exchange unicycle robot. The robot uses a driving wheel to provide stabilization in longitudinal direction, while a second moment exchange wheel with a large inertia is used for stabilization in lateral direction. For validation purposes, the resulting equations of motion are compared independently against the simulation results of a finite element package called SPACAR. The model includes the coupling between the lateral and longitudinal motion, which makes it possible to control the yaw angle and the model can be used to design a stabilizing feedback controller.

1 INTRODUCTION

Using basic principles of kinematics and dynamics, dynamic models of robotic systems can be derived independently by hand or by an automated computer program to ensure cross model validity. This paper illustrates this approach on an inherently unstable Moment Exchange Unicycle Robot (MEUR) depicted in Figure 1 that requires stabilization in both a lateral and longitudinal direction. Using Newtonian mechanics and a finite element package called SPACAR, three dimensional non-linear models that incorporates the coupling between the lateral and longitudinal motion of the MEUR are derived. The resulting dynamic model captures the dynamics of a wide variety of moment exchange unicycle robots and can be used to design stabilizing feedback control algorithms.

Earlier work on dynamic modeling of robotic unicycles can be distinguished by their mechanism to achieve lateral stabilization (van Pommeren, 2007). An inertia wheel moving in the horizontal plane for stabilization is used in many earlier applications (Schoonwinkel, 1987; Vos and Flotow, 1990; Naveh et al., 1999). This approach is comparable to the twisting torso motion of a real unicyclist (Ohsaki et al., 2008; Sheng and Yamafuji, 1997). The gyroscopic effect of two inertias spinning in opposite direction in the horizontal plane is used in (Zenkov et al., 1999). A pendulum moving in the vertical plane perpendicular to that of the driving wheel is used in (Dao and Liu, 2005) and an inverted pendulum in the same plane is used in (Nakajima et al., 1997) where lateral stabilization is further improved by a barrel shaped wheel. The work of (Au and Xu, 1999) makes use of the gyroscopic effect for stabilization.

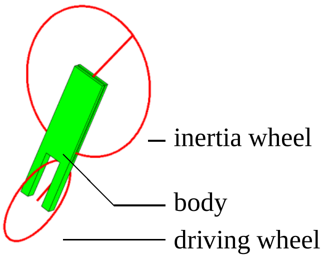


Figure 1: Schematic drawing of a Moment Exchange Unicycle Robot (MEUR).

The three dimensional model of the MEUR derived in this paper includes the coupling between the lateral and longitudinal motion, which makes it possible to control the yaw angle similar as in (Majima et al., 2006). For validation purposes of the dynamic model, the simulation results obtained from the equations of motion are compared independently against the simulation results of a finite element package called SPACAR (Aarts et al., 2008).

2 EQUATIONS OF MOTION

2.1 Model Structure and Orientation

The configuration and position of the the MEUR in Figure 1 can be described by six independent coordinates displayed in Table 1. The six coordinates do not include the pitch angle of the inertia wheel, as we assume that the inertia wheel is simply replaced by a moment acting on the body. This is a simplification of the problem, discarding the quadratic velocity terms caused by the (small) pitch velocity of the inertia wheel. Figure 2 shows all the transformations needed to apply Newtonian mechanics, where Table 2 describes all the coordinate systems involved.

Table 1: Definition of independent coordinates.

	description	applies to	axis
ψ	Yaw angle	wheel & body	a_3
γ	Roll angle	wheel & body	b_1
φ	Body pitch angle	wheel & body	c_2
θ	Wheel pitch angle	wheel	d_2
X	x-position point Q	wheel & body	a_1
Y	y-position point Q	wheel & body	a_2

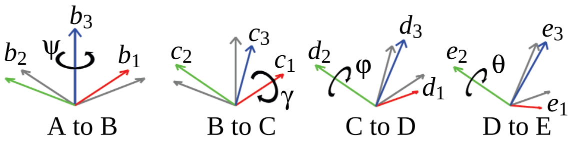


Figure 2: All coordinate system transformations.

Table 2: Definition of the coordinate systems (CS).

CS	description	unit vectors
A	Inertial CS	a_1, a_2, a_3
B	Contact force CS	b_1, b_2, b_3
C	No-slip constraint CS	c_1, c_2, c_3
D	Body CS	d_1, d_2, d_3
E	Wheel CS	e_1, e_2, e_3

The inertial coordinate system will be used to apply Newton's second law of motion, since this law is only valid when observing translational accelerations from this coordinate system. The contact force coordinate system is used to define the forces acting in the contact point. The no-slip constraint coordinate system is used for the no-slip constraint since the contact point is a stationary point only in this coordinate system. The body and wheel coordinate systems are used to apply Euler's equations, since their rotational inertias remain constant in these coordinate systems (Hughes, 1986).

The transformations $\underline{B} = \underline{A}\theta_{BA}$, $\underline{C} = \underline{B}\theta_{CB}$, $\underline{D} = \underline{C}\theta_{DC}$ and $\underline{E} = \underline{C}\theta_{EC}$ between the different coordinate systems is captured by the rotation matrices

$$\theta_{BA} = \begin{bmatrix} C_\psi & -S_\psi & 0 \\ S_\psi & C_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}, \ \theta_{CB} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & C_\gamma & -S_\gamma \\ 0 & S_\gamma & C_\gamma \end{bmatrix}$$

$$\theta_{DC} = \begin{bmatrix} C_\varphi & 0 & S_\varphi \\ 0 & 1 & 0 \\ -S_\varphi & 0 & C_\varphi \end{bmatrix}, \ \theta_{EC} = \begin{bmatrix} C_{\theta+\varphi} & 0 & S_{\theta+\varphi} \\ 0 & 1 & 0 \\ -S_{\theta+\varphi} & 0 & C_{\theta+\varphi} \end{bmatrix}$$

where the $\cos(x)$ and $\sin(x)$ terms in the rotation matrices are shortened to respectively C_x and S_x. For dynamic analysis, the derivatives of the rotation matrices are given by $\dot{\theta}_{BA} = \theta_{BA}\tilde{\omega}_{BA}$, $\dot{\theta}_{CB} = \theta_{CB}\tilde{\omega}_{CB}$, $\dot{\theta}_{DC} = \theta_{DC}\tilde{\omega}_{DC}$ and $\dot{\theta}_{EC} = \theta_{EC}\tilde{\omega}_{EC}$ where

$$\tilde{\omega}_{BA} = \begin{bmatrix} 0 & -\dot{\psi} & 0 \\ \dot{\psi} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \ \tilde{\omega}_{CB} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -\dot{\gamma} \\ 0 & \dot{\gamma} & 0 \end{bmatrix}$$

$$\tilde{\omega}_{DC} = \begin{bmatrix} 0 & 0 & \dot{\varphi} \\ 0 & 0 & 0 \\ -\dot{\varphi} & 0 & 0 \end{bmatrix}, \ \tilde{\omega}_{EC} = \begin{bmatrix} 0 & 0 & \dot{\alpha} \\ 0 & 0 & 0 \\ -\dot{\alpha} & 0 & 0 \end{bmatrix}$$

where $\dot{\alpha} = \dot{\theta} + \dot{\varphi}$.

2.2 Wheel Equations

The angular velocity vector of the wheel $\underline{\omega}$ can be expressed as the sum of all individual angular velocities each defined in their own coordinate system. When all these angular velocities are transformed to coordinate system C, leads to an expression

$$\underline{\omega}_C = \theta_{CB}{}^T \begin{Bmatrix} 0 \\ 0 \\ \dot{\psi} \end{Bmatrix} + \begin{Bmatrix} \dot{\gamma} \\ 0 \\ 0 \end{Bmatrix} + \theta_{EC} \begin{Bmatrix} 0 \\ \dot{\theta} + \dot{\varphi} \\ 0 \end{Bmatrix} \qquad (1)$$

and correspondingly, $\tilde{\omega}_C$ is a matrix composed of the elements of vector $\underline{\omega}_C$ given by

$$\tilde{\omega}_C = \begin{bmatrix} 0 & -\underline{\omega}_{C3} & \underline{\omega}_{C2} \\ \underline{\omega}_{C3} & 0 & -\underline{\omega}_{C1} \\ -\underline{\omega}_{C2} & \underline{\omega}_{C1} & 0 \end{bmatrix} \qquad (2)$$

Vector $\underline{r}_{QP}$ is shown in Figure 3 and is the vector going from point P (the contact point of the wheel) to point Q (the center of the wheel), pointing in the c_3-direction and its length is equal to radius of the wheel $\underline{r}_{QP,C} = \{0\ 0\ R\}^T$. Since no slip is assumed, the velocity at the contact point satisfies $\underline{v}_{P,C} = \underline{0}$. The velocity of the center of the wheel expressed in coordinate system C is equal to the velocity of the contact point plus the relative velocity due to rotation

$$\underline{v}_{Q,C} = \underline{v}_{P,C} + \underline{\omega}_C \times \underline{r}_{QP,C} = \tilde{\omega}_C \underline{r}_{QP,C}$$

following (Kolve, 1993) to describe the derivative of the rotation matrices and replacing the cross product by a matrix multiplication. The same velocity can be transformed to coordinate system A, which is the velocity $\underline{v}_{Q,A}$ with respect to the fixed world given by

$$\underline{v}_{Q,A} = \theta_{BA}\theta_{CB}\left(\underline{v}_{P,C} + \tilde{\omega}_C \underline{r}_{QP,C}\right) \tag{3}$$

Writing out the right hand side of (3) leads to the differential equations for the components of $\underline{v}_{Q,A}$:

$$\dot{x} = R\left(\left(\dot{\theta} + \dot{\varphi} + \dot{\psi}S_\gamma\right)C_\psi + \dot{\gamma}C_\gamma S_\psi\right) \tag{4}$$

$$\dot{y} = R\left(\left(\dot{\theta} + \dot{\varphi} + \dot{\psi}S_\gamma\right)S_\psi - \dot{\gamma}C_\gamma C_\psi\right) \tag{5}$$

$$\dot{z} = -R\dot{\gamma}S_\gamma \tag{6}$$

Solving (4) and (5) for x and y results in the solution for the first 2 independent coordinates. This can only be done numerically since these equations are non-integrable. The solution for $z = RC_\gamma$ can be obtained from the analytical solution of the integral of (6).

To find an expression for the acceleration of point Q, first (1) will be differentiated to obtain

$$\begin{aligned}\underline{\dot{\omega}}_C &= -\tilde{\omega}_{CB}{\theta_{CB}}^T \begin{Bmatrix} 0 \\ 0 \\ \dot{\psi} \end{Bmatrix} + {\theta_{CB}}^T \begin{Bmatrix} 0 \\ 0 \\ \ddot{\psi} \end{Bmatrix} + \begin{Bmatrix} \ddot{\gamma} \\ 0 \\ 0 \end{Bmatrix} \dots \\ &\dots + \theta_{EC}\tilde{\omega}_{EC} \begin{Bmatrix} 0 \\ \dot{\theta} + \dot{\varphi} \\ 0 \end{Bmatrix} + \theta_{EC} \begin{Bmatrix} 0 \\ \ddot{\theta} + \ddot{\varphi} \\ 0 \end{Bmatrix}\end{aligned}$$

and again, $\dot{\tilde{\omega}}_C$ is a matrix composed of the elements of vector $\underline{\dot{\omega}}_C$ given by

$$\dot{\tilde{\omega}}_C = \begin{bmatrix} 0 & -\underline{\dot{\omega}}_{C3} & \underline{\dot{\omega}}_{C2} \\ \underline{\dot{\omega}}_{C3} & 0 & -\underline{\dot{\omega}}_{C1} \\ -\underline{\dot{\omega}}_{C2} & \underline{\dot{\omega}}_{C1} & 0 \end{bmatrix}$$

Differentiating (3) now leads to an expression for the acceleration of point Q given by

$$\underline{\dot{v}}_{Q,A} = \theta_{BA}\left(\tilde{\omega}_{BA}\theta_{CB}\tilde{\omega}_C + \theta_{CB}\tilde{\omega}_{CB}\tilde{\omega}_C + \theta_{CB}\dot{\tilde{\omega}}_C\right)\underline{r}_{QP,C}$$

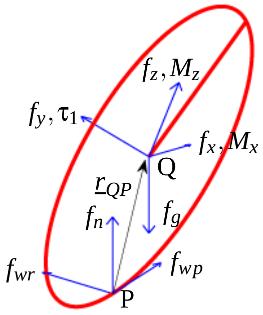


Figure 3: Forces and moments acting on the wheel.

The gravity force vector $\underline{F}_{G,B}$ does not move with any of the angles and since its direction is unaffected by rotation matrix θ_{BA} it can be defined in the coordinate system B via $\underline{F}_{G,B} = \{0\ 0\ \ -m_w g\}_B^T$ because of its simplicity later on. The contact force vector $\underline{F}_{C,B}$ is chosen to be defined in coordinate system B, so that it always rotates with the yaw angle ψ but stays in the same plane as the ground on which the wheel moves. In this way f_{wp} always points in driving direction and f_{wr} points in perpendicular direction. The third element working on the same point is the normal force f_n creating $\underline{F}_{C,B} = \{f_{wp}\ \ f_{wr}\ \ f_n\}_B^T$. Finally, the reaction force vector $\underline{F}_{R,D} = \{f_x\ \ f_y\ \ f_z\}_D$ is chosen to be defined in coordinate system D, so that it moves with the body.

Newton's second law of motion yields

$$\sum \underline{F}_A = m_w \underline{\dot{v}}_{Q,A}$$

where the left hand side is the sum of previously mentioned forces, transformed to the inertial coordinate system A, given by

$$\sum \underline{F}_A = \theta_{BA}\left(\underline{F}_{C,B} + \underline{F}_{G,B} + \theta_{CB}\theta_{DC}\underline{F}_{R,D}\right)$$

and leads to an expression for the unknown contact force vector

$$\underline{F}_{C,B} = {\theta_{BA}}^T m_w \underline{\dot{v}}_{Q,A} - F_{G,B} - \theta_{CB}\theta_{DC}F_{R,D} \tag{7}$$

The reaction moment vector $\underline{M}_{R,D}$ moves with the body, just like the reaction force vector, and is therefore defined in coordinate system D. The second element of $\underline{M}_{R,D}$ is equal to the torque applied by the motor between the driving wheel and body leading to $\underline{M}_{R,D} = \{M_x\ \tau_1\ M_z\}_D$. Since the wheel is rotation symmetric about the c_2-axis, the pitch angles have no influence on the inertias and Euler's equation

$$\sum \underline{M} = \underline{\dot{h}} \tag{8}$$

applies in coordinate system C. The angular momentum is expressed in coordinate system C by

$$\underline{h} = J \cdot \underline{\omega} = \left(\underline{C}J_C\underline{C}^T\right) \cdot \left(\underline{C}\underline{\omega}_C\right) = \underline{C}J_C\underline{\omega}_C$$

where

$$J_C = \begin{bmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{bmatrix}$$

The derivative of $\underline{h}$ can then be written as

$$\begin{aligned}\underline{\dot{h}} &= \underline{\dot{C}}J_C\underline{\omega}_C + \underline{C}\dot{J}_C\underline{\omega}_C + \underline{C}J_C\underline{\dot{\omega}}_C \\ &= \underline{C}\left(\tilde{\omega}_{CA}J_C\underline{\omega}_C + J_C\underline{\dot{\omega}}_C\right)\end{aligned} \tag{9}$$

and substituting (9) into (8) with $\sum\underline{M} = \underline{C}\sum\underline{M}_C$ yields

$$\sum \underline{M}_C = \tilde{\omega}_{CA}J_C\underline{\omega}_C + J_C\underline{\dot{\omega}}_C \tag{10}$$

Finally, the left hand side of (10) can be written as

$$\sum \underline{M}_C = \theta_{DC}\underline{M}_{R,D} - \underline{\tilde{r}}_{QP,C}{\theta_{CB}}^T\underline{F}_{C,B} \tag{11}$$

where

$$\underline{\tilde{r}}_{QP,C} = \begin{bmatrix} 0 & -\underline{r}_{QP,C3} & \underline{r}_{QP,C2} \\ \underline{r}_{QP,C3} & 0 & -\underline{r}_{QP,C1} \\ -\underline{r}_{QP,C2} & \underline{r}_{QP,C1} & 0 \end{bmatrix}$$

As a final note it can be observed from Figure 3 that the only force vector creating a moment on point Q is the contact force vector and the arm for this moment is in opposite direction of $\underline{r}_{QP,C}$. For the final moment balance, the reacting force and moment vector have to be transformed to coordinate system C first.

2.3 Body Equations

Figure 4 shows the forces and moments acting on the body, where $\underline{r}_{RQ,D} = \{0\ 0\ d_{RQ}\}_D$ is the vector going from point Q to R and the components of $\underline{F}_R$ and $\underline{M}_R$ are defined in positive direction on the wheel and thus work in opposite direction on the body. The gravity force vector $\underline{F}_{GB,A} = \{0\ 0\ -m_b g\}_A^T$ acts on point R, the center of gravity for the body and inertia wheel together. The inertia wheel is simply replaced by a moment, $\underline{M}_{I,D} = \{\tau_2\ 0\ 0\}_D$, where the assumption is made that low inertial wheel velocity allows moments caused by quadratic velocity terms to be discarded.

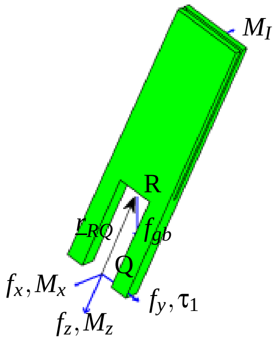


Figure 4: Forces and moments acting on the body.

The angular velocity of the body is equal to that of the wheel

$$\underline{\omega}_{b,C} = \theta_{CB}{}^T \begin{Bmatrix} 0 \\ 0 \\ \dot{\psi} \end{Bmatrix} + \begin{Bmatrix} \dot{\gamma} \\ 0 \\ 0 \end{Bmatrix} + \theta_{EC} \begin{Bmatrix} 0 \\ \dot{\varphi} \\ 0 \end{Bmatrix}$$

however it does not rotate with angle θ. The velocity of point R is equal to the velocity of point Q plus the relative velocity due to rotation

$$\underline{v}_{R,A} = \underline{v}_{Q,A} + \tilde{\omega}_{b,A}\underline{r}_{RQ,A} \tag{12}$$

where $\tilde{\omega}_{b,A}$ and $\underline{r}_{RQ,A}$ in (12) can be expressed by the known equations for $\tilde{\omega}_{b,C}$ and $\underline{r}_{RQ,D}$ given by

$$\tilde{\omega}_{b,A} = \theta_{BA}\theta_{CB}\tilde{\omega}_{b,C}\theta_{CB}{}^T\theta_{BA}{}^T \tag{13}$$

$$\underline{r}_{RQ,A} = \theta_{BA}\theta_{CB}\theta_{DC}\underline{r}_{RQ,D} \tag{14}$$

and substitution of (13) and (14) into (12) yields

$$\underline{v}_{R,A} = \underline{v}_{Q,A} + \theta_{BA}\theta_{CB}\tilde{\omega}_{b,C}\theta_{DC}\underline{r}_{RQ,D} \tag{15}$$

Finally, differentiating the right hand side of (15) yields

$$\begin{aligned} \dot{\underline{v}}_{R,A} = \dot{\underline{v}}_{Q,A} + \theta_{BA}\left(\tilde{\omega}_{BA}\theta_{CB}\tilde{\omega}_{b,C} + \theta_{CB}\tilde{\omega}_{CB}\tilde{\omega}_{b,C} \cdots\right. \\ \left.\cdots + \theta_{CB}\dot{\tilde{\omega}}_{b,C} + \theta_{CB}\tilde{\omega}_{b,C}\tilde{\omega}_{DC}\right)\theta_{DC}\underline{r}_{RQ,D} \end{aligned} \tag{16}$$

that can be used in Newton's second law

$$\sum \underline{F}_A = m_b\dot{\underline{v}}_{R,A} \tag{17}$$

for the translational motion of the body. The left hand side of (17) can be written as

$$\sum \underline{F}_A = -\theta_{BA}\theta_{CB}\theta_{DC}F_{R,D} + \underline{F}_{GB,A} \tag{18}$$

and substitution of (18) into (17) and rearranging leads to an expression for the reaction force vector

$$\underline{F}_{R,D} = \theta_{DC}{}^T\theta_{CB}{}^T\theta_{BA}{}^T\left(\underline{F}_{GB,A} - m_b\dot{\underline{v}}_{R,A}\right) \tag{19}$$

Application of Euler's equation on the rotational motion of the body will result in

$$\sum \underline{M}_D = \tilde{\omega}_{DA}J_{b,D}\underline{\omega}_{b,D} + J_{b,D}\dot{\underline{\omega}}_{b,D}$$

where the left hand side can be written as the sum of all moments

$$\sum \underline{M}_D = -\underline{r}_{RQ,D} \times -\underline{F}_{R,D} - \underline{M}_{R,D} + \underline{M}_{I,D}$$

acting on point R. Rewriting this last expression leads the reaction moment vector

$$\begin{aligned} \underline{M}_{R,D} = & \tilde{r}_{RQ,D}\underline{F}_{R,D} + \underline{M}_{I,D} - \tilde{\omega}_{DA}J_{b,D}\underline{\omega}_{b,D} \cdots \\ & \cdots - J_{b,D}\dot{\underline{\omega}}_{b,D} \end{aligned} \tag{20}$$

where $\tilde{\omega}_{DA}$ and $\underline{\omega}_{b,D}$ in (20) can be written as

$$\tilde{\omega}_{DA} = \theta_{DC}{}^T\left(\theta_{CB}{}^T\tilde{\omega}_{BA}\theta_{CB} + \tilde{\omega}_{CB}\right)\theta_{DC} + \tilde{\omega}_{DC} \tag{21}$$

$$\underline{\omega}_{b,D} = \theta_{DC}{}^T\underline{\omega}_{b,C} \tag{22}$$

3 SPACAR

To validate the model derived in this paper, a second independent model is created using SPACAR (Aarts et al., 2008). SPACAR is based on the nonlinear finite element theory for multi-degree of freedom mechanisms and runs in a Matlab environment and capable of analyzing the dynamics of planar and spatial mechanisms and manipulators with flexible links. The code listed below constructs the model of the MEUR in this paper. The SPACAR model consists of a spatial rigid beam element, a spatial wheel element and four spatial hinge elements defined by the commands `RBEAM`, `WHEEL` and `HINGE`. The first number after the command is the element number. The next two or three numbers are the coordinates, where the numbers 1 to 5 are rotational coordinates and 6 to 8 are translational coordinates. The last three numbers represent the initial orientation.

```
HINGE  1    1  2      0.0 0.0 1.0
HINGE  2    2  3      1.0 0.0 0.0
HINGE  3    3  4      0.0 1.0 0.0
WHEEL  4    6  5  7   0.0 1.0 0.0
HINGE  5    4  5      0.0 1.0 0.0
RBEAM  6    6  4  8   0.0 0.0 1.0

X  6    0.0      0.0  0.0495
X  7    0.0      0.0  0.0
X  8    0.00001  0.0  1.0495

FIX  1
DYNE 1 1
DYNE 2 1
DYNE 3 1
DYNE 5 1
KINX 6 1 2

END
HALT

XM 6 0.1
XM 5 0.000021213 0 0 0.00003 0 0.000021213

XM 8 0.2
XM 4 0.8000 0 0 0.002 0 0.4000

GRAVITY 0 0 -9.81

STARTDE 1 1 0                  0
STARTDE 2 1 0.6                0
STARTDE 3 1 3.741592653589793  0
STARTDE 5 1 0                  0

TIMESTEP 4 400

END
END
```

The initial values are defined in the second block by the command X and assigns the robot dimensions. In the third block the first rotational coordinate is fixed to the world with FIX. DYNE defines the degrees of freedom, being the first deformation of the elements 1, 2, 3 and 5, equal to respectively ψ, γ, θ and ϕ. KINX defines two coordinates where the no-slip condition holds. XM defines the point masses in coordinate 6 and 8 and the inertia's along the x, y and z axis. GRAVITY takes care of the external forces acting on the masses. The third and fourth number of STARTDE defines the initial conditions for ψ, γ, ϕ and θ and the initial conditions for their time derivatives. TIMESTEP defines the simulation time followed by the amount of time steps.

4 SIMULATIONS

To cross validate the model derived in Section 2 and the model provided by SPACAR in Section 3, time domain simulations are carried out using the numerical values listed in Table 3. During the simulation, the trajectory of the center of the driving wheel is chosen as a measure for the cross validation.

Table 3: Numerical values of MEUR parameters.

Wheel			Body		
m_w	0.1	kg	m_b	0.2	kg
R	0.0495	m	d_{rq}	1	m
J_{xx}	0.00002	kgm^2	$J_{b,xx}$	0.8	kgm^2
J_{yy}	0.00003	kgm^2	$J_{b,yy}$	0.002	kgm^2
J_{zz}	0.00002	kgm^2	$J_{b,zz}$	0.4	kgm^2
g	9.81	ms^{-2}			

Different non-zero initial conditions and constant motor torques are used and listed in Table 4. Time simulations are computed using a non-stiff differential equation solver (Cooper, 2004) and implemented via ode45 in Matlab. The initial values under #1 in Table 4 and the definition of the angles in Table 1 are chosen such that the model starts as a stable mechanical system during the simulation. In addition, a small initial roll angle $\gamma = 0.6$ is chosen to demonstrate the coupling effects between the lateral and longitudinal motion of the MEUR.

Table 4: Non-zero initial conditions for the simulations.

	set #1	set #2	
$\gamma(0)$	0.6	0.6	rad
$\varphi(0)$	$\pi + 0.6$	0	rad
τ_1	0	0.1	Nm
τ_2	0	0	Nm

It can be observed from the coinciding simulation results depicted in Figure 5 that the time trajectory $(x(t), y(t))$ of the center of the driving wheel starting in point $(0,0)$ undergoes a periodic oscillation in the x direction due to the initial non-zero body pitch angle $\varphi(0) = \pi + 0.6$. Interesting to see is also the small motion in the y direction of the center of the driving wheel due to the a initial roll angle $\gamma(0) = 0.6$, causing a small change in yaw angle $\psi(t)$. With simulation results of the body pitch angle $\varphi(0) = 0$ and the roll angle $\gamma(0) = 0.6$ depicted in Figure 6, the MEUR model now starts in upward and unstable direction and during the simulation we assume the MEUR can fall and oscillate through the base plane. In addition, a constant torque $\tau_1 = 0.1$ is applied between the driving wheel and body. In Figure 6, both models follow the same trajectories very closely for some time but diverge eventually. This is due to the fact that the model of the mechanical system is unstable at the initial condition, and small numerical errors in either initial conditions or numerical integration leads to exponentially increasing differences in the simulation.

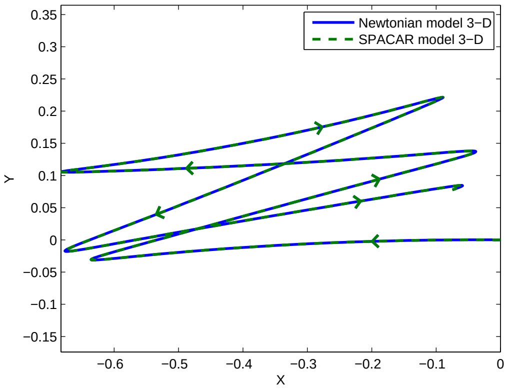


Figure 5: Simulation results of the trajectory of the center of the driving wheel starting in point $(0,0)$ using the parameters of set #1 in Table 4.

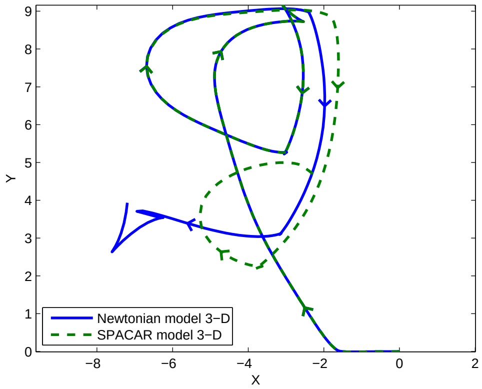


Figure 6: Simulation results of the trajectory of the center of the driving wheel starting in point $(0,0)$ using the parameters of set #2 in Table 4.

5 CONCLUSIONS

This paper presents a three dimensional nonlinear dynamic model for a Moment Exchange Unicycle Robot (MEUR). The model is derived using both Newtonian mechanics and a non-linear finite element package for multi-degree of freedom mechanisms called SPACAR. The simulation results presented in this paper cross validate the Newtonian and the SPACAR model, as simulations of the center of the driving wheel coincide. Differences is simulations attributed to small errors in the in either initial conditions or numerical integration can only be observed in case of an unstable initial condition. In addition, the simulation results demonstrate the coupling between lateral and longitudinal motion the center of the driving wheel. Coupling effects are small only in the case of limited (stabilized) motions of the MEUR.

REFERENCES

Aarts, R., Meijaard, J., and Jonker, J. (2008). *SPACAR User Manual*.

Au, K. and Xu, Y. (1999). Decoupled dynamics and stabilization of single wheel robot. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, volume 1, pages 197–203.

Cooper, J. (2004). *An Introduction to Ordinary Differential Equations*. Cambridge University Press.

Dao, M. and Liu, K. (2005). Gain-scheduled stabilization control of a unicycle robot. *JSME International Journal Series C: Mechanical Systems, Machine Elements and manufacturing*, 48(4):649–656.

Hughes, P. (1986). *Spacecraft Attitude Dynamics*. J. Wiley.

Kolve, D. (1993). Describing an attitude. In *Proc. Annual Rocky Mountain Guidance and Control Conference*, volume 93, pages 289–303.

Majima, S., Kasai, T., and Kadohara, T. (2006). A design of a control method for changing yaw direction of an underacuatted unicycle robot. In *Proc. TENCON 2006, IEEE Region 10 Conference*, pages 1–4.

Nakajima, R., Tsubouchi, T., Yuta, S., and Koyanagi, E. (1997). A development of a new mechanism of an autonomous unicycle. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, volume 2, pages 906–912.

Naveh, Y., Bar-Yoseph, P., and Halevi, Y. (1999). Nonlinear modeling and control of a unicycle. *Dynamics and Control*, 9:279–296.

Ohsaki, H., Kinoshita, M., Sugimoto, Y., Yoshida, K., Iwase, M., and Hatakeyama, S. (2008). Development of a control system for supporting to grow human skill of a unicycle. In *SICE Annual Conference*, pages 2229–2233.

Schoonwinkel, A. (1987). *Design and Test of a Computer Stabilized Unicycle*. PhD thesis, Stanford Univ., CA.

Sheng, Z. and Yamafuji, K. (1997). Postural stability of a human riding a unicycle and its emulationby a robot. *IEEE Transactions on Robotics and Automation*, 13:709–720.

van Pommeren, J. (2007). The unibot, design and control of a self-balancing unicycle robot. Master's thesis, University of Twente.

Vos, D. and Flotow, A. V. (1990). Dynamics and nonlinear adaptive control of an autonomous unicycle: Theory and experiment. *Proc. of the 29th IEEE Conf. on Decision and Control*, 1:182–187.

Zenkov, D., Bloch, A., and Marsden, J. (1999). Stabilization of the unicycle with rider. In *Proc. 38th IEEE Conf. on Decision and Control*, volume 4, pages 3470–3471.

MULTI-SCALE COLLABORATIVE SEARCHING THROUGH SWARMING

Wangyi Liu, Yasser E. Taima, Martin B. Short and Andrea L. Bertozzi
Department of Mathematics, University of California, Los Angeles, CA 90095-1555, U.S.A. [*]
{*bobbyliu, ytaima, mbshort, bertozzi*}*@math.ucla.edu*

Keywords: Mobile robots, Autonomous agents, Searching, Swarming, Target detection.

Abstract: This paper presents a multi-scale searching and target-locating algorithm for a group of agents moving in a swarm and sensing potential targets. The aim of the algorithm is to use these agents to efficiently search for and locate targets with a finite sensing radius in some bounded area. We present an algorithm that both controls agent movement and analyzes sensor signals to determine where targets are located. We use computer simulations to determine the effectiveness of this collaborative searching. We derive some physical scaling properties of the system and compare the results to the data from the simulations.

1 INTRODUCTION

Collaborative sensing has long attracted research interest. Researchers have investigated scenarios where sensors require localization (Bullo and Cortes, 2005), where they are used to control collaborative movement (Bopardikar et al., 2007), detect a scalar field (Gao et al., 2008), or perform a collaborative task (Smith and Bullo, 2009). Using such collaborating sensors to detect and locate targets within an area has been studied in reference to the “mine countermeasure” problem (Cook et al., 2003), the specific military task of locating ground or water-based mines.

In this paper, we develop a multi-scale search and target-locating algorithm for a type of mine countermeasure problem in which a number of independent agents are given the task of determining the precise location of targets within a domain. The algorithm is designed to handle problems where the scale of the target sensing radius is much smaller than the domain size. The focus of this work is to identify optimality of the algorithm as a function of the swarm size, the number of agents per group and the distribution of resources into different groups.

We assume a simply-connected domain, and use noisy sensors that detect a scalar quantity emitted by each target, but only when an agent is within a fixed distance r_s from a target. We control the motion of the agents with a model that makes the individuals form distinct swarms, and present filtering techniques that allow for locating targets despite noisy data. The inspiration for this approach comes in part from biology, as in the example of birds forming flocks when flying and searching for food (Travis, 2007).

Next, we analytically derive some of the system’s main scaling properties, such as the relationship between swarm size, distance between agents and target sensing radius, and compare with the experimentally recorded data. We conclude that our analytical approach matches well the data from the simulations.

We assume a sensing radius r_s much smaller than the domain but comparable to or less than the swarm size. Other assumptions, however, may require different algorithms. For example, in (Burger et al., 2008) the sensing radius is infinite, but sensing is limited by obstacles, and in (Olfati-Saber, 2007) communication between agents is not always possible.

1.1 Scenario Description

We consider M targets in a two-dimensional simply-connected domain, that is, a flat enclosed area with no holes, and N agents able to move freely within the domain. Each target emits a radially symmetric scalar signal $g(r)$ that decays with the distance r from the target, and drops to zero at some r_s, the target’s sensing radius. Agents detect this signal with an additional Gaussian, scalar white noise component added. If an agent is within the sensing radius of multiple targets, it detects only the sum of the individual signals, again with a noise component added. We suppose

[*] This paper is supported by ARO MURI grant 50363-MA-MUR, ONR grant N000141010641 and NSF grant DMS-0914856.

that an agent takes sensor readings at regular intervals (once per "time step") spaced such that the noise between time steps can be assumed independent.

The algorithm accomplishes 3 tasks: it filters the noisy sensor data, controls the coordinated movement of the agents based on this data, and determines when a target has been acquired and where it is located.

1.2 Structure of the Paper

The algorithm is described in the next three sections: Section 2 focuses on the techniques we use to process sensor data, Section 3 describes the movement control of the agents, and Section 4 describes the method for locating a target. The algorithm is evaluated in Section 5. Scaling properties are derived and checked against simulations in Section 6, followed by a conclusion and ideas for future research in Section 7.

2 SENSOR DATA PROCESSING

Due to noise in the agent sensor readings and the sensing radius r_s being finite, we employ two distinct filters to the data from the readings: a Kalman filter and a Cumulative Sum (CUSUM) filter. The Kalman filter reduces or eliminates the noise in the data, while the CUSUM filter is well-suited to determining whether or not an agent is within the sensing radius of a target. The sensor data model follows as a mathematical formula. As explained in Section 1, the formula describes sensor readings as the sum of scalar signals that depend only on the distance from a target, together with a noise component.
Given the M targets at positions y_j and an agent i with current position $x_i(t_k)$ at timestep k, the agent sensor reading $s_i(t_k)$ is given by

$$s_i(t_k) = \sum_{j=1}^{M} g(|y_j - x_i(t_k)|) + n_i(t_k) \ , \tag{1}$$

where $n_i(t_k)$ denotes sensor noise and $g(r)$ is the signal strength at a distance r from the target. For simplicity, we assume $g(r)$ is isotropic, smooth, decaying, the same for all targets, and has a cutoff at r_s.

2.1 Kalman Filtering

Before using the agent sensor readings to locate targets or control agent motion, we pass the sensor readings through a Kalman filter. Since the signal from the target is presumed to be varying smoothly with the distance to the target r (up to the cutoff point r_s) as the agents navigate the environment, a Kalman filter is a natural choice to eliminate or reduce noise in the sensor readings. The Kalman filter takes the sensor reading $s_i(t_k)$ of agent i at time t_k, and converts it into the filtered data $f_i(t_k)$ according to

$$P_i(t_k) = \frac{P_i(t_{k-1})R_i(t_k)}{P_i(t_{k-1}) + R_i(t_k)} + Q_i(t_k) \ , \tag{2}$$

and

$$f_i(t_k) = f_i(t_{k-1}) + \frac{P_i(t_k)(s_i(t_k) - f_i(t_{k-1}))}{P_i(t_k) + R_i(t_k)} \ . \tag{3}$$

Here $R_i(t_k)$ is the square of the noise amplitude, known or estimated by the agent, and $Q_i(t_k)$ is the square of the change of the signal amplitude between two time steps, either fixed beforehand or estimated using the current velocity of the agent (in this paper, it is fixed beforehand). $P_i(t_k)$ is roughly the variance of the sensor reading's amplitude. The output $f_i(t_k)$ of this filter is then used in target-locating, as described in Section 4.

2.2 Threshold Check and the CUSUM Filter

Before attempting to locate targets, an agent needs to determine whether or not it is receiving an actual signal, rather than just noise. In other words, an agent needs to determine whether it is within the sensing radius of a target at each time-step t_k. This information is then used both in controlling the movement of the agents and in determining when to begin estimating a target's position. In order to determine the sensing status of an individual agent, we employ a CUSUM filter, as this type of filter is well-suited to determining abrupt changes of state (Page, 1954), and has been used in the similar task of boundary tracking (Jin and Bertozzi, 2007; Chen et al., 2009). The filter keeps a sort of running average of the signal and notes when this average seems to have risen above a cetain threshold, indicating that the agent is now within the sensing radius of a target. As the noise is effectively summed up by the filter, it tends to cancel out.

In the original form of the CUSUM filter, we imagine a sensor that returns a sequence of independent observations $s(t_1)...s(t_n)$, each of which follows one of two probability density functions: a prechange function g_0 and a post-change function g_1. The log-likelihood ratio is

$$Z(t_k) = \log[g_1(s(t_k))/g_0(s(t_k))] \ , \tag{4}$$

and we define the CUSUM statistic as

$$U(t_k) = \max(0, Z(t_k) + U(t_{k-1})), \ \ U(t_0) = 0 \ . \tag{5}$$

We then choose a threshold $\bar{U}$, and when $U(t_k) \geq \bar{U}$ for the first time, the algorithm ends and we declare that the state has changed from g_0 to g_1. The threshold should be chosen so as to minimize both false-alarms (these happen more frequently for small $\bar{U}$) and time to detection (this gets larger as $\bar{U}$ increases).

In our system, we choose the special case where sensor reading follows a Gaussian distribution. In the pre-change state g_0, the agent is outside the sensing radius of any target and reads only noise, which we model as a Gaussian with zero mean and variance σ^2. In the post-change state g_1, the agent enters the sensing radius of a target, and although the probability distribution is still a Gaussian with the same variance, the mean is now larger than zero, which we set to be $2B$. Then

$$\begin{aligned} Z(t_k) &= \log\left[\frac{e^{-[s(t_k)-2B]^2/2\sigma^2}/(\sigma\sqrt{2\pi})}{e^{-s(t_k)^2/2\sigma^2}/(\sigma\sqrt{2\pi})}\right] \\ &= \frac{-[s(t_k)-2B]^2}{2\sigma^2}+\frac{s(t_k)^2}{2\sigma^2} \\ &= \frac{2B}{\sigma^2}[s(t_k)-B]\ . \end{aligned} \qquad (6)$$

We also modify the algorithm so that it can detect status changes both into and out of detection zones. Thus, we implement two filter values: $U_i(t_k)$ to determine when an agent has entered a zone, and $L_i(t_k)$ to determine if it has left a zone. We also define a binary function $b_i(t_k)$ which denotes the status of an agent: $b_i(t_k) = 1$ means that the agent is near a target and $b_i(t_k) = 0$ means otherwise. The filter values all start at zero, and are thus updated according to

$$U_i(t_k) = \max(0, s_i(t_k) - B + U_i(t_{k-1}))\ , \qquad (7)$$

$$L_i(t_k) = \min(0, s_i(t_k) - B + L_i(t_{k-1}))\ , \qquad (8)$$

and

$$b_i(t_k) = \begin{cases} 1 & b_i(t_{k-1}) = 0,\ U_i(t_k) > \bar{U} \\ 0 & b_i(t_{k-1}) = 1,\ L_i(t_k) < \bar{L} \\ b_i(t_{k-1}) & \text{otherwise.} \end{cases} \qquad (9)$$

In addition, when the status of agent i changes, we reset the corresponding U_i or L_i to zero. Lastly, we have set the constant coefficient $\frac{2B}{\sigma^2} = 1$ for convenience.

Recall that B is a sensor value that is less than the predicted mean when inside a sensing radius, and $\bar{U}$ is our chosen detection threshold. So, when the agent is near a target, the sensor reading $s_i(t_k)$ tends to be larger than B, causing $U_i(t_k)$ to grow quickly until it is larger than $\bar{U}$, indicating a change in status. The converse is true if an agent leaves the sensing region of a target. The values of filter parameters, $\bar{U}$, $\bar{L}$ and B are problem-specific, and should be set in a manner that minimizes false-alarms while keeping the average time to detection as low as possible, as mentioned above.

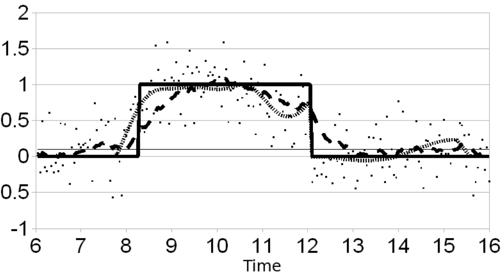


Figure 1: Example filter output for an agent as a function of time, from one of our simulations. The densely-dotted line represents the true signal that ought to be detected by the agent. The dots are the actual noisy signal detected by the agent (i.e., the densely-dotted curve plus noise). The thicker step function is the signal status returned by the CUSUM filter, and the thinner straight line represents the value $B = 0.1$. The sparsely-dashed curve is the output of the Kalman filter when applied to the detected noisy signal.

An example of sensor reading for an agent from one of our simulations is in Figure 1. The Kalman filter does a good job of reducing noise, bringing the sensor readings much closer to the true signal. Near the middle of the plot, the agent enters into the sensing radius of a target; this is reflected by a transition within the CUSUM filter from $b = 0$ to $b = 1$. There is, as expected from the behavior of CUSUM, a slight delay between when the agent actually enters into the radius and when this transition of b occurs. After spending some time within the sensing radius, the estimated target location stabilizes, the agent subtracts the true signal from its measurements (this will be explained in Section 4), and the agent leaves to find further targets.

3 AGENT MOVEMENT CONTROL

We have chosen to control the movement of our agents by breaking up our total agent population N into a number of distinct, leaderless “swarms”. This is done for a variety of reasons. Firstly, it increases robustness, as any individual swarm member is not critical to the functioning of the swarm as a whole. Secondly, since we imagine that any sensor data acquired by readings from the agents is local in space, a swarm provides a method of extending the effective sensing zone to the whole swarm. Thirdly, a swarm of nearby agents may use their combined measurements to decrease sensor noise. Finally, a swarm provides the ability to locate targets via triangulation or gradient methods. Each swarm may search within its designated region of space if a divide-and-conquer tactic

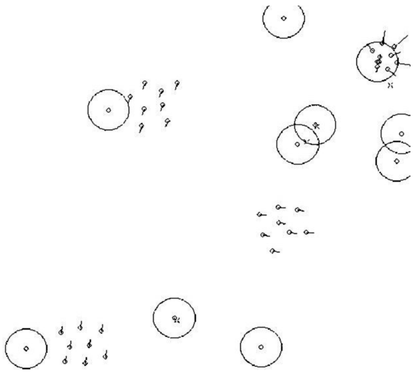


Figure 2: A screenshot from the simulation. Four swarms with eight agents each are used. Three of them are in the searching phase, and the upper right swarm is in the target-locating phase. The large circle around each target marks the sensing radius. Small crosses are already located targets.

is desired, or it may be free to roam over the entire region. In the following two sections we mainly focus on the control of movement for one swarm.

Since the agents detect a limited sensing radius, we choose to employ two different phases of swarm motion. When there are no targets nearby, the agents should move through the space as quickly and as efficiently as possible, performing a simple flocking movement as legs of a random search. After a signal is sensed via the CUSUM filter, the agents should stop, then slowly move around the area, searching for the exact position of the nearby target. We call these two phases the searching phase and the target-locating phase, respectively. For a general idea of the two types of motion, see Figure 2.

3.1 The Swarming Model

There are a variety of mathematical constructs that lead to agent swarming (see for example (Justh and Krishnaprasad, 2004), (Vicsek et al., 1995), and (Sepulchre et al., 2008). Here we choose a second-order control algorithm similar to that described in (D'Orsogna et al., 2006) and (Chuang et al., 2007), which has been successfully implemented as a control algorithm for second-order vehicles on real testbeds (Nguyen et al., 2005; Leung et al., 2007). In this system, each agent of the swarm is subject to self-propulsion and drag, and attractive, repulsive, and velocity alignment forces from each of the other agents. The position x_i and velocity v_i of an individual agent i with mass m_i in a swarm of N agents are governed by

$$\frac{dx_i}{dt} = v_i \ , \tag{10}$$

and

$$m_i \frac{dv_i}{dt} = (\alpha - \beta |v_i|^2) v_i - \nabla U(x_i) + \sum_{j=1}^{N} C_o (v_j - v_i) \ , \tag{11}$$

where

$$U(x_i) = \sum_{j=1}^{N} C_r e^{-|x_i - x_j|/l_r} - C_a e^{-|x_i - x_j|/l_a} \ . \tag{12}$$

C_r and l_r are the strength and characteristic length of the repulsive force, respectively, and C_a and l_a are the corresponding values for the attractive force. C_o is the velocity alignment coefficient, α is the self-propulsion coefficient and β is for drag. Depending on these parameters, the swarm can undergo several complex motions (D'Orsogna et al., 2006), two of which are flocking and milling, and in some cases the swarm can alter its motion spontaneously (Kolpas et al., 2007). For our purposes, we simply set these parameters to obtain the type of motion desired.

3.2 Searching Phase

In this phase, the agents move together in one direction as a uniformly-spaced group travelling with a fixed velocity. Since the agents know nothing yet about the location of targets, a random search is chosen here. Specifically, we use a Lévy flight, which is optimally efficient under random search conditions (Viswanathan et al., 1999), and is the same movement that some birds employ while flocking and searching for food (Travis, 2007). To accomplish this type of search, we simply command the swarm to turn by a random angle after flocking for some random amount of time. For a Lévy flight, the time interval Δt between two turns follows the heavy-tailed distribution

$$P(\Delta t) \sim \Delta t^{-\mu} \ , \tag{13}$$

where μ is a number satisfying $1 < \mu < 3$. The value of μ should be chosen optimally according to the scenario in question, as in (Viswanathan et al., 1999). For destructive searching (where targets, once located, are no longer considered valid targets), μ should be as close to 1 as possible. For non-destructive searching (i.e. located targets remain as valid future targets), the optimal $\mu \sim 2 - 1/[\ln(\lambda/r_s)]^2$, where λ is the mean distance between targets and r_s is the sensing radius.

3.3 Target Locating Phase

When enough agents agree that a target is nearby (see Section 2.2), the target-locating phase begins. This minimum number of agents is set by the swarm consensus parameter p, such that the swarm decides to enter this phase when $p\%$ of the agents or more in the swarm are sensing a target. Once in this phase, we want the agents to move only towards the target, so we remove the velocity alignment force ($C_o = 0$), disable self-propulsion ($\alpha = 0$), and issue a halt command so that all agents begin target-locating with zero velocity. In addition, data from agents within the sensing radius is used to continually estimate the position $\bar{y}$ of the target (see Section 4), and the agents in the swarm then try to move towards it, thus attracting other agents in the swarm not yet in the sensing radius to move closer to the target as well. To make the agents move towards the target, we add another potential in Equation 12,

$$U_c = C_c(x_i - \bar{y})^2/2 \; , \tag{14}$$

where $\bar{y}$ is the estimated position of the target, and C_c is an adjustible parameter. The full control equations in the target-locating phase therefore become Equation 10 and

$$m_i \frac{dv_i}{dt} = -\beta |v_i|^2 v_i - \nabla U(x_i) \; , \tag{15}$$

where

$$U(x_i) = \frac{1}{2} C_c (x_i - \bar{y})^2 + \sum_{j=1}^{N} C_r e^{-|x_i - x_j|/l_r} - C_a e^{-|x_i - x_j|/l_a} \; . \tag{16}$$

To show that this system converges to a stationary swarm centered on the target, we note that the total energy of the target-locating system,

$$E = \frac{1}{2} \sum_{i=1}^{N} m_i |v_i|^2 + \sum_{i=1}^{N} U(x_i) \; , \tag{17}$$

serves as a Lyapunov function, so that the collective tends to minimize it. That is,

$$\dot{E} = -\beta \sum_{i=1}^{N} |v_i|^4 \leq 0 \; . \tag{18}$$

Hence, velocities will eventually reach zero (due to drag) and the swarm members will spatially reorder themselves so as to minimize the potential energy, forming a regular pattern centered at the target position. This stationary state serves as a spiral sink, however, so the swarm tends to oscillate about the target position for some amount of time that depends on the value of C_c, with a high C_c yielding less oscillation. However, since the potential being minimized now includes a term that is effectively attracting all of the agents towards the center of mass, the swarm will be more compact than it was before the target-locating potential was added, so too large of a C_c will make the swarm smaller than desired. In practice, we want C_c just large enough to minimize the oscillations in space without making the swarm get too compact.

4 LOCATING TARGETS

During the target-locating phase of motion, all agents of the swarm that are within the sensing radius keep a common register of all of their positions and signal readings made since entering the radius (see "Threshold Check", Section 2.2, above). The agents then use a least-squares algorithm to give an estimate $\bar{y}$ of where the target is located via

$$\bar{y} = \min_{y} \sum_{k=1}^{N'} \left[g(|y - x(t_k)|) - f(t_k) \right]^2 \; , \tag{19}$$

where N' is the number of sensor readings in the common register.

Solving this least-squares minimization is straightforward, but certain assumptions for workability and precision are needed. It is assumed that the form of $g(r)$ is known by the agents for the algorithm to work. For certain classes of targets and scalar fields, we believe this assumption to be fair. For precisely estimating a target's position, we also assume that only one target is within sensing range, or that target sensing radii do not overlap significantly, so that one target is much closer to the agents than any other target. When the sensing radii are small compared to the average distance between targets, these assumptions should hold true. If, instead they prove to be invalid for the particular system at hand, other methods such as gradient estimation could be used.

If the estimated position of the target stabilizes, it is considered to have been located, and the agents register the position of the target and return to the searching phase: the model signal $g(r)$ from the registered target will be subtracted from further sensor readings so that it is not detected again, a form of destructive searching. We thus modify Equation 1 to read:

$$s_i(t_k) = \sum_{j=1}^{M} g(|y_j - x_i(t_k)|) + n_i(t_k) - \sum_{j=1}^{M'} g(|\bar{y}_j - x_i(t_k)|) \; , \tag{20}$$

where M' is the total number of registered targets. Note that the positions of these targets may or may

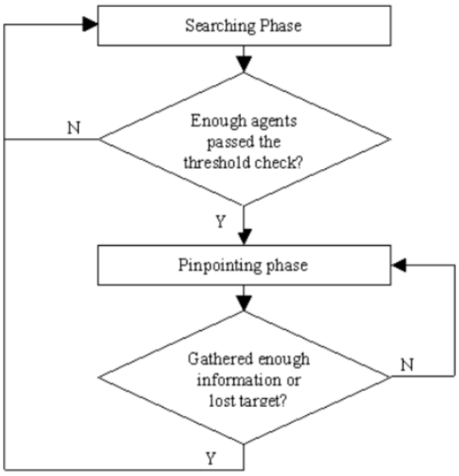

Searching Phase
Enough agents passed the threshold check?
N
Y
Pinpointing phase
Gathered enough information or lost target?
N
Y

Figure 3: A simple flowchart of the algorithm.

not be accurate, due to noise and other errors. If, instead of the estimated target location stabilizing, the agents lose track of the target, they simply return to the searching phase without registering the target. For a general idea of the entire algorithm, see Figure 3.

5 PERFORMANCE EVALUATION

Two main criteria for the evaluation of this algorithm are efficiency and accuracy. These are roughly determined by the two phases: efficiency is mainly related to the searching phase, while accuracy is mainly related to the target-locating phase. To evaluate the performance of the algorithm, we divided the agents into groups and took the following measurements: the average time needed for a swarm to detect and locate a target (Average time), the average distance between the actual and estimated target positions (Average error), and the percentage of registered positions that are not within any actual sensing radius (False registers percentage). Note that false registers are not included in the average error calculation.

We ran computer simulations of the algorithm in a dimensionless 20 by 20 area, with a total of 32 agents and a number of randomly placed targets each with a dimensionless sensing radius of 1. The signals are Gaussian (as previously described in Section 2), with a peak signal-to-noise ratio of about 10.5 dB. Two cases were considered. In the first case, there were 20 targets and we restricted the duration of the simulation, the main goal being that of measuring efficiency. In the second case, we distributed just 5 targets randomly, and used a much longer time limit, with the main goal of measuring accuracy. In both cases, the simulation ends either when time runs out or when all targets are found. For each case, we performed 100 trials and calculated the average of the measurements.

Table 1: Case 1: 20 targets, time limit 50.0. Asterisks denote the use of the divide-and-conquer tactic.

Swarms	Agents/ swarm	Average time	Average error	False reg.
1	32	9.17	0.163	9.77%
2*	16	4.83	0.155	8.40%
2	16	5.45	0.159	11.90%
4*	8	3.15	0.158	8.68%
4	8	3.52	0.16	10.59%
8*	4	2.67	0.208	9.91%
8	4	2.9	0.200	11.73%
16*	2	2.64	0.257	15.59%
16	2	2.64	0.253	15.17%

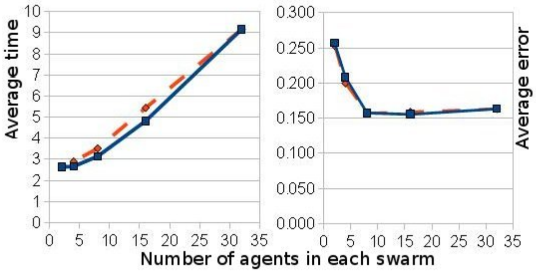


Figure 4: Average detection time (left) and average error (right) as a function of the number of agents in each swarm for case 1 (20 targets and time limit 50.0). The continuous line is for the divide-and-conquer tactic and the dashed line is for the results without divide-and-conquer.

Since we considered multiple groups of agents, it was important to decide how they must cooperate with one another. We tried two different policies. One was a simple divide-and-conquer tactic where we divide the whole region into sub-regions before the simulation, with each swarm in charge of a single subregion (Enright et al., 2005; Hsieh et al., 2006), re-

Table 2: Case 2: 5 targets, time limit 200.0. Asterisks denote the use of the divide-and-conquer tactic.

Swarms	Agents/ swarm	Average time	Average error	False reg.
1	32	45.53	0.128	10.76%
2*	16	25.51	0.116	8.06%
2	16	26.89	0.117	8.95%
4*	8	14.22	0.134	8.96%
4	8	16.64	0.118	8.79%
8*	4	8.35	0.161	7.24%
8	4	10.58	0.172	8.97%
16*	2	8.31	0.223	11.97%
16	2	8.91	0.252	13.79%

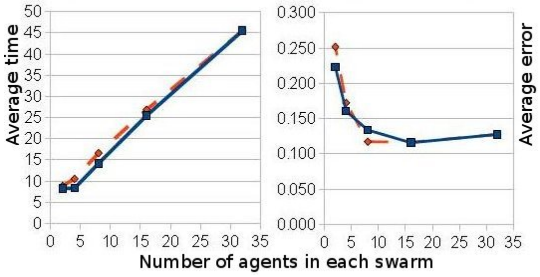


Figure 5: Average detection time (left) and average error (right) as a function of the number of agents in each swarm for case 2 (5 targets and time limit 200.0). The continuous line is for the divide-and-conquer tactic, and the dashed line is the for the results without divide-and-conquer.

maining within that area the entire time, and performing a Lévy flight search pattern confined to its designated area. The other policy allows the swarms to search the entire region independently of one another. In the results, we denoted the use of the divide-and-conquer tactic with an asterisk (*).

An important factor that influenced our results is the number of groups into which we divided the agents, or equivalently, the number of agents in each swarm. We therefore present the results for several choices of this number. They are in Tables 1 and 2, with the associated plots presented in Figs. 4 and 5.

From the tables and their corresponding plots we see that the number of agents in the swarm works as a balance between accuracy and efficiency. As could have been anticipated, larger swarms give more accurate results (smaller target position error, less false registers), while multiple, smaller swarms make the search more efficient (shorter detection time). To have an acceptably low error and low false register percentage, groups of at least four agents should be used. This is perhaps due to the fact that at least three agents are needed to locate a target, using triangulation. Also, we note that the divide-and-conquer tactic seems to work somewhat better for this scenario.

6 SCALING PROPERTIES

Having noted the results above, one may wonder how these are affected by the various scales present in the system, such as the swarm size, distance between agents, target sensing radius, etc. Below we present some arguments for determining optimal search parameters given these scales.

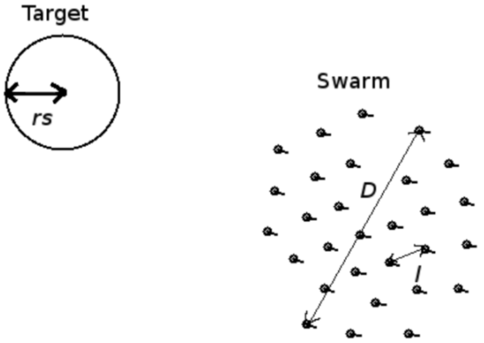


Figure 6: Scales influencing target-locating time: the swarm diameter D, inter-agent length l, and sensing radius r_s.

6.1 Estimating the Swarm Diameter

We first define a measure for the swarm size, the swarm diameter $D = \max(|x_i - x_j|)_{i=1}^{N}$, where N is the number of agents in the swarm. Let us also define the inter-agent distance $l = |x_i - x_j|$ for any two nearest-neighbor agents i and j (see Figure 6).

For the remainder of this section (and for the results in Figure 7), we choose the parameters of motion so that the system is either in regime VI (catastrophic) or VII (H-stable) as defined in (D'Orsogna et al., 2006), with the swarms flocking naturally in VII, and in VI due to the velocity alignment term C_o in Equation 11. Under these regimes, D and l stabilize after a transient period, so for the purposes of this section we will consider them to be constant in time. In such a stable swarm, agents are uniformly distributed in space, so that the swarm diameter D and inter-agent length l are related geometrically as follows: since the area occupied by a single agent in the swarm is $A_a \approx \pi l^2/4$ and the total swarm area is $A_s = \pi D^2/4 \approx NA_a$, then

$$D \simeq \sqrt{N} l \,. \tag{21}$$

Thus, D scales with l, and for $N = 16$ (as used in Figure 7) we get $D \simeq 4l$. Since l is approximately the distance which minimizes the inter-agent potential of Equation 12, we can easily adjust the swarm diameter D by varying the system parameters.

6.2 An Upper Bound on the Optimal Swarm Diameter

Consider a setting with one target of sensing radius r_s and one swarm of diameter D, as in Figure 6. We measure the average time to locate the target $\bar{T}$ by starting the swarm at the center of the search field at time t_0, placing the target at a random location within the field, allowing the simulation to run until time T

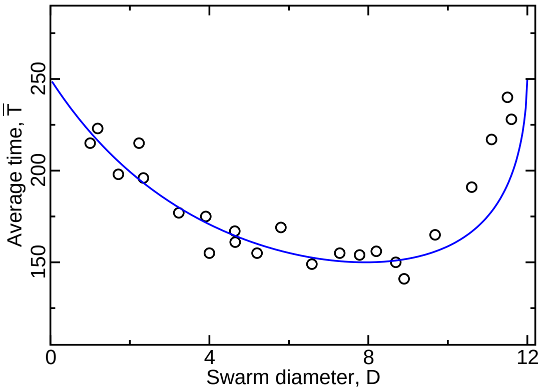


Figure 7: Average time to reach a target $\bar{T}$ as a function of the swarm diameter D for $r_s = 3.0$. Values of D were obtained over a range $C_r = 2.0 - 12.0$, $l_r = 0.2 - 0.7$, with $C_a = l_a = 1.0$. Averaging was carried out over 200 simulated trials, yielding the numerical results (circles); the theoretical results of Equation 26 with the best fitting τ are shown as a line. The number of agents $N = 16$, and $D_{\text{opt}} \approx 8$.

when the target is found, and averaging these T values over many runs of the simulation. As D grows, we observe (see Figure 7) that $\bar{T}$ decreases until an optimal swarm diameter D_{opt} is reached, after which $\bar{T}$ increases again, growing without bound. We wish to explain this, first by finding an upper bound on D_{opt}.

Let us begin by fixing the swarm consensus percentage at $p = 25\%$; i.e., the swarm of agents decides a target is present when 1/4 of the agents or more are within the sensing area of a target, $A_t = \pi r_s^2$ (see Figure 8). Clearly, the borderline case between detection and non-detection occurs when the target area is completely subsumed within the swarm area, yet there are only just enough agents (25% of the total) within the target sensing radius to detect it. If we assume a constant density of agents in the swarm, this means that we have detection when the target area is at least $p\%$ = 1/4 of the swarm area. So, it must be the case that

$$D \leq 4r_s \; , \tag{22}$$

or else the target will not be detected at all. This condition therefore gives an upper bound on D_{opt}. Condition (22) can also be written in terms of l, so that $l \leq r_s$; the borderline case is illustrated in Figure 8(b). In Figure 9, snapshots from a simulation show how the swarm flies over the target without being able to detect it in a case when condition (22) is violated.

6.3 An Approximation for the Optimal Swarm Diameter

Now that we have an upper bound on D_{opt}, we assume that condition (22) is met and look for approximate expressions for D_{opt} and $\bar{T}$. We note that the area of overlap A_o between the target and swarm areas, when the centers are separated by a distance d, is given by

$$A_o = r_s^2 \arccos\left[\frac{z}{r_s}\right] + \frac{D^2}{4}\arccos\left[\frac{2(d-z)}{D}\right] - z\sqrt{r_s^2 - z^2} - (d-z)\sqrt{D^2/4 - (d-z)^2} \; , \tag{23}$$

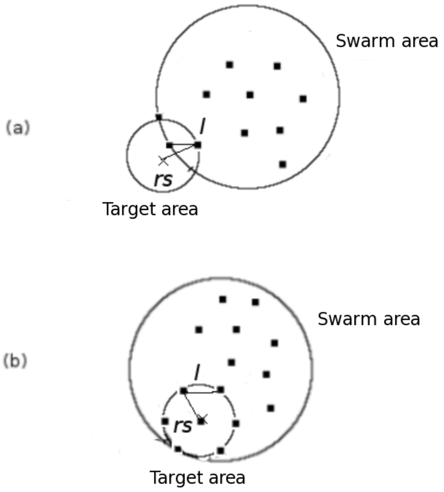


Figure 8: Sensing configurations for the case when 4 or more agents are required to sense the target before it can be detected. (a) Though there is overlap between the swarm and target, too few agents can sense the target for it to be detected. (b) The largest inter-agent distance l while still allowing for detection, $l = r_s$.

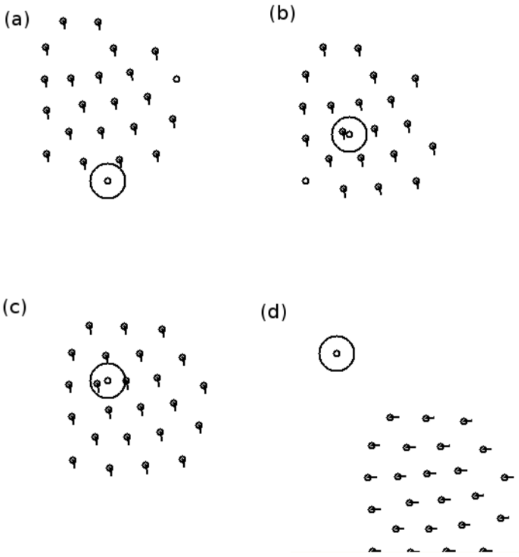


Figure 9: When r_s is too small compared to D, the swarm does not detect the target. Snapshots (a)-(d) show the swarm flying over the target without locating it. Here $N = 24$, $r_s = 1.5$, required percentage for consensus is p = 25%, and D stabilizes at ≈ 13.5.

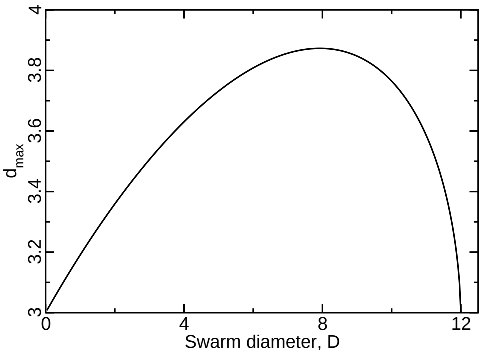


Figure 10: Parameter $d_{\max}$ versus D for $r_s = 3.0$. Note that $d_{\max}$ increases at first, as the growing size of the swarm allows it to be further away while still easily satisfying (22). However, after the peak at $D_{\text{opt}} \approx 8$, condition (22) becomes the limiting factor, requiring greater overlap between the two areas for detection to occur.

where

$$z \equiv \frac{r_s^2 - D^2/4 + d^2}{2d} . \qquad (24)$$

Equation 23 is valid for $|r_s - D/2| \leq d \leq r_s + D/2$. Now, as above, we require that A_o be at least equal to 25% of the swarm area in order for the target to be detected. Thus, we obtain an implicit equation for the maximum separation $d_{\max}$ between the center of the swarm and the target location such that the target is detected:

$$A_o(r_s, D, d_{\max}) = \pi D^2/16 . \qquad (25)$$

The parameter $d_{\max}$ will depend therefore upon r_s and D. At least in terms of the time spent within the searching phase of the algorithm, the shortest time $\bar{T}$ until detection ought to occur when, for a given r_s, D is chosen such that $d_{\max}$ is maximized (see Figure 10), giving the largest effective target size to hit; hence this D should be D_{opt}. Furthermore, we expect a scaling law such that the time to detection is roughly given by

$$\bar{T} \approx \tau \left[\frac{A_{\text{field}}}{\pi d_{\max}^2} - 1 \right] , \qquad (26)$$

where τ is a characteristic timescale and A_{field} is the total area of the search field.

We have experimentally verified this scaling, with experimental results usually quite close to the theoretical values, as illustrated by the example with $r_s = 3.0$ in Figure 7. We note that the actual time to detection is a bit above the theory for $D > D_{\text{opt}}$, presumably due to our assumption of constant density in deriving Equation 25; that is, (especially for large D) the area of overlap between target and swarm may be sufficient, but still not contain at least 25% of the agents, causing the time to detection to be above that expected.

7 CONCLUSIONS

We considered a mine counter-measure scenario using multiple agents that move cooperatively via swarming. The agents use a variety of signal filters to determine when they are within sensing range of a target and to reduce noise for more accurate control and target position estimation. We explored the parameter space through simulations, determining optimal values for some of the search parameters. We derived scaling properties of the system, compared the results with data from the simulations, and found a good analytical-experimental fit.

There are many openings for future research. First, we could use alternative methods in some parts of the algorithm. A potential change is to use a compressed-sensing method (Cai et al., 2008) instead of least-squares for estimating a target's location, which would enable us to find multiple overlapping targets at the same time. Another interesting modification would be to use an anisotropic Lévy search, and take previously covered paths into account. Different scenarios could also be evaluated, which might lead to different results for accuracy and efficiency, or even suggest new algorithms. For example, we could extend the two dimensional problem to 3-D, as would be the case for underwater targets. Or, perhaps the model for the detected signal is unknown, in which case we would employ a different method to estimate the target positions. Finally, apart from numerical simulations, we plan to do experiments on a real testbed, with small robotic vehicles as agents. This would provide an evaluation of the algorithm in the presence of real sensor noise, which may not be entirely Gaussian in nature.

ACKNOWLEDGEMENTS

The authors would like to thank Alex Chen and Alexander Tartakovsky, whose suggestions on the use of the CUSUM filter were very helpful.

REFERENCES

Bopardikar, S. D., Bullo, F., and Hespanha, J. P. (2007). A cooperative homicidal chauffeur game. In *IEEE Conference on Decision and Control*.

Bullo, F. and Cortes, J. (2005). Adaptive and distributed coordination algorithms for mobile sensing networks. In *Lecture Notes in Control and Information Sciences*. Springer Verlag.

Burger, M., Landa, Y., Tanushev, N., and Tsai, R. (2008). Discovering point sources in unknown environments.

In *The 8th International Workshop on the Algorithmic Foundations of Robotics*.

Cai, J., Osher, S., and Shen, Z. (2008). Linearized Bregman iterations for compressed sensing. In *UCLA CAM report 08-06*.

Chen, A., Wittman, T., Tartakovsky, A., and Bertozzi, A. (2009). Image segmentation through efficient boundary sampling. In *8th international conference on Sampling Theory and Applications*.

Chuang, Y., Huang, Y., D'Orsogna, M., and Bertozzi, A. (2007). Multi-vehicle flocking: Scalability of cooperative control algorithms using pairwise potentials. In *IEEE International Conference on Robotics and Automation*.

Cook, B., Marthaler, D., Topaz, C., Bertozzi, A., and Kemp, M. (2003). Fractional bandwidth reacquisition algorithms for vsw-mcm. In *Multi-Robot Systems: From Swarms to Intelligent Automata*, volume 2, pages 77–86. Kluwer Academic Publishers, Dordrecht.

D'Orsogna, M., Chuang, Y., Bertozzi, A., and Chayes, L. (2006). Self-propelled particles with soft-core interactions: Patterns, stability and collapse. In *Physical Review Letters*, 96(10):104302.

Enright, J., Frazzoli, E., Savla, K. and Bullo, F. (2005). On Multiple UAV Routing with Stochastic Targets: Performance Bounds and Algorithms. In *Proceedings of the AIAA Conference on Guidance, Navigation, and Control*.

Gao, C., Cortes, J., and Bullo, F. (2008). Notes on averaging over acyclic digraphs and discrete coverage control. In *IEEE Conference on Decision and Control*.

Hsieh, C., Chuang, Y.-L., Huang, Y., Leung, K., Bertozzi, A., and Frazzoli, E. (2006). An Economical Micro-Car Testbed for Validation of Cooperative Control Strategies. In *Proceedings of the 2006 American Control Conference*, pages 1446–1451.

Jin, Z. and Bertozzi, A. (2007). Environmental boundary tracking and estimation using multiple autonomous vehicles. In *Proceedings of the 46th IEEE Conference on Decision and Control*.

Justh, E. and Krishnaprasad, P. (2004). Equilibria and steering laws for planar formations. In *Systems & Control Letters*, 52(1):25–38.

Leung, K., Hsieh, C., Huang, Y., Joshi, A., Voroninski, V., and Bertozzi, A. (2007). A second generation micro-vehicle testbed for cooperative control and sensing strategie. In *Proceedings of the 2007 American Control Conference*.

Kolpas, A., Moehlis, J., and Kevrekidis, I. (2007). Coarse-grained analysis of stochasticity-induced switching between collective motion states. In *Proceedings of the National Academy of Sciences USA*, 104:5931–5935.

Nguyen, B., Yao-Ling, C., Tung, D., Chung, H., Zhipu, J., Ling, S., Marthaler, D., Bertozzi, A., and Murray, R. (2005). Virtual attractive-repulsive potentials for cooperative control of second order dynamic vehicles on the caltech mvwt. In *Proceedings of the American Control Conference*.

Olfati-Saber, R. (2007). Distributed tracking for mobile sensor networks with information-driven mobility. In *Proceedings of the 2007 American Control Conference*.

Page, E. (1954). Continuous inspection schemes. In *Biometrika*, 41(1-2):100–115.

Sepulchre, R., Paley, D., and Leonard, N. (2008). Stabilization of planar collective motion with limited communication. In *IEEE Transactions on Automatic Control*.

Smith, S. L. and Bullo, F. (2009). The dynamic team forming problem: Throughput and delay for unbiased policies. In *Systems & Control Letters*.

Travis, J. (2007). Do wandering albatrosses care about math? In *Science*, 318:742–743.

Vicsek, T., Czirok, A., Ben-Jacob, E., Cohen, I., and Shochet, O. (1995). Novel type of phase transition in a system of self-driven particles. In *Physical Review Letters*, 75:1226–1229.

Viswanathan, G., Buldyrev, S., Havlin, S., da Luzk, M., Raposok, E., and Stanley, E. (1999). Optimizing the success of random searches. In *Nature*, 401(6756):911–914.

LOCALIZATION IN AN AUTONOMOUS UNDERWATER MULTI-ROBOT SYSTEM DESIGNED FOR COASTAL AREA MONITORING

Zhongliang Hu, Eemeli Aro, Tapani Stipa†, Mika Vainio and Aarne Halme
Finnish Center of Excellence in Generic Intelligent Machines Research, Aalto University School of Science and Technology
P.O. Box 15500, FI-00076 Aalto, Finland
†Marine Hydrodynamics, Finnish Meteorological Institute, Erik Palmenin aukio 1, P.O. Box 50, FI-00101, Helsinki, Finland
{zhongliang.hu, eemeli.aro, mika.vainio, aarne.halme}@tkk.fi, tapani.stipa@fmi.fi

Keywords: Multi-robotic system, Autonomous underwater localization, Underwater acoustic communication, Coastal area monitoring.

Abstract: In this paper we present an multi-robotic system for underwater exploration, specifically for coastal seas. The novelty of this system is its enhanced performance in underwater localization using underwater acoustic ranging and data transfer between the floats. This process is not dependent on any fixed infrastructure, which is usually a requirement for such missions. An algorithm is implemented in the floats for underwater localization, based on an Extended Kalman Filter. Extensive real scenario simulation test results are presented.

1 INTRODUCTION

The value of services provided by the coastal seas, including estuaries, to the human welfare is estimated to be higher than those of terrestrial or open ocean systems (Costanza et al., 1998). The relevant spatial and seasonal scales for biological variability are often related to hydrophysical events; these are mostly unpredictable and practically impossible to cover by traditional monitoring with sparse sampling.

Autonomous underwater multi-probe system for coastal area / shallow water monitoring (SWARM) was an EU-funded (FP5, 2003-2005) project aiming to design, implement and test a multi-robot system that could measure local and transient biological and physical variability in the Baltic Sea and similar areas at the scale relevant for single events.

Based on the initial ideas and hardware designed and built during the SWARM project, the Finnish Center of Excellence in Generic Intelligent Machines (GIM) has continued the related research. As one of the results, a localization method has been developed that requires no additional infrastructure, fixed beacons or expensive inertial units, while producing accurate real-time location estimates for the members of the system. The method is based solely on underwater inter-robot acoustic communication and occasional GPS fixes while on the surface. It is worth mentioning that the internal intelligence which is essential for the units to navigate and survive in the dynamic and hostile underwater environment makes the units stand out from mere buoyancy floats to intelligent underwater robots.

In this paper, the main features of this system working in an extremely challenging environment will be first briefly introduced. Then, the main topic of this paper, i.e., a novel localization approach will be presented and verified with suitable tests. Finally, some conclusions are drawn and some information regarding the future work will be presented.

2 SYSTEM DESCRIPTION

The scientific objective of the SWARM project was to design, implement and test a novel highly redundant underwater monitoring system for shallow water areas (Vainio et al., 2004). The system consists of multiple homogenous, robust and easy to use underwater robots (autonomous intelligent profiling floats) that can perform two week missions autonomously. The floats control their buoyancy but move otherwise freely with the water flows. They communicate with the control station via Iridium satellite communication and use inter-robot acoustic ranging and communication for localization and data exchange. In addition to measuring the standard variables (pressure, temperature, conductivity), the system observes certain algal groups with a novel fluorometer.

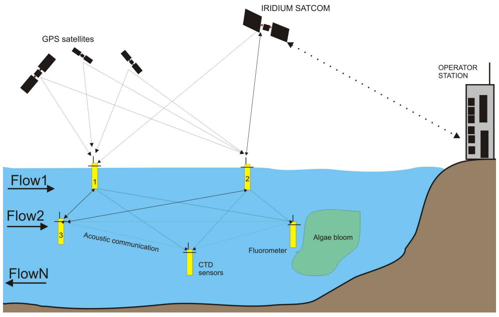


Figure 1: The general description of the SWARM system.

The diving profile of the floats was selected based on two main issues: to enable the system to measure the key environmental parameters from the main areas of interest and to support the proposed localization method. It also matches the standard profile used by Argo floats globally, but of course with shallower depths (Davis et al., 2001).

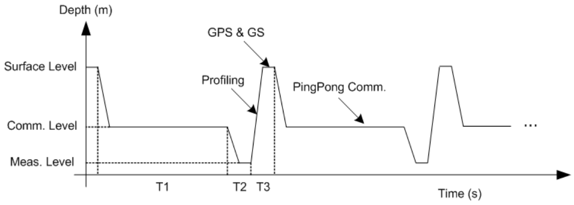


Figure 2: Diving profile of a unit.

T1: Waiting time. The float spends most of its time waiting on this level. The benefit of this is such that the float does not need to stay on the surface all the time but by taking advantage of communication it can still keep a track of its location during the mission.

T2: Diving time. When T1 expires, the float will start to dive to a deeper level and wait there for the start of profiling.

T3: Profiling time. After T2 expires, the floats ascents to surface, during which time it takes CTD(conductivity, temperature and density) profile. When the float reaches the surface, it gets a GPS fix and contact the operator station for possible commands and reports data.

3 EXPERIMENTAL SETUP

During the EU-funded phase, the functionality of each of the subsystems were tested separately before final integration into a prototype. Major part of these tests were carried out in Gulf of Finland on the south coast of Finland. In addition to the real system, a MATLAB based simulator was designed at an early phase of the project mainly for the testing of localization algorithms. After the end of the EU funding a more sophisticated simulator was constructed and used for the testing presented in this paper.

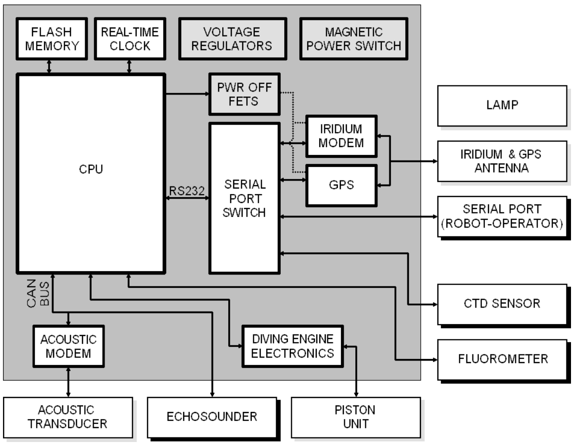


Figure 3: Subsystems shown in a block diagram.

A simplified block diagram shown in Figure 3 illustrates that the developed robotic prototypes are full-blooded autonomous profiling floats which are equipped with extensive repertoire of environmental sensors. Furthermore, they possess certain level of intelligence, which is required to survive autonomously in a dynamic and hostile coastal waters.

3.1 Physical Realization

The main physical output of the SWARM project was four 3^{rd} generation functional floats ready for open sea testing. These floats have since been extensively modified and tested in the lab. Full-scale tests are expected to start from summer 2010. See Figure 4.

3.2 Simulator

A new simulator is implemented as an attempt to produce a near-realistic simulated environment for autonomous floats. The simulator uses a server-client architecture, with the environment as a server to which each float connects. Specialized clients may also connect to the environment to act as ground stations or data display/logging agents. Time in the simulator advances at a multiple of real time. The interfaces available to a program acting as a simulated float are modeled to match those of an actual float, including simulated sensor errors and measurement

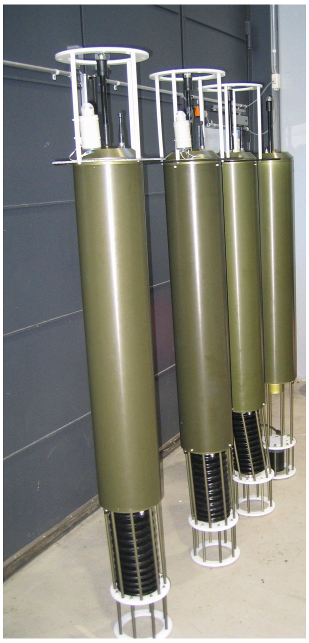


Figure 4: Available physical floats (own by SWARM consortium). Floats are around 2 m tall and weigh approximately 40 kg.

time delays. This allows for hardware-in-the-loop testing using the real float hardware, with each sensor connected using a serial cable to an external computer running an interface program connected to the simulated environment.

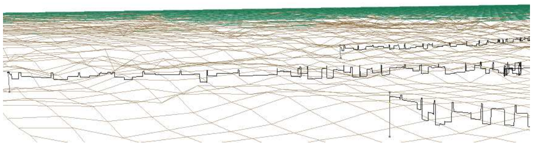


Figure 5: Snapshot of the developed simulator.

The simulator components use GIMnet (Saarinen et al., 2007) to communicate, allowing each program to run on a separate networked computer if required. The architecture is designed to scale, and can simultaneously model hundreds of floats at a simulation speed far in excess of real time. The simulator can make use of data from anywhere in the world, provided that bathymetry and three-dimensional time-variant salinity, temperature and current vector information is available to some precision; intermediary values will be interpolated. The data used for the tests presented here is from the BalEco ecosystem model of the Finnish Institute of Marine Research (currently Finnish Meteorological Institute).

4 FLOAT LOCALIZATION

The core element of a SWARM unit positioning system is a GPS receiver, which is included in each of the robots. When on the surface, a unit gets its position from the satellites. The frequency of these GPS fixes must be considered carefully. They will naturally always bring much needed fresh and accurate information to the system, but on the other hand, they force the units to come to the surface which consumes energy and increase the possibility of being hit by a passing boat. In any case, after the GPS fix, the inter-robot acoustic communication system between robots is then utilized in positioning the underwater units.

It is essential to relate the measurement data that the floats have gathered underwater with the geographical coordinates where they have been measured. A conventional float profiling operation, as shown in Figure 2, gives reasonable results for a series of measurement when the float comes back to the surface from the deep depth, since such surfacing happens in a quite short time and the currents affecting this movement are negligible. However, this movement only gives a vertical series measurements at a given location. We are more interested in the sea conditions in a larger area, including prevailing sea currents. One approach for this can be found in (McFarland and Honary, 2003).

4.1 Baltic Sea

Shallow seas pose specific problems for robotic ocean instruments. They are often strongly stratified, turbid and have vibrant maritime activities. The bottom is soft, to the extent that it may have adhesive powers on instruments that land on the bottom. Figure 6 shows an example of the complex vertical structure of temperature, salinity and fluorescence, with almost unnoticeable physical features that nevertheless have their impact on the biological situation.

The spatial scales of motion are often small in the horizontal direction as well. Therefore, the state of motion in one place is not a very good predictor of the state some kilometers away. These factors make it harder to observe shallow seas with robotic instruments than the ocean. There should be a large number of robotic instruments to capture the spatial variability, yet the loss rate of robotic instruments in shallow seas is expected to be significantly larger than in the blue oceans, where a typical Argo float may survive 3-5 years.

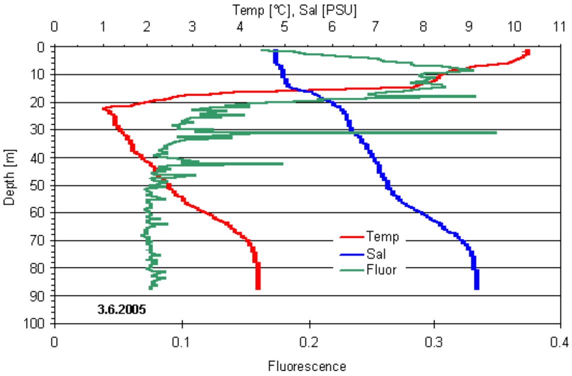


Figure 6: Complex vertical structure of temperature, salinity and fluorescence against depth in the Baltic Sea.

4.2 Acoustic Communication

The range and reliability of underwater acoustic communication is affected by a number of factors, including signal attenuation, noise, reverberation and sound channeling. A transmitted signal will be reflected from the surface or seabed and interfere with the direct path signal. Variations in the temperature and salinity at different depths will result in variations in the speed of sound, which will influence the path of the signal. The sound speed profile for the Baltic Sea is in fact conducive to the formation of a sound channel, a depth at which acoustic signals will reflect internally due to a minimum in the speed of sound. This channel is present mostly during the summer, and varies in depth from 20m to 80m. Using this sound channel, floats in the Baltic Sea can theoretically achieve communications and ranging over distances of up to 10km. In addition to communication, acoustic transmissions may also be used for ranging, by tracking the time of flight of messages. The SWARM floats implement a ping-pong type scheme for measuring the round-trip travel time between floats.

In this paper we use a simple linear model of the probability of construction a communication between drifters. When two floats are at immediate vicinity they have a near guaranteed communication probability, whereas at around 6 kilometers distance, the probability of having a communication channel between two floats falls almost to zero. To determine the position of one float from another, a ranging signal is used. In effect, the float transmits a message and waits for the reply from the receiving float.

A unit that has been on the surface sends its newly obtained GPS position information through acoustic modem once it has reached the defined communication depth and it is its time to communicate. Knowing their own estimated positions and past trajectories, the underwater units can reduce the uncertainty of their position based on this information. A positional accuracy in the scale of hundreds of meters is acceptable in this application. The biggest advantage of this positioning system is the fact, that there is no need for any additional positioning devices, e.g. fixed acoustic buoys used in (Akyildiz et al., 2006; Dario et al., 2005).

4.3 Algorithms and Implementations

The Kalman filter is an efficient and robust recursive filter that estimates the internal state of a dynamic system from a series of noisy measurements. It can also estimate the variables which can't be directly measured (Welch and Bishop, 1995). In our case, this corresponds to the underwater sea current.

For this paper, the underwater localization estimation is based on an Extended Kalman Filter(EKF). The state variables are geographical positions of the units and the vectors describing the currents in the sea in different depth layers, while GPS fixes and distances between the floats through an acoustic ranging being the measurements. Note that in this particular case, we do not include the depth coordinate into the estimation as well as the vertical part of the sea current vector because the measurement of the depth itself is accurate enough. Since we can have a relatively good measurement of the depth from the CTD sensor, and taking the system propulsion into account is not beneficial. This arrangement will also reduce the complexity of the computation which is a limiting factor in the real system onboard.

First we start with defining the state and measurement vectors. The current vector at level j at position (x,y) at k time step of EKF:

$$F_j(x,y)(k) = [f_{jx}(k) \quad f_{jy}(k)]^T \tag{1}$$

In Section 4.1 we have described the basic situation for the Baltic Sea current and its unique character of being layered. Thus, the environment can be thought as series of layers that each have a flow vector representing the sea currents, which slowly varies from one location to another. The units are located at these layers as shown in Figure 7.

To better estimate such phenomena, we divided the sea current vectors into three layers. In addition to limit the computation complexity, we also choose this number to correspond to the diving profile as shown in Figure 2. The flow vector at position (x,y) at k time step of EKF is defined as:

$$F_C(x,y)(k) = [F_1(x,y)(k) \;\; F_2(x,y)(k) \;\; F_3(x,y)(k)]^T \tag{2}$$

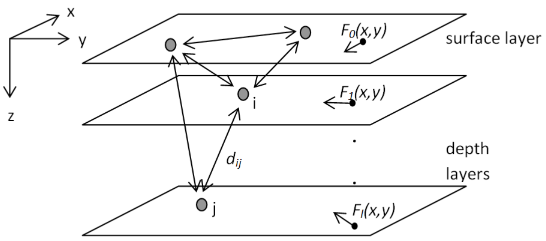


Figure 7: The units are located at layers that have different flow vectors.

Position of a float at position(x,y) at k time step of EKF:

$$P_j(x,y)(k) = [x_j(k) \;\; y_j(k)]^T \tag{3}$$

In this project, we set EKF update rate to $\Delta t = 5minutes$. Considering the real data collected by Finnish Marine Research Institute, we can assume the current vector is changing slowly both with respect to horizontal location and time. The current dynamics is modeled as:

$$F_C(k) = F_C(k-1) + v(k) \tag{4}$$

where $v(k)$ is the system error of the current.

Let's define the extended state Vector of EKF at time step k:

$$\tilde{X}(k) = [P_1(k) \;\; P_2(k) \;\; \ldots \;\; P_n(k) \;\; F_C(k)]^T \tag{5}$$

The corresponding extended state equation:

$$\begin{aligned}\tilde{X}(k) &= A(k)\tilde{X}(k-1) \\ &= \begin{bmatrix} I & \Delta T \\ 0 & I \end{bmatrix} \begin{bmatrix} P(k-1) \\ F_C(k) \end{bmatrix}\end{aligned} \tag{6}$$

Here, system matrix A is updated each time according to the depth layers (DL) of respective floats. In other words, the prior prediction of the float location is updated only by the current vector in its present layer.

The example ΔT Matrix for 3 floats, while float 1 in depth 3, float 2 in depth 1, float 3 in depth 2:

$$\Delta T = \begin{bmatrix} 0 & 0 & \Delta t \\ \Delta & t & 0 \\ 0 & \Delta t & 0 \end{bmatrix} \tag{7}$$

All the floats maintain a depth list of all the known floats. Since the measurements that these floats get are different, even having the same equations running internally, they have different estimations of the state vector. Since the floats spends most of its time in the waiting layer, when there is no depth update from the others, they are assumed to be in the waiting layer (check Figure 7).

Measurement equation:

$$\begin{aligned} Z(k) = [&P_{GPS1}(k) \;\; P_{GPS2}(k) \;\; \ldots \;\; P_{GPSn}(k) \\ &d_{12}(k) \;\; d_{13}(k) \;\; \ldots \;\; d_{(n-1)n}(k)]^T \end{aligned} \tag{8}$$

Expected measurement:

$$\begin{aligned} h(\tilde{X}(k-1)) = [&P_1(k) \;\; P_2(k) \;\; \ldots \;\; P_n(k) \\ &\|P_1(k)-P_2(k)\| \;\; \|P_1(k)-P_3(k)\| \\ &\ldots \;\; \|P_n(k)-P_{n-1}(k)\| \;\;]^T \end{aligned} \tag{9}$$

To linearize h with respect to X for each update step:

$$H(k) \;\; = \;\; \partial h(\tilde{X}(k-1))/\partial \tilde{X}(k-1) \tag{10}$$

The overall running EKF for k time step:

$$\tilde{X}(k) = \tilde{X}^-(k) + K_k(Z(k) - h(\tilde{X}^-(k))) \tag{11}$$

where the Kalman gain k is calculated in a standard EKF formula.

Note that in reality, it is rarely the case that the float will get all the distances from all the other floats and the distances between them. And it is only occasionally when the float has a GPS fix on the surface and gets GPS information from the others. In such case the EKF automatically assumes those measurements are missing and therefore not take them into account while updating the state variables.

5 TESTS AND RESULTS

The simulated environment contains the flow vectors and other environmental data that affect the positions of the floats. The actual positions of the units are maintained by the simulator, but they may be measured by the floats only in a way that is similar to the real world, i.e. only the units that are on the surface know their positions, and distances between floats are only measured in seconds of sound travel time. The user can control whether the units know only the distances from the unit itself to other units, or if they know all the distances between all pairs of units. Every unit contains a Kalman Filter that estimates the positions of the units. The estimation is based on the model that utilizes both the measured distances between the units and the measured positions of the units while on the surface. The real and the estimated positions can be visually tracked and compared in real time. The following figures are the comparison result for a random chosen float in the system during their respective missions.

5.1 Turning vs. Line Traveling

In an area where the current vectors change quickly, a float will typically make sharp turns. In such circumstances, its position estimates can be quite inaccurate. On the other hand, in a case where the currents are more homogenous, the float will follow a approximately straight line and its estimation is obviously better as time passes.

5.2 Test Configuration

The simulation test area is selected to be a bay area not far from the Finnish coast. The key parameters are set as shown in Table 1.

Table 1: Key parameters setting for Simulation.

Acoustic comm. interval	once/hour
Surfacing interval	4 times/day
Surfacing time	10 minutes/dive
Comm. level waiting time	5 hours/dive
Meas. depth waiting time	40 minutes/dive
Kalman filter update interval	once/5 minutes

These parameters are consist with the future real test parameters.

5.3 Baseline Estimation

As a baseline for estimating the quality of the localization method, we use an alternative position estimate which makes no use of acoustic ranging or communication, and instead assumes the current position to match the latest GPS position fix. See Figure 8 for demonstration.

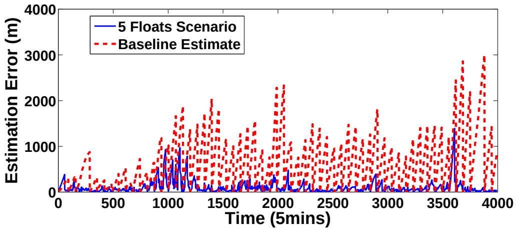


Figure 8: Estimation error for 5 floats randomly deployed within 50m radius area with the the same diving profile. Error is in Euclidian distance between the estimated and real position.

5.4 Data Information Exchange

High bandwidth acoustic data exchange between floats in the Baltic Sea is extremely difficult. It is especially true when the floats are not always staying in the communication layer, which does not even exist in some cases. In Figure 9, we show the results from both only acoustic communication enabled scenario and acoustic & data exchange enabled case. Both of them follow the communication probability model explained in Section 4.1.

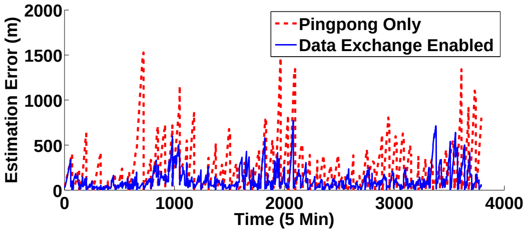


Figure 9: Estimation error comparison between acoustic ranging only and data exchange scenario. (5 units that deployed in a 50m radius area with same diving profile).

5.5 Deployment Strategy

To deployment pattern of the float is an important factor that has a considerable effect on the outcome of the whole mission. When the drifters are deployed within a small perimeter (radius 50*m*), for example from an anchored boat, the high probability of exchanging information will increase the possibility to have accurate position estimates. It is especially important for the drifters at their initialization phase. Since they will experience almost the same current vectors throughout their journey, which makes them most likely remain close to each other throughout the mission. The downside is of course the fact that the system can only obtain data from a small proportion of the area.

When deployed in a large area (radius 2*km*), the floats disperse nicely during the mission. However, the probability for the floats to get extra information(information besides its own GPS fix when surfacing) is very low. It is so low that their estimation of respective location is of a similar level of the benchmark estimation which only rely on GPS fixes.

To comprise, one possible solution is to deploy floats into the water in a relatively small perimeter, and make their waiting depths slightly different. Since the current field in the Baltic sea is clearly layered, the current vectors they experience is different. By the end of the mission the floats drift nicely away from each others (up to 6*km*). Of course, one downside for this approach is that, in reality, the communication channel is correlated to a specific depth level, which will lower the probability of data exchange. See Figure 10.

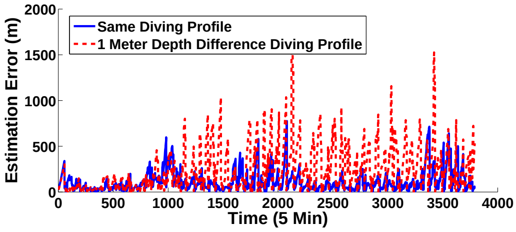


Figure 10: Estimation error comparison between 0m waiting depth (same profile) difference against 1m depth difference deployment for 5 floats.

5.6 Current Estimation

One of the main target of the mission is to accurately measure the subsurface currents speed and directions. Usually this is of particular difficulty since the complex situation underwater is hard to be measured directly. With the help of acoustic inter-robot communication, we have achieved some promising results.

Figure 11 and 12 show the current magnitude and angle estimation generated by Kalman filter from the one randomly chosen float.

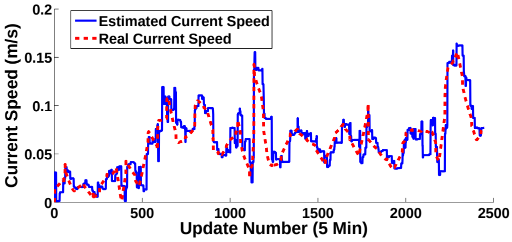


Figure 11: Current estimation magnitude error for 5 floats randomly deployed within 50m radius with the same diving profile.

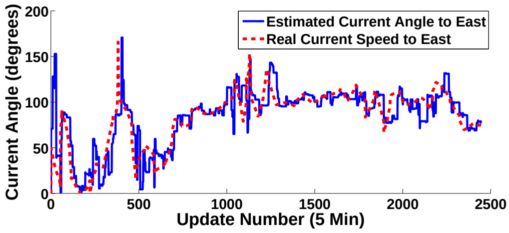


Figure 12: Current estimation angular error for 5 floats randomly deployed within 50m radius with the same diving profile.

There is a small lagging of the filter result towards the real environmental current data, which can be explained by the filter properties. However, the preliminary results are promising and deserve further research.

5.7 Float Numbers

Obviously, one of the most important factor is the number of floats in a given mission. During a mission, with a greater number of floats, there are more GPS fixes and distance measurements to share and there is a higher probability to gather such information from the communication between the floats. See Figures from 13, 14, 15.

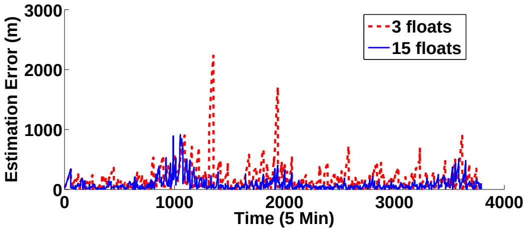


Figure 13: Estimation error comparison between 3 units and 15 units mission. Deployed in 50m radius area with 1 meter depth difference in communication level.

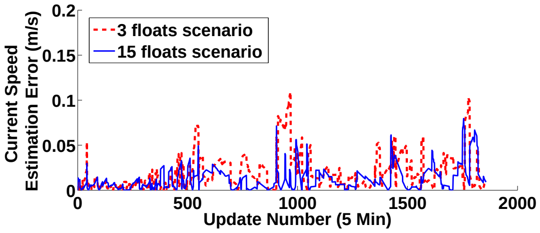


Figure 14: Comparison of absolute current speed estimation error between 3 floats and 15 floats scenarios. Both scenarios deploy within 50 meter radius and 1 meter waiting depth difference.

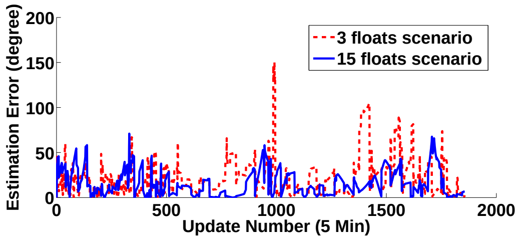


Figure 15: Comparison of absolute current direction estimation error between 3 floats and 15 floats scenarios. Both scenarios deploy within 50 meter radius and 1 meter waiting depth difference.

The number of floats in a given mission gives an interesting topic to research. The performance improvement also added a degree of difficulty and complexity for acoustic communication, data exchange via acoustic modem and the localization algorithm.

6 CONCLUSIONS AND FUTURE WORK

Multi-robot systems have a great inherent potential in various applications requiring localization and navigation capabilities. In this paper, we have shortly introduced a multi-robot system suitable for operations in the shallow water conditions near the coastline. The main contribution of this paper is the validation of our implementation of a localization algorithm in these demanding conditions. The obtained results, which are drawn from extensive simulations based on real case scenarios and environmental data, have proved the validity and effectiveness of our approach for the localization of these intelligent autonomous profiling floats.

However, there is much more yet to be explored in terms of our research. We will briefly introduce a better underwater communication channel modeling, in particular for the layered Baltic sea case. Further tests scenarios with different surfacing intervals, acoustic communication intervals and dynamic mission planning will be conducted. Later research will also include optimization between communication bandwidth, energy and reliability, data compression in data communication and other filtering algorithms. Real world robotic tests with renewed four unit system will be started in summer 2010 and they will further guide the future research.

ACKNOWLEDGEMENTS

We would like to thank the Academy of Finland for funding the Finnish Center of Excellence in Generic Intelligent Machines (GIM), the SWARM consortium, Jorma Selkäinaho, Janne Paanajärvi , Sami Terho from GIM, and Antti Westerlund (Finnish Meteorological Institute).

REFERENCES

Akyildiz, I. F., Pompili, D., and Melodia, T. (2006). State-of-the-art in protocol research for underwater acoustic sensor networks. In *WUWNet '06: Proceedings of the 1st ACM international workshop on Underwater networks*, pages 7–16, New York, NY, USA. ACM.

Costanza, R., d'Arge, R., de Groot, R., Farber, S., Grasso, M., Hannon, B., Limburg, K., Naeem, S., O'Neill, R. V., and Paruelo, J. (1998). The value of the world's ecosystem services and natural capital. *Ecological Economics*, 25(1):3–15.

Dario, I. A., Akyildiz, I. F., Pompili, D., and Melodia, T. (2005). Underwater acoustic sensor networks: Research challenges. *Ad Hoc Networks (Elsevier*, 3:257–279.

Davis, R. E., Sherman, J. T., and Dufour, J. (2001). Profiling ALACEs and other advances in autonomous subsurface floats. *J. Atm. Ocean. Tech.*, 18(6):982–993.

McFarland, D. and Honary, E. (2003). Flock distortion: A new approach in mapping environmental variables in deep water. *Robotica*, 21(4):365–383.

Saarinen, J., Maula, A., Nissinen, R., Kukkonen, H., Suomela, J., and Halme, A. (2007). Gimnet - infrastructure for distributed control of generic intelligent machines. In *Proceedings of the 13th IASTED International Conference on Robotics and Applications Telematics 2007*. The 13th IASTED International Conference on Robotics and Applications Telematics.

Vainio, M., Halme, A., Troshin, I., Stipa, T., Seppala, J., Pollehne, F., Bauerfeind, E., Haardt, H., Brault, P., Seube, N., Smerdon, A., Caine, S., Swale, B., and Hakala, A. (2004). Autonomous underwater multiprobe system for coastal area/shallow water monitoring (swarm). In *Proceedings of the EurOCEAN 2004 Conference*, Ireland. EurOCEAN 2004 Conference, Galway 2004.

Welch, G. and Bishop, G. (1995). An introduction to the kalman filter.

PARTICLE SWARM OPTIMIZATION USED FOR THE MOBILE ROBOT TRAJECTORY TRACKING CONTROL

Adrian Emanoil Serbencu, Adriana Serbencu and Daniela Cristina Cernega
Control Systems and Industrial Informatics Department, Computer Science Faculty
"Dunarea de Jos" University from Galati, 80021 Str. Domneasca 111, Galati, Romania
{Adrian.Serbencu, Adriana.Serbencu, Daniela.Cernega}@ugal.ro

Keywords: Trajectory Tracking, Nonlinear Control, Sliding Mode Control, Particle Swarm Optimization.

Abstract: The wheeled mobile robot is a nonlinear system. The trajectory tracking problem is solved using the sliding mode control. In this paper an optimization technique is investigated in order to obtain the best values for the sliding mode control law parameters. The performances of the control law with the optimum parameters are analyzed in order to establish some rules. The conclusions are based on the simulation results.

1 INTRODUCTION

To solve the trajectory tracking problem for a Wheeled Mobile Robot (WMR) it is used a nonlinear model (Slotine and Li, 1991):

$$x^{(n)} = f(x,t) + b(x,t) \cdot u \tag{1}$$

where x is the state variable; $x^{(n)} = [x, \dot{x}, \ddot{x}, \dots, x^{(n-1)}]$; $x^{(n)}$ is the n^{th}-order derivative of x; f is a nonlinear function; b is the gain and u is the control input.

The design of a variable structure control (VSC) (Gao and Hung, 1993) for a nonlinear system implies two steps: (1). "reaching mode" or nonsliding mode; (2). sliding mode.

For the reaching mode, the desired response usually is to reach the switching manifold s, described by:

$$s(x) = c^T \cdot x = 0 \tag{2}$$

in finite time with small overshoot with respect to the switching manifold.

The distance between the state trajectory and the switching manifold, s is stated as:

$$s(x,t) = \left(\frac{d}{dt} + \lambda \right)^{n-1} \cdot \tilde{x} = 0 \tag{3}$$

where $\tilde{x}$ is the tracking error and λ is a strictly positive constant which determines the closed-loop bandwidth. For example, if $n = 2$,

$$s = \dot{\tilde{x}} + \lambda \cdot \tilde{x} \tag{4}$$

Hence the corresponding switching manifold is *s(t)=0.* For a system having m inputs, m switching functions are needed.

It is proved that the most important virtue of the VSC systems is robustness. Properly design of the switching functions for a VSC system ensures the asymptotic stability. A number of design criteria exist for this purpose (Utkin and Young, 1978; Dorling and Zinober, 1986). Sliding Mode is also known to possess merits such as the invariance to parametric uncertainties. Dynamic characteristics of the reaching mode are very important, and this type of control suffers from the chattering phenomenon which is due to high frequency switching over discontinuity of the control signal.

The parameters of the control laws have to be positive, and their values influence the reaching rate and the chattering. The values of these parameters are not specified in the literature. In this paper the optimal values for these parameters will be searched.

Many optimization methods were proposed in literature. Recently, the Particle Swarm Optimizer (PSO) proposed by Eberhart and Kennedy (Kennedy and Eberhart, 1995), gained a huge popularity due to its algorithmic simplicity and effectiveness. The PSO is presented in Section 2. Section 3 is dedicated to the Trajectory Tracking Problem for the WMR. This problem is solved within the Sliding Mode approach and the result is the sliding-mode trajectory-tracking controller. The parameters p_i and q_i of the control law are not specified in the literature. In Section 4 PSO is used to determine the optimal values of the control law parameters in order to ensure maximum possible reaching rate of the switching manifold and minimum chattering and the results obtained are presented. Section 5 is dedicated to experimental results. Section 6 is dedicated to the conclusion and future work directions.

2 PARTICLE SWARM OPTIMISATION ALGORITHM

Particle swarm optimization (PSO) is one of the evolutionary computation techniques introduced in 1995 (Kennedy and Eberhart, 1995). The algorithm is initialized with a population of random solutions and searches for optima by updating generations. One entity of the population is named particle. PSO makes use of a velocity vector to update the current position of each particle in the swarm. Each particle keeps track of its coordinates in the problem space which are associated with the best solution it has achieved so far. This value is called *personal best*. The fitness value of *personal best* is also stored.

Another "best" position that is tracked by the PSO is the best one, obtained so far by any particle in the neighbours of the particle. This position is called *local best*. When a particle takes all the population as its topological neighbours, the best position is called global best. The PSO concept consists of, at each time step, changing the velocity of each particle toward its *personal best* and *local best* locations (local version of PSO). Every component of velocity is weighted by a random term, which assures the exploration of problem space.

This process is iterated a set number of times, or until a stop criterion is achieved, for example a threshold of distance (absolute or relative) between the two last positions, below which it is not necessary to go. Using a population of solutions allows PSO to avoid, in most cases, convergence to local optimum.

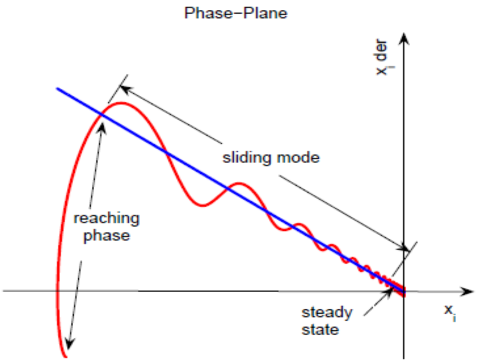


Figure 1: Diagram for the concept of 3-Stages approach.

3 TRAJECTORY TRACKING PROBLEM

In this paper the model used for the controlled robot is a 2-order MIMO (Multiply Input Multiply Output) nonlinear system that is "linear in control". The model and the control law used are:

$$\ddot{x} = f(x,\dot{x},t) + B(x,\dot{x},t)\cdot u \tag{5}$$

$$u = p(x,\dot{x},t) \tag{6}$$

where $x = [x_1, x_2, \cdots, x_n]$, $x \in \Re^n$, f is a vector of nonlinear functions, $f \in \mathcal{L}_2^n$, B is a matrix of gains, $B \in \Re^{n\times n}$; $\det(B) \neq 0$; u is the control vector, $u \in \Re^n$.

For the 2nd-order MIMO nonlinear system having the model shown in (6) efficient sliding mode control can be achieved via the following stages (see Figure 1):

1st reaching phase motion; during this stage the trajectory is attracted towards the switching manifold (if the reaching condition is satisfied); characterized by $s_i \neq 0, \tilde{x}_i \neq 0, \dot{\tilde{x}}_i \neq 0$

2nd sliding mode motion; during this stage the trajectory stays on the switching manifold, i.e. $s_i = 0, \tilde{x}_i \neq 0, \dot{\tilde{x}}_i \neq 0$

3rd steady state; during this stage both the state variable and the state velocity will converge to the steady state value, therefore:

$$s_i = 0, \text{ and} \begin{cases} \tilde{x}_i \to 0, \dot{\tilde{x}}_i \to 0 \quad or \\ \quad \tilde{x}_i = 0, \dot{\tilde{x}}_i = 0 \end{cases}$$

The reaching law is a differential equation which specifies the dynamics of a switching function *s(x)*. The differential equation of an asymptotically stable *s(x)*, is itself a reaching condition. In addition, by the choice of the parameters in the differential equation, the dynamic quality of the VSC system in the reaching mode can be controlled.

Gao and Hung (Gao and Hung, 1993) proposed a reaching law which directly specifies the dynamics of the switching surface by the differential equation

$$\dot{s} = -Q\cdot sgn(s) - P\cdot h(s) \tag{7}$$

where $Q = diag[q_1, q_2, \cdots, q_n], q_i > 0, i = 1, 2, \dots, n$
$\mathrm{P} = diag[p_1, p_2, \cdots, p_n], p_i > 0, i = 1, 2, \dots, n$
and $\mathrm{sgn}(s) = [\mathrm{sgn}(s_1), \mathrm{sgn}(s_2), \cdots, \mathrm{sgn}(s_n)^T]$
$h(s) = [h_1(s_1), h_2(s_2), \cdots, h_n(s_n)]^T$; $s_i \cdot h_i(s) > 0, \ h_i(0) = 0.$

In this paper, a constant plus proportional rate reaching law proposed in Gao and Hung is investigated:

$$\dot{s} = -\mathrm{Q}\cdot\mathrm{sgn}(\mathrm{s}) - P\cdot s \tag{8}$$

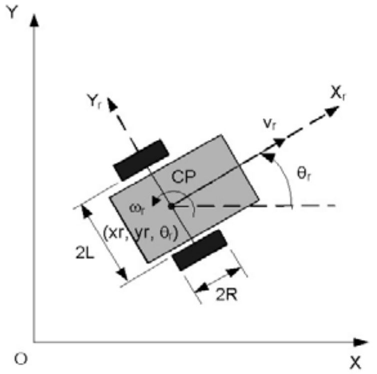


Figure 2: WMR model and symbols.

Clearly, by using the proportional rate term $-P \cdot s$, the state is forced to approach the switching manifolds faster when s is large.

The purpose of the trajectory tracking is to control the non-holonomic WMR to follow a desired trajectory, with a given orientation relatively to the path tangent, even when different disturbances exist. In the case of trajectory-tracking the path is to be followed under time constraints. Trajectory tracking is formulated as having the WMR following a virtual target which is assumed to move exactly along the path with specified velocity profile.

3.1 Kinematic Model of a WMR

Figure 2 presents a WMR with two diametrically opposed drive wheels (radius R) and free-wheeling castors. *Pr* is the origin of the robot coordinates system. *2L* is the length of the axis between the drive wheels. ω_R and ω_L are the angular velocities of the right and left wheels. Let the pose of the mobile robot be defined by the vector, $q_r = [x_r \ y_r \ \theta_r]^T$ where $[x_r \ y_r]^T$ denotes the robot position on the plane and θ_r the heading angle with respect to the x-axis. In addition, v_r denotes the linear velocity of the robot, and ω_r the angular velocity around the vertical axis.

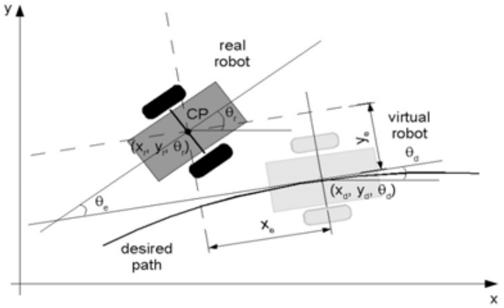


Figure 3: Lateral, longitudinal and orientation errors (trajectory-tracking).

For a unicycle WMR rolling on a horizontal plane without slipping, the kinematic model can be expressed by: which represents a nonlinear system.

$$\begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\theta}_r \end{bmatrix} = \begin{bmatrix} cos\theta_r & 0 \\ sin\theta_r & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v_r \\ \omega_\mathrm{r} \end{bmatrix} \tag{9}$$

3.2 Trajectory-tracking

Without loss of generality, it can be assumed that the desired trajectory $q_d(t) = [x_d(t) \ y_d(t) \ \theta_d(t)]^T$ is generated by a virtual unicycle mobile robot (see Figure 3). The kinematic relationship between the virtual configuration $q_d(t)$ and the corresponding desired velocity inputs $[v_d(t) \ \omega_d(t)]^T$ is analogue with (9):

$$\begin{bmatrix} \dot{x}_d \\ \dot{y}_d \\ \dot{\theta}_d \end{bmatrix} = \begin{bmatrix} cos\theta_d & 0 \\ sin\theta_d & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v_d \\ \omega_\mathrm{d} \end{bmatrix} \tag{10}$$

When a real robot is controlled to move on a desired path it exhibits some tracking error. This tracking error, expressed in terms of the robot coordinate system, as shown in Figure 3, is given by

$$\begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} = \begin{bmatrix} cos\theta_d & sin\theta_d & 0 \\ -sin\theta_d & cos\theta_d & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_r\text{-}x_d \\ y_r\text{-}y_d \\ \theta_r\text{-}\theta_d \end{bmatrix} \tag{11}$$

Consequently one gets the error dynamics for trajectory tracking as

$$\begin{cases} \dot{x}_e = -v_d + v_r \cdot cos\theta_e + \omega_\mathrm{d} \cdot y_e \\ \dot{y}_e = v_r \cdot sin\theta_e - \omega_\mathrm{d} \cdot x_e \\ \dot{\theta}_e = \omega_\mathrm{r} - \omega_\mathrm{d} \end{cases} \tag{12}$$

3.3 Sliding-mode Trajectory-tracking Control

Uncertainties which exist in real mobile robot applications degrade the control performance significantly, and accordingly, need to be compensated. In this section, is proposed a sliding-mode trajectory-tracking (SM-TT) controller, in Cartesian space, where trajectory-tracking is achieved even in the presence of large initial pose errors and disturbances.

Let us define the sliding surface $s = [s_1 \ \ s_2]^T$ as

$$\begin{aligned} s_1 &= \dot{x}_e + k_1 \cdot x_e \\ s_2 &= \dot{y}_e + k_2 \cdot y_e + k_0 \cdot sgn(y_e) \cdot \theta_e \end{aligned} \tag{13}$$

where k_0, k_1, k_2 are positive constant, x_e, y_e and θ_e are

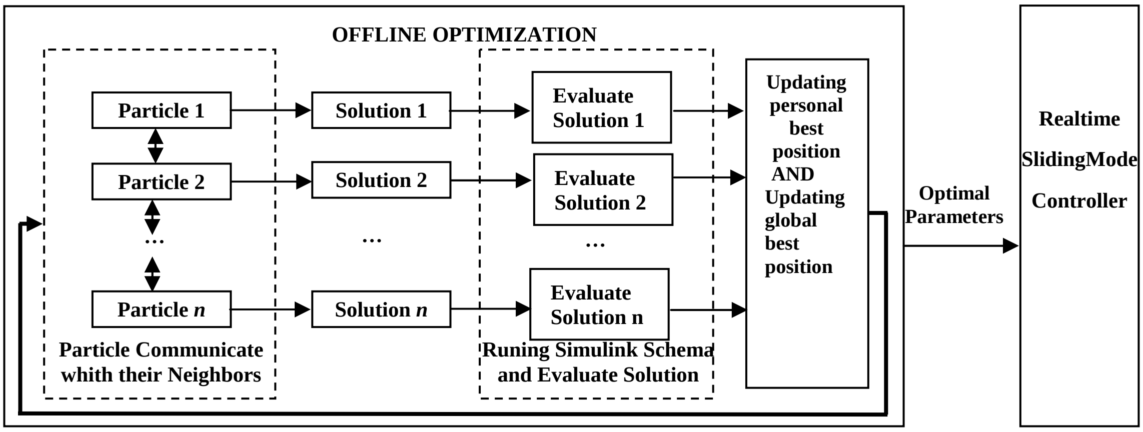

OFFLINE OPTIMIZATION
Particle 1
Particle 2
...
Particle n
Particle Communicate whith their Neighbors
Solution 1
Solution 2
...
Solution n
Evaluate Solution 1
Evaluate Solution 2
...
Evaluate Solution n
Runing Simulink Schema and Evaluate Solution
Updating personal best position AND Updating global best position
Optimal Parameters
Realtime SlidingMode Controller

Figure 4: SM-TT controller parameters optimization with PSO.

the trajectory-tracking errors defined in (11).

If s_1 converges to zero, trivially x_e converges to zero. If s_2 converges to zero, in steady-state it becomes $\dot{y}_e = -k_e \cdot y_e - k_0 \cdot sign(y_e) \cdot \theta_e$.

For $y_e < 0 \Rightarrow \dot{y}_e > 0$ if only if $k_0 < k_2 \cdot |y_e| / |\theta_e|$.

For $y_e > 0 \Rightarrow \dot{y}_e < 0$ if only if $k_0 < k_2 \cdot |y_e| / |\theta_e|$. Finally, it can be known from s_2 that convergence of y_e and $\dot{y}_e$ leads to convergence of θ_e to zero.

From the time derivative of (13) and using the reaching laws defined in (8), yields:

$$\begin{aligned} \dot{s}_1 &= \ddot{x}_e + k_1 \cdot \dot{x}_e = -q_1 \cdot sgn(s_1) - p_1 \cdot s_1 \\ \dot{s}_2 &= \ddot{y}_e + k_2 \cdot \dot{y}_e + k_0 \cdot sgn(y_e) \cdot \dot{\theta}_e = \\ &= -q_2 \cdot sgn(s_2) - p_2 \cdot s_2 \end{aligned} \tag{14}$$

From (11), (12) and (14), and after some mathematical manipulation, the output commands of the sliding-mode trajectory-tracking controller result:

$$\dot{v}_c = \frac{-p_1 \cdot s_1 - q_1 \cdot sgn(s_1) - k_1 \cdot \dot{x}_e - y_e \cdot \dot{\omega}_d}{cos(\theta_e)} + \frac{-\dot{y}_e \cdot \omega_d + v_r \cdot \dot{\theta}_e \cdot sin(\theta_e) + \dot{v}_d}{cos(\theta_e)} \tag{15}$$

$$\omega_c = \frac{-p_2 \cdot s_2 - q_2 \cdot sgn(s_2) - k_2 \cdot \dot{y}_e + x_e \cdot \dot{\omega}_d + \dot{x}_e \cdot \omega_d - \dot{v}_r \cdot sin(\theta_e)}{v_r \cdot cos(\theta_e) + k_0 \cdot sgn(y_e)} + \omega_d$$

Let us define $V = \frac{1}{2} \cdot s^T \cdot s$ as a Lyapunov function candidate, therefore its time derivative is

$$\begin{aligned} \dot{V} &= s_1 \cdot \dot{s}_1 + s_2 \cdot \dot{s}_2 = s_1 \cdot (-q_1 \cdot s_1 - p_1 \cdot sgn(s_1)) + \\ &+ s_2 \cdot (-q_2 \cdot s_2 - p_2 \cdot sgn(s_2)) = \\ &= s^T \cdot Q \cdot s - p_1 \cdot |s_1| - p_2 \cdot |s2| \end{aligned}$$

For $\dot{V}$ to be negative semi-definite, it is sufficient to choose q_i and p_i such that q_i, $p_i \geq 0$. But the optima values for q_i, $p_i \geq 0$ will be determined in the next section.

The *signum* functions in the control laws were replaced by *saturation* functions, to reduce the chattering phenomenon (Slotine and Li, 1991).

4 SLIDING MODE CONTROLLER PARAMETERS EVALUATED WITH PSO

Solving the Trajectory Tracking Problem with a SMC, leads to the reaching laws (15). In literature, the parameters *q1, q2*, *p1* and *p2* are usual determined through experiments (Solea and Cernega, 2009) and have great impact on the performance of the controller. *q1, q2* influence the rate at which the switching variable *s(x)* reach the switching manifold *S.* Parameters *p1, p2* force the state *x* to approach the switching manifolds faster when *s* is large.

Choosing parameters through experiments only depends on experience or repeated debugging. In this paper is presented a method of choosing the parameters of the Sliding Mode Controller using the PSO algorithm. The advantages of PSO are: simplicity and efficiency, proven in many other parameters training problems (Mendes *et al.*, 2002; Kim *et al.*, 2008). The optimisation algorithm is working off-line. The *P* and *Q* parameters found by PSO can be used in real-time implementation of SM-TT controller on PatrolBot Robot (see Figure 4).

PSO algorithm is generating, at each step, a number of solutions equal to the number of particles. The quality (fitness) of the solutions is evaluated with the objective function. PSO algorithm requires current solutions fitness to calculate new solutions at the next iteration. The objective function used in PSO takes into account booth the speed of reaching

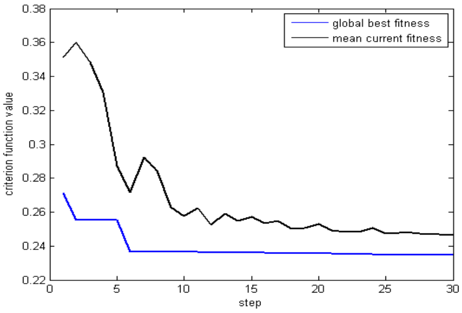


Figure 5: The evolution of best value found by PSO and of the mean criterion function over all particle.

manifolds and the amplitude of the chattering. This is accomplished using the sum of root mean square of the two errors x_e and y_e (11). The evaluation of every set of parameters is achieved after running a numerical simulation of the SM-TT control structure implemented in a Matlab Simulink schema that contains the model of the robot.

The horizon of simulation and initial conditions are chosen to allow a correct comparison between sets of parameters. The step of simulation is selected according to the one used to control the PatrolBot.

Let us suppose that the swarm is composed by n particles. Each particle i is recorded as a structure that transfer from the current iteration t to the next iteration the four elements specified below:

- The current position of i^{th} particle in the search space at the moment t is given through a vector with 4 components

$$x_i(t)=(x_{q1i}(t),\ x_{q2i}(t),\ x_{p1i}(t),\ x_{p2i}(t)).$$

- The current velocity $v_i(t)$ of i particle is a vector with components for each direction of the search space too.
- The best position found up to now by this particle is given by a vector $l_i(t)$ with the same meaning as $x_i(t)$.
- The quality of personal best position $l_i(t)$.

In order to compute the next position where to move, every particle of the swarm needs one more information: the best position found by its neighbours stored in the vector $g_i(t)$. Generally, this is written simply x, v, l, and g. The d^{th} component of one of these vectors is indicated by the index d, for example x_d. With these notations, the motion equations of a particle are, for each dimension $d\in\{\ q1i,\ q2i,\ p1i,\ p2i\}$:

$$\begin{cases} v_d \leftarrow c_1 v_d + c_2(l_d - x_d) + c_3(g_d - x_d) \\ x_d \leftarrow x_d + v_d \end{cases} \tag{16}$$

The confidence coefficients are defined as

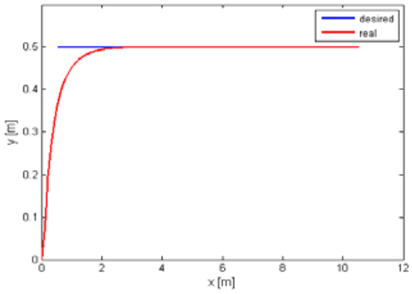


Figure 6: Simulated trajectory with the parameters value found by PSO. Experimental SM-TT control starting from an initial error state ($x_e(0)$ = -0.5, $y_e(0)$ = -0.5, $\theta_e(0)$ = 0).

follows:

- c_1 is (confidence in its own movement) the
- inertia weight, with linear variance from 0.9 to 0.4.
- c_2, c_3 (respectively confidence in its best performance and that of its best informant) are randomly selected at each step according to an uniform distribution in the interval [0, $c_{\max}$].

A circular neighbourhood is used as graph of influence.

5 EXPERIMENTAL RESULTS

Based on the above analysis, mathematical simulation software MATLAB was used to accomplish the experiment simulation study.

The mobile robot PatrolBot used in simulation is assumed to have the same structure as in Figure 2. Parameter values of the PatrolBot are: mass of the robot body 46 [Kg], radius of the drive wheel 0.095 [m], and distance between wheels 0.48 [m]. The parameters of sliding modes were held constant during the experiments: k_1 = 0.75, k_2 = 3.75, and k_0 = 2.5; and the desired trajectory is given by v_d = 0.5 [m/s], ω_d = 0 [rad/s].

The experiments were done on the robot with the initial error (x_e = -0.5 [m], y_e= -0.5 [m], θ_e = 0 [deg]) and used the reaching law (8).

Settings, used in Matlab implementation of PSO algorithm, are: particle number n = 20; maximal number of iteration = 30, $c_{\max}$ =1.9. The criterion function used for solution evaluation are the sum of root mean square (RMS) of the two errors longitudinal - x_e and lateral - y_e. RMS error is an old, proven measure of control and quality. Taking into account that parameters must be positive and value too large can causes chattering, the search interval for each SM parameters was selected to be [0.01 5].

In Figure 5 the evolution of criterion function

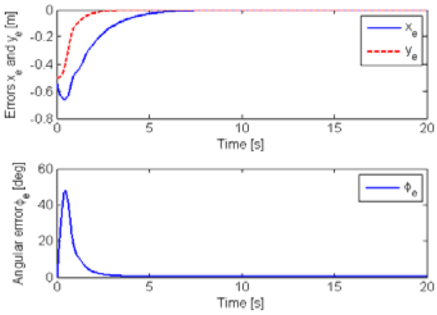


Figure 7: Longitudinal, lateral and orientation errors for experimental SM-TT control.

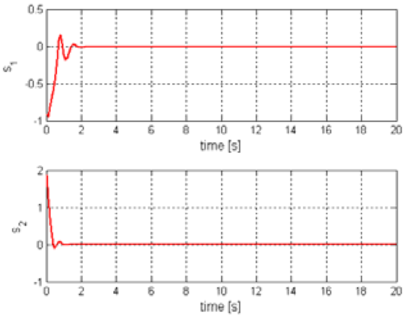


Figure 8: Sliding surface for SM-TT controller.

best value found by PSO is presented. Average value of criterion function for all particles is also recorded. Note that a number of 15 iterations are sufficient to find a good set of values of parameters.

The parameters values for the considered PatrolBot, found by PSO are $q1$=0.4984, $q2$=1.9075, $p1$=1.0230 and $p2$=2.4872.

In Figures 6 and 7, the simulation results for the case of optimised parameters are presented.

In Figure 8 the two sliding manifolds are represented. In Figure 8 one can also see the value of the reaching time.

Figure 9 presents the response of the robot corresponding to the situation of a poor choice of the control law parameters without any optimisation.
It is easy to see the difference between the performances between Figures 6 and 9.

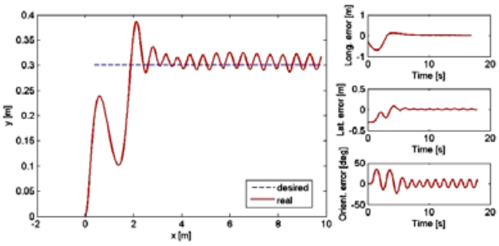


Figure 9: An unfavourable case of experimental SM-TT control.

6 CONCLUSIONS

The paper proposed an efficient method to determine the optimum set of parameters for the sliding mode controller. The PSO algorithm proved to be adequate for this problem because it eliminates the need for repeated simulations in order to find a satisfactory set of parameters. The tests have proven that this optimization technique is efficient for the problem to be solved. A very good solution without chattering was found in a quite acceptable time interval and number of iterations.

The search of the optimum values for the sliding mode trajectory tracking control laws parameters was done in order to use, in the future, such optimum parameters into a supervised control structure having the ability to switch between different controllers.

ACKNOWLEDGEMENTS

This work was supported by the CNCSIS - UEFISCSU, project number IDEI-506/2008.

REFERENCES

Slotine, J., Li, W., 1991. *Applied Nonliner Control*, Prentice Hall, New Jersey.

Gao, W., Hung, J., 1993. Variable structure control of nonlinear systems: A new approach. *IEEE Transactions on Industrial Electronics*, 40, pp. 45–55.

Utkin, V., Young, K., 1978. Methods for constructing discontinuity planes in multidimensional variable structure systems. *Automation and Remote Control, 39*, pp. 1466–1470.

Dorling, C., Zinober, A., 1986. Two approaches to hyperplane design in multivariable variable structure control systems. *Int Journal Control*, Vol. 44, p.65–82

Kennedy, J., Eberhart, R. C., 1995. Particle Swarm Optimization. *IEEE International Conference on Neural Networks*, Perth, Australia, pp. 1942-1948.

Solea, R., Cernega, D. C., 2009. Sliding Mode Control for Trajectory Tracking Problem - Performance Evaluation. *Artificial Neural Networks – ICANN 2009, Lecture Notes in Computer Science*, Vol. 5769, pp. 865-874.

Mendes, R., Cortez, P., Rocha, MR., Firns, J., 2002. Particle Swarms for Feedforward Networks Trening. *International Conference on Neural Networks*, Honolulu (Hawaii), USA, pp. 1895-1889.

Kim, T. H., Maruta, I., Sugie, T., 2008. Robust PID controller tuning based on the constrained particle swarm optimization. *Automatica*, Vol. 44, pp. 1104–1110.

EFFICIENT LOCOMOTION ON NON-WHEELED SNAKE-LIKE ROBOTS

Julián Colorado, Antonio Barrientos, Claudio Rossi, Mario Garzón, María Galán and Jaime del Cerro
Robotics and Cybernetics Group, Robotics and Automation Center UPM-CSIC, Universidad Politécnica de Madrid, Spain
jcolorado@etsii.upm.es, {antonio.barrientos, claudio.rossi, j.cerro}@upm.es

Keywords: Serpenoid curves, Snake bio-mechanisms, Bio-inspired locomotion, Crawling robots.

Abstract: This article presents our current work on studying energy efficient locomotion on crawling snake-like robots. The aim of this work is to use existing biological inspired methods to demonstrate lateral undulation planar gaits for efficiently controlling high-speed motion as a function of the terrain surface. A multilink non-wheeled snake-like robot is being developed for experimentation and analysis of efficient serpentine locomotion based on simulation results.

1 INTRODUCTION

In nature, snakes are able to move on different environments. Generally speaking, snakes can adapt to a particular terrain employing changes in their muscles-shape (Kane and Lecision, 2000). This potential provides snakes with higher rough terrain adaptability on irregular surfaces compared to legged animals.

The first attempts of approaching biological inspired snake motion using an artificial counterpart was conducted by Hirose in the 70's (Hirose, 1993). He made the analysis of limbless motions experimental data and suggested mathematical description of the snake's instant form. The curve was called *–serpenoid–* and is widely used for the snake robot's control assignment nowadays. The first designs of Hirose's snake robots had modules with small passive wheels, and since then, most of the current developments (Downling, 1997), (Chirikjian and Burdick, 1990), (Ostrowski, 1995) remain using snake robots with wheels in order to facilitate forward propulsion. Nonetheless, snake-like robots that have no wheels are closer to their biological counterparts. The difficulty in analyzing and synthesizing snake locomotion mechanisms is not as simple as wheeled mechanisms. One of the main drawbacks relies on their poor power efficiency for surface traction, and consequently locomotion. While most works address contributions in terms of snake control and full autonomous navigation (Kamegawa et al., 2002), (Prautsch and Mita, 1996), (Transeth et al., 2006) our work is focused on providing modeling foundations to use a non-wheeled snake robot that can adapt to the environment at the advantage of energy efficiency.

Our goal is to establish a mathematical framework for modeling that relates the existing knowledge of biological snake locomotion with the dynamics behavior that achieves minimal energy waste when the snake moves at high speeds over 1m/s. Section 2 of this article briefly presents how to achieve lateral undulation serpentine gaits using Hirose's serpenoid curves and how to integrate that approach within the dynamics equations of motion. A friction model is also addressed in order to achieve the proper forward motion based on internal joint torques. Section 3 introduces how to optimize snake locomotion by choosing the optimal serpenoid curve parameters that minimize energy consumption. Simulation results show efficient motion over ground. Finally, Section 4 presents conclusions and upcoming future work will present experimental validation using an experimental testbed (under current development) that consists on nine articulated modules serially connected.

2 SERPENTINE MODELING

Almost all limbless vertebrates, including snakes, mimic their ancestors by shaping their bodies in a *–S-shaped–* curve that travels tailwards (Gray and Lissmann, 1950). Snakes commonly propel themselves on the ground or water by summing longitudinal resultants of lateral forces. This kind of motion is called *Lateral Undulation* (see Figure 1).

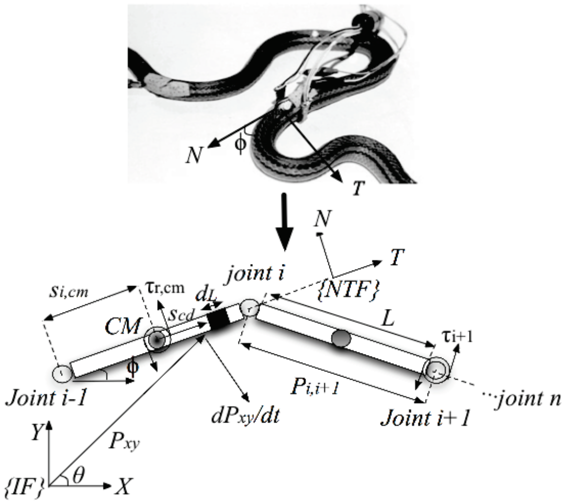


Figure 1: (Above): The *s-shape* that biological snakes perform to move forward using lateral undulation pattern. (Below): serially coupled rigid body system description.

Table 1 introduces a description of the variables involved within the framework of snake-like robot modeling based on Figure 1.

Table 1: Description of snake-like robot parameters.

Description	Notation	Units
Number of links	n	-
Joint position at body-*i*	ϕ_i	[rad]
Body-*i* orientation with respect to inertial frame {IF}	θ_i	[rad]
Vector position from joint's frame to body's CM	$\vec{s}_{i,cm}$	[m]
Distance from body's CM to differential length d_L	s_{cd}	[m]
Link-*i* length	L	[m]
Vector position from joint-*i* frame to joint *i+1*	$\vec{p}_{i,i+1}$	[m]
Joint torque of body-*i*	τ_i	[Nm]
Friction torque of body-*i* CM	$\tau_{r,cm}$	[Nm]
Friction coefficients	C_N, C_T	[s^{-1}]

2.1 Serpenoid Curves

Hirose found that snakes take their body onto so-called serpenoid curve when they move with a serpentine gait. Considerer the snake-like robot depicted in Figure 1, which consist on *n*-links serially connected through *n-1* joints. The undulatory motion of a snake can be imitated by changing the relative angles ϕ of snake's bodies as shown in Equation 1. See details in (Hirose and Morishima, 1990).

$$\phi_i(t) = 2\alpha \sin\bigl(\omega_s t + (i-1)\beta\bigr) + \gamma \tag{1}$$

The term $\phi_i(t)$ is a sinusoidal function varying along the arc length i/n $(\forall_i : i = 1 \ldots n-1)$ at ω_s angular speed propagation. The terms, α, β and γ are the parameters that determine the shape of the serpentine curve realized by the snake-like robot (e.g. if $\gamma = 0$, the curve will describe a straight path and when $\gamma \neq 0$ the curve will trace a circular path). Equation 2 shows these parameters.

$$\alpha = a\left|\sin\left(\frac{\beta}{2}\right)\right|, \quad \beta = \frac{b}{n}, \quad \gamma = -\frac{c}{n} \tag{2}$$

Based on different choices of the parameters a, b and c, Figure 2 shows several serpenoid curves profiles. The parameter a determines the degree undulation, b the number of periods in a unit length, and c is the motion circular bias.

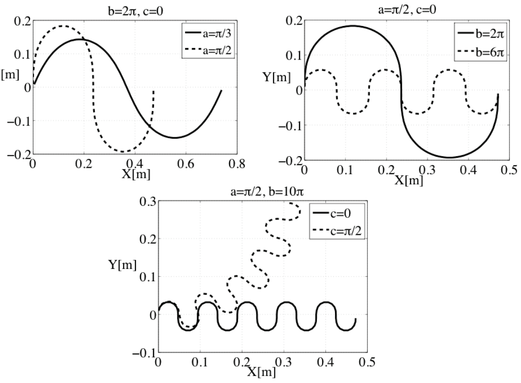


Figure 2: Computing Lateral Undulation serpentine gaits using serpenoid curves.

Our first objective is to merge this serpentine curve approach into our snake dynamics model. The key property of snakes in achieving forward locomotion is the difference in the friction coefficients for the tangential–*T* and the normal–*N* directions with respect to the body. In particular, the normal friction tends to be much larger than the tangential friction, leading to avoidance of side slipping. In order to analyze such property, next subsection introduces the solution of the dynamic's Equations of Motion –EoM of a multilink articulated body system as shown in Figure 1, and the incorporation of a simple friction model to provide accurate snake forward propulsion.

2.2 Snake-Like Robot Dynamics

Applying the D'Alembert's principle (Fu et al., 1987) and assuming a snake's tail (base) to end-body (head) recursive propagation of kinematics spatial velocities in (3) [angular and linear components stacked in a single 6-dimensional quantity]:

$$V_i = P^T_{i-1,i} R^T_{i,i-1} V_{i-1} + H_i \dot{\phi}_i \quad \{\forall_i : i = 1 \ldots n\} \tag{3}$$

In multibody dynamics, spatial quantities must be propagated and projected onto unique frames in order to be operator on. For this purpose, operators for translation: $P_{i-1,i} \in \Re^{6x6}$ and rotation: $R_{i,i-1} \in \Re^{6x6}$ are defined as:

$$P_{i-1,i} = \begin{bmatrix} I & \tilde{p}_{i-1,i} \\ 0 & I \end{bmatrix}, \quad R_{i,i-1} = \begin{bmatrix} r_{i,i-1} & 0 \\ 0 & r_{i,i-1} \end{bmatrix}, \tag{4}$$

where $I \in \Re^{3x3}$ is the identity operator, $\tilde{p}_{i-1,i} \in \Re^{3x3}$ is the skew symmetric matrix corresponding to the vector cross product of $\vec{p}_{i-1,i} \in \Re^3$, which is any vector joining e.g. joint *i* to joint *i+1* in Figure 1. The term $r_{i,i-1} \in \Re^{3x3}$ refers to the generalized rotation matrix that takes any point in coordinate frame-*i* and projects it onto frame *i-1*. The joint velocity $\dot{\phi}_i$ is obtained by taking the derivative of Equation (1) with respect to time (serpenoid curve). Finally the $H_i \in \Re^6$ vector allows the projection of the joint velocity with respect to the axis of motion of the joint.

Differentiating Equation (3) with respect to time, the spatial accelerations are:

$$\begin{aligned} \dot{V}_i &= P^T_{i-1,i} R^T_{i,i-1} \dot{V}_{i-1} + H_i \ddot{\phi}_i + \dot{P}^T_{i-1,i} R^T_{i,i-1} V_{i-1} \\ &+ \dot{H}_i \dot{\phi}_i \qquad \{\forall_i : i = 1 \ldots n\}, \end{aligned} \tag{5}$$

where the third and fourth terms corresponds to coriolis and centrifugal accelerations. Finally, a backward propagation of spatial forces yields:

$$\begin{aligned} F_i &= J_i \dot{V}_i + \left[\dot{J}_i - \dot{S}_{i,cm} J_i\right] V_i + R_{i,i-1} P^T_{i-1,i} F_{i+1} \\ &+ S_{i,cm} F_{ri,cm} \quad \{\forall_i : i = n \ldots 1\}, \end{aligned} \tag{6}$$

where $J_i \in \Re^{6x6}$ is the mass operator defined by the inertia tensor. The operator $S_{i,cm} \in \Re^{6x6}$ has the same structure of $P_{i-1,i} \in \Re^{6x6}$ in Equation (4), and corresponds to the distance ($\vec{r}_{i,cm} \in \Re^3$) between the joint frame and the CM of the body. Finally the joint toques are: $\tau_i = H_i^T F_i$.

2.3 Modeling Surface Friction

Friction force is essential to achieve forward motion. From Equation (6), the term $F_{ri,cm} \in \Re^6$ is the friction force referred to the CM frame and yields:

$$F_{ri,cm} = \begin{bmatrix} 0 & 0 & \tau_{r,cm} & f_{r,xi} & f_{r,yi} & 0 \end{bmatrix}^T, \tag{7}$$

where $\tau_{r,cm}$ is the torque friction component due to planar rotation, and the terms $\left[f_{r,xi}, f_{r,yi}\right]$ are the components due to translation. Differentiating the position vector P_{xy} with respect to time:

$$P_{xy} = \begin{bmatrix} X \\ Y \end{bmatrix} + \begin{bmatrix} c\theta \\ s\theta \end{bmatrix} s_{cd}, \; \dot{P}_{xy} = \begin{bmatrix} \dot{X} \\ \dot{Y} \end{bmatrix} + \begin{bmatrix} -s\theta \\ c\theta \end{bmatrix} s_{cd} \dot{\theta} \tag{8}$$

Modeling the linear friction for the differential d_L with respect to the {NTF}-frame yields:

$$\begin{bmatrix} df_{r,T} \\ df_{r,N} \end{bmatrix} = -\begin{bmatrix} C_T & 0 \\ 0 & C_N \end{bmatrix} \begin{bmatrix} \tilde{v}_T \\ \tilde{v}_N \end{bmatrix} dm_i \tag{9}$$

Considering that the tangential and normal velocity $\left[\tilde{v}_T, \tilde{v}_N\right]$ are related to the linear velocity $\dot{P}_{xy}$ with the following transformation:

$$\begin{bmatrix} \tilde{v}_T \\ \tilde{v}_N \end{bmatrix} = \begin{bmatrix} c_\theta & s_\theta \\ -s_\theta & c_\theta \end{bmatrix} \begin{bmatrix} \dot{X} \\ \dot{Y} \end{bmatrix}, \tag{10}$$

the translation friction force is:

$$\begin{bmatrix} f_{r,xi} \\ f_{r,yi} \end{bmatrix} = -m \begin{bmatrix} c_\theta^2 C_T - s_\theta^2 C_N & s_\theta c_\theta \left(C_T - C_N\right) \\ s_\theta c_\theta \left(C_T - C_N\right) & s_\theta^2 C_T + c_\theta^2 C_N \end{bmatrix} V_i \tag{11}$$

The C_T and C_N parameters are the tangential and normal coefficients. In addition, the component of the total friction torque for the differential d_L is:

$$\begin{aligned} \tau_{r,cm} &= -C_N \int s_{cd} \left(\tilde{v}_N + s_{cd} \dot{\theta}\right) dm \\ &= -C_N \left(\int s_{cd} \tilde{v}_N dm + \int s_{cd}^2 \dot{\theta} dm\right) \end{aligned} \tag{12}$$

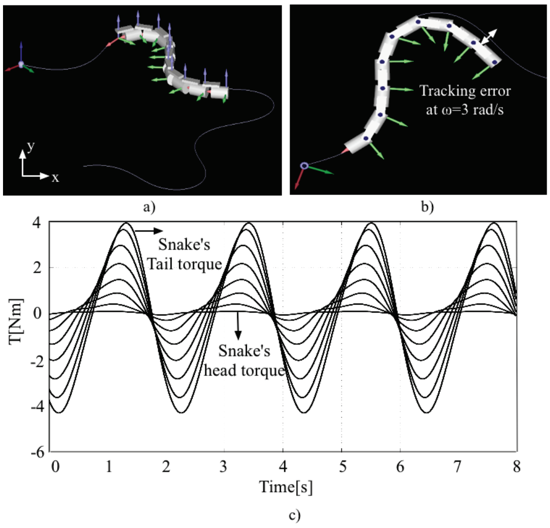


Figure 3: Snake-like robot simulation: a). Circular serpenoid curve using $\gamma = 1.2^{\circ}$, b). *Top view* tracking error due to inefficient friction and serpenoid parameters tuning, c). Joint torques to move the snake forward.

Assuming the relation: $dm = m \cdot L^{-1} ds$ (being L the length of a body-*i*; see Figure 1), the friction torque about the body-*i* CM is:

$$\tau_{r,cm} = -C_N \frac{m}{L}\dot{\theta}\int_0^L s_{cd}^2 ds = -\frac{C_N m L^2}{3}\dot{\theta} \qquad (13)$$

Using Equations (11) and (13), the total friction force vector denoted by Equation (7) is incorporated into the dynamic's EoM of the snake-like robot (see Equation (6)).

Once the snake modelling is completed, Figure 3 depicts the 8-degree of freedom snake-like robot (*n=8*) performing a circular serpenoid trajectory profile: $\alpha = 0.5 rad,\ \beta = 1.1 rad,\ \omega_s = 2.5 rad/s$.

Friction coefficients make the robot to generate larger friction forces in the normal direction than in the tangential direction of the motion. Considering $C_T = 12$, $C_N = 20$ and the mass of each snake's link in $m = 0.4 kg$. Despite the snake is capable of propelling forward, a tracking error (caused by external friction forces) appears when the snake is speeding up. These preliminary results suggest us to analyze how to relate energy consumption (input power against speed), and how to tune friction parameters as a function of that speed and serpenoid curve terms. For this purpose, our goal in the next section is to find how to change the serpenoid curve parameters in Equation (2) to achieve the energy-efficient locomotion based on the dynamic's EoM in Equation (6).

3 EFFICIENT SNAKE MOTION

Friction force represents a power loss. Part of the input power generated by the snake's actuators is converted into kinetic energy K_E and the rest is lost due to the friction.

The objective of this section is to find the optimally efficient motion within the dynamics framework of serpentine locomotion. More precisely, our challenge is related to choosing the parameters α, β, ω_s that make the average power loss P_L in Equation (14) minimal while keeping a prescribed average speed.

$$P_L = m V_i^T \begin{bmatrix} -C_N m L^2/3 & 0 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} c_\theta^2 C_T - s_\theta^2 C_N & s_\theta c_\theta (C_T - C_N) \\ s_\theta c_\theta (C_T - C_N) & s_\theta^2 C_T + c_\theta^2 C_N \end{bmatrix} V_i \qquad (14)$$

The approach we used is to increase the speed of the snake and verifying where is the saturation point that preserves an efficient locomotion in terms of power loss compared to the total energy of the system described in Equation (15) as:

$$K_E = \sum_{i=1}^{n} \frac{1}{2}\dot{\phi}_i^T J(\phi)\dot{\phi}_i \cdot \qquad (15)$$

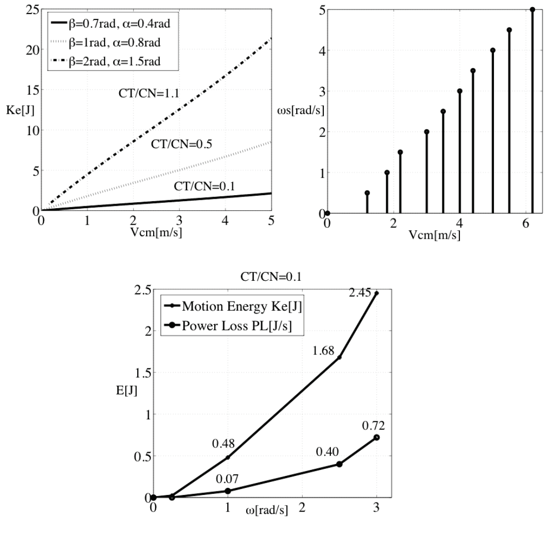


Figure 4: Energy and velocity relationship for different average of serpenoid curve profiles α, β, ω_s for *n*=8.

The first set of tests consists on varying the serpenoid curve parameters described in Equation

(2). Considering angular speeds from $\omega_s = [0...4]$, we have analyzed the relationship between input energy and speed. Results are shown in Figure 4.

Two preliminary important considerations can be made. First of all, the relation to maintain between friction coefficients as a function of the snake morphology for *n*=8 (i.e. size, weight, mass distribution), and the serpenoid curve profile (i.e. undulation degree of the curve, speed, direction) is about $C_T / C_N = 0.1$. This friction ratio has been found from taking the simulation-results average for what choices of the serpenoid parameters α, β, ω_s the percentage of power loss is minimal. In this case, at maximum serpentine speed angular propagation of $\omega_s = 3 rad/s$, we achieved a maximum power loss about 30%. Increasing the ratio of friction coefficients under the same characteristics, the energy consumption also increases and consequently performance was compromised. Note that these friction ratios strictly depend on the snake's dynamics and the terrain characteristics. For experimental testing, we will have to explore and test different kind of materials that achieve the proper friction ratio dependent of different surfaces of motion. Using this relation, the snake is capable to move forward wasting the minimum energy and achieving the required velocity; in other words, this is the optimal relation between serpenoid curve parameters, snake morphology, and friction forces. In addition, also note in Figure 4 that using this optimal relationship, the snake linear velocity V_{cm} and the serpenoid curve speed ω_s are roughly proportional to each other.

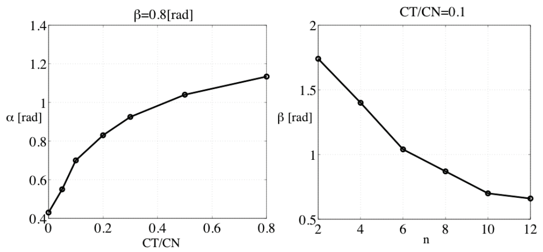


Figure 5: Serpenoid curve phase β and undulation degree α as a function of friction coefficients C_T / C_N and the snake's number of links *n*.

The tests performed in Figure 4, allowed us to find the optimal energy relation between:

- Energy consumption K_E and P_L.
- Velocity V_{cm} and ω_s.
- Friction C_T / C_N and parameters β and α.

Nonetheless note from Equation 2 that β and α are also strictly dependent on the snake configuration, this means, the snake number of links: *n*.

Thus, we have carried out a second series of tests varying the number of links (*n*) of the snake and changing the terrain surface, i.e., changing ratios of C_T / C_N. From the results depicted in Figure 5, we found that α is affected as a function of friction modification, whereas β depends on the number of links-*n*. Regarding the same simulation scenario in Figure 3 (circular snake motion using [illegible] 1.2°), this time we use optimal relationships to achieve energy-efficient and reliable snake locomotion. The optimal parameters configuration is: $\alpha = 0.7 rad, \beta = 0.8 rad$, $n = 8,\ C_T = 1,\ C_N = 10,\ \omega_s = 3 rad/s$.

Figure 6 illustrates the simulation results for open-loop tracking of the circular serpenoid path.

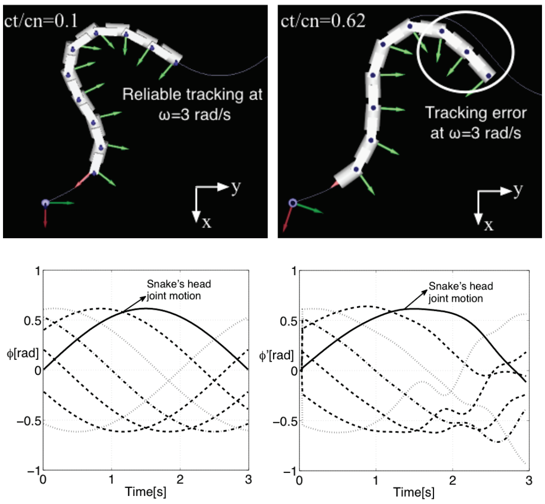


Figure 6: Cartesian (above) and Joint positions (below) between efficient snake locomotion (left) against improper parameters tuning (right), at snake speed of $\omega_s = 3 rad/s$.

Table 2: Efficiency of serpenoid locomotion for $C_T / C_N = 0.1$.

	Test-1	Test-2	Test-3	Test-4
ω_s [rad/s]	0.25	1.04	2.5	3
%error ϕ	0.59%	0.98%	1.15%	2.2%
K_E[J]	0.022	0.48	1.68	2.45
P_L [J/s]	9.2x10^{-4}	0.077	0.40	0.72
% P_L	4.11%	5.75%	23.80%	29.38%
Efficiency	0.95	0.83	0.76	0.70

The simulation scenario shown in Figure 3 is now compared to the efficient approach of serpentine locomotion showed in Figure 6. The joint position error ϕ' (inefficient approach) is about 12.5% against 2.2% for the efficient approach (ϕ). Table 2 consigns several simulations on testing efficient snake locomotion for different speed profiles.

4 CONCLUSIONS AND FUTURE WORK

The dynamics framework for the modelling and simulation of non-wheeled snake-like robots has been presented. Bio-inspired kinematics locomotion was efficiently integrated into our approach of achieving efficient serpentine locomotion at high-speeds. Simulation results depicted in Table II showed that our first hypothesis was indeed correct. Considering speeds up to 4 m/s we obtained efficient motion (less than 30%) of power loss due to friction. For speeds >4m/s, this efficiency decreases because of the increase of the angular speed, which also makes the friction force increases and subsequently generating more power loss average that makes the control effort too energetic. This speed boundary was obtained from several simulations performed in Figures 4 and 5. In conclusion, the key aspects in regarding energy efficient serpentine locomotion are basically synthesized as: 1). α is an increasing function of C_T/C_N, thus, the snake robot should undulate with larger amplitude when the friction ratio is larger (i.e. the snake-like robot tends to slip in the normal direction), 2). ω_s is basically a liner function of the linear speed V_{cm}, and, 3). β is a decreasing function of n.

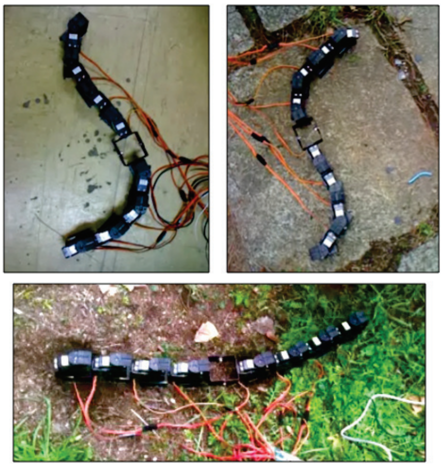


Figure 7: Locomotion testing experiments over different friction terrains.

These relationships are useful for determining the optimal control law for the snake robot. Upcoming work is oriented towards the full implementation of the hardware/software that allow the snake robot to be fully controlled. Using a first prototype depicted in Figure 7, our current work is focused on researching which materials and shapes of the snake's skeleton generate the proper friction and traction using our modeling approach.

ACKNOWLEDGEMENTS

This work is funded by the project ROBOCITY 2030 (S2009/DPI-1559).

REFERENCES

Chirikjian, G. S., and Burdick, J.W., 1990. An obstacle avoidance algorithm for hyper-redundant manipulators. *In IEEE International Conference on Robotics and Automation*, pag. 625–631, Cincinnati.

Dowling, K. J., 1997. Limbless Locomotion: Learning to Crawl with a Snake Robot. *PhD thesis, Carnegie Melon University*, Pittsburgh, USA.

Fu, K. S., Gonzalez, R.C., and Lee C.S.G., 1987. *Robotics: Control, Sensing, Vision, and Intelligence*. New York:McGraw-Hill.

Gray, J., and Lissmann, H., 1950. The kinetics of locomotion of the grass-snake, *J. Exp. Biol*., vol. 26, no. 4, pp. 354-367.

Hirose, S., 1993. Biologically inspired robots (snake-like locomotor and manipulator). *In Oxford University Press*.

Hirose, S., and Morishima, A., 1990. Design and control of a mobile robot with an articulated body, *Int. J. Robot. Res*., vol. 9, no. 2, pp. 99-114.

Kane, T., and Lecison, D., 2000. Locomotion of snakes: A mechanical 'explanation', *Int. J. Solids Struct*., vol. 37, no. 41, pp. 5829–5837.

Kamegawa, T., Matsuno, F., and Chatterjee, R., 2002. Proposition of Twisting Mode of Locomotion and GA based Motion Planning for Transition of Locomotion Modes of a 3-dimensional Snake-like Robot, *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 1507

Ostrowski, J., 1995. The Mechanics of Control of Undulatory Robotic Locomotion. *PhD thesis, California Institute of Technology*.

Prautsch, P., and Mita, T., 1999. Control and Analysis of the Gait of Snake Robots. *Proceedings of the IEEE International Conference on Control Applications*, Kohala Coast-Island of Hawaii, Hawaii, USA, pp. 502.

Transeth, A., and Pettersen, K.Y., 2006. Developments in snake robot modeling and locomotion, in *Proc. IEEE Int. Conf. Control, Automation, Robotics and Vision*, pp. 1393–1400.

DECENTRALISED ACTIVE CONTROLLER

Chiheb Ameur Abid and Belhassen Zouari
LIP2 Laboratory, University of Tunis, El Manar 2, Tunis, Tunisia
chiheb.abid@gmail.com, belhassen.zouari@fst.rnu.tn

Keywords: Supervisory control theory, Coloured Petri nets, Discrete event systems, Distributed systems.

Abstract: This paper deals with the synthesis problem of controllers based on Coloured Petri nets in a decentralized setting. An approach is proposed to derive a decentralized controller consisting of a set of local controllers. Each local controller, modeled by a CP-net, observes and controls a portion of the plant model. Communication between local controllers is performed only when it is necessary allowing the appropriate local controller to restrict the behavior of its observed portion in order to meet the control specification.

1 INTRODUCTION

The supervisory control theory for discrete-event systems (DES) aims to control a plant model by the automatic synthesis of a controller (Ramadge and Wonham, 1989). Due to their easily understood graphical representation and well-formed mathematical formalism, Petri nets have been widely used to investigate this theory (Giua and DiCesare, 1994; Holloway et al., 1997; Ghaffari et al., 2003).

In practice, implementing a centralised controller is not always feasible since a plant model can be a collection of several semi-autonomous components geographically dispersed. As a solution, the decentralised control (Rudie and Wonham, 1992) allows to looks for a fixed number of controllers such that each controller observes and controls a portion of the plant. The decentralised control can be performed with communication (Guan and Holloway, 1995), without communication (Basile et al., 2007), or by introducing a central coordinator (Chen and Baosheng, 1991).

Although, they seem to be powerful enough to describe complex systems in a manageable way, and their expressiveness capabilities, few works have used Coloured Petri nets (CP-nets) (Jensen et al., 2007) to investigate the control problem. In (Makungu et al., 1999), the problem of forbidden states is tackled for controlled CP-nets. In our previous works (Abid and Zouari, 2008; Abid et al., 2010), we have proposed a first approach combining both theory of regions and CP-nets in which the obtained controller is reduced to one place. Also, we have optimised the same approach for symmetric systems to tackle the state explosion problem. In a second approach, we generate a CP-net controller as an active process without using the theory of regions.

The work presented in this paper addresses the supervisory control problem in a decentralised setting. The considered plant model consists of several modules, modelled by CP-nets, which communicate through the synchronisation of common actions. Based on global control specifications expressed in terms of forbidden states, we propose to build a *decentralised active controller*. Such a controller consists of a set of communicating *local controllers* associated with modules. The role of a each local controller is to observe and to control its associated module.

The remainder of the paper is organised as follows. Section 2 introduces the CP-net models. Section 3 provides the computation of necessary information related to the building of a decentralised active controller. Section 4 deals with the building of a CP-net model representing a local controller. Finally, section 5 summarizes the main conclusions and perspectives of this work.

2 PRELIMINARIES

In this paper, the plant model to be considered is assumed to be made up of several communicating modules. The communication is ensured via the execution of common actions. Further, we assume that each module corresponds to a *Resource Allocation System* (RAS). A RAS, as defined in (Reveliotis et al., 1997), corresponds to a set of resources, that can be of many types and available in any amount of copies, and a set of generic processes, that can be instantiated in a collection of processes.

Formally, a RAS is a tuple $(R, GP, GetR, PutR)$ where:

- $R = (RT, Rnb)$ is the resource specification, s.t.RT is the finite set of resource types, and $Rnb \in Bag(RT)^1$ is a multiset on RT representing the amount of available copies of each resource type.
- GP is the finite set of generic processes.
  A generic process $Gi =< P_{si}, F_i, C_i >$ is defined by a finite state machine (FSM) as follows:
  - $P_{si} = \{p_{0i}, p_{i1}, p_{i2}, ..., p_{il(s)}\}$ is the finite set of states, where p_{0i} represents the 'idle' place of the generic process. The 'idle' place is the initial place of whole process instance.
  - $F_i : P_{si} \to P_{si}$ is the flow relation representing the state transitions.
  - C_i is the finite set of identities of process instances.
- $GetR : F_i \to Bag(R)$ is the resource allocation function
- $PutR : F_i \to Bag(R)$ is the resource restitution function

A RAS can be modelled by a CP-net. Such a CP-net is built from a collection of FSMs (representing the generic processes), a resources place, and connection arcs related to the resource management.

A marked CP-net N representing a RAS, where $N =< P, T, C, W^-.W^+, \Phi, M_0 >$, is defined as following:

- $P = P_S \cup \{p_R\}$ is a finite set of places where $P_S = \bigcup_{i=1}^{|GP|} P_{si}$, and p_R represents the resources place such that $P_{si} \cap P_{sj} = \emptyset$ and $P_S \cap \{p_R\} = \emptyset$.
- T is a set of transitions recursively built as follows:
  Initially $T = \emptyset$, then $T = T \cup \{f_{ij}\}; i = 1, .., |GP|; j = 1, .., |F_i|$.
  The elements of T may be lexicographically renamed so as we can use the notation $T = \{t_i\}; i = 1, .., |T|$.
- $C = \bigcup_{i=1}^{|GP|} C_i \bigcup_{i=1}^{|GP|} (C_i \times C_r) \cup C_r$ where C_r is a colour domain of $p_R, C_r = RT$.
- $W^- = W_S^- \cup W_R^-$, where $W_S^- : (P_s \times T) \to \{0, \bigcup_{i=1}^{|GP|} X_i\}$ such that X_i is a variable defined on C_i, and $W_R^- : (\{p_R\} \times T) \to Bag(C_r)$ such that $\forall t \in T, W_R^-(p_R, t) = GetR(f_{ij})$.
- $W^+ = W_S^+ \cup W_R^+$, where $W_S^+ : (P_s \times T) \to \{0, \bigcup_{i=1}^{|GP|} X_i\}$ such that X_i is a variable defined on C_i, and $W_R^+ : (\{p_R\} \times T) \to Bag(C_r)$ such that $\forall t \in T, W_R^+(p_R, t) = PutR(f_{ij})$.
- M_0 the initial marking is a function defined on P, such that $M_0(p) = Bag(C(p))$, for all $p \in P$. M_0 is defined as follows:
  $\forall p \in P_S \setminus (\bigcup_{i=1}^{|GP|}\{p_{0i}\}); M_0(p) = 0; M_0(p_R) = Rnb$ and $M_0(p_{0i}) \geq 1$.
- Φ is a function which associated a guard with any transition. By default $\Phi(t)$ is *true* for any transition t.

We consider a decentralised system consisting of several modules that communicate. We assume that each module corresponds to a RAS modelled by a CP-net. The communication between CP-nets can be performed through two possible forms. The asynchronous form by means of shared places, or in synchronous form by means of shared transitions. Since communication via shared places can be transformed into synchronisation via shared transitions, it suffices to support the latter (Christensen and Petrucci, 2000). Modular CP-nets (Christensen and Petrucci, 1995) allow to model efficiently such systems. The latter models introduce structure to CP-nets by letting modules specified separately. For further details about modular CP-nets, one may refer to original papers (Christensen and Petrucci, 1995).

Let be $K = \{1, ..., m\}$. A modular CP-net is a pair $MCPN = (\{N_k | k \in K\}, TF)$ satisfying:

- $\{N_k | k \in K\}$ is a set of modules such that $\forall k \in K, N_k =< P_k, T_k, C_k, W_k^-.W_k^+, \Phi_k >$ is a CP-net,
- $TF \subseteq 2^T$ is a finite set of transition fusion sets where $T = \bigcup_{k \in K} T_k$.

In a modular CP-net, all transitions belonging to the same transition fusion set are fired as one indivisible action sharing the values assigned by a common *binding*. The binding is a function used to assign only one value for a variable of a transition fusion set, i.e. a variable name refers to the same value for all transitions in a transition fusion set.

Example 1. *Throughout this paper, we consider the producer-consumer problem. In this problem, there are two kinds of processes, namely producers and consumers, sharing a stock (a resource). Figure 1(a) gives a CP-net modelling this problem.* $C_1 = \{pr\}$, $C_r = \{f, o\}$ *and* $C_2 = \{co\}$ *are the colour classes of this net. A token pr is used to model the behaviour of a producer, while a token co allows to model the behaviour of a consumer. The token f indicates the number of free places in the stock, while the token o indicates the produced objects. We assume that the stock capacity is two. The colour domains of places are:* $C(a1) = C(a2) = C_1$, $C(r) = C_r$ *and*

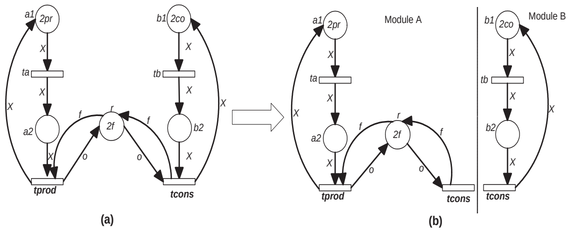


Figure 1: Example of CP-nets.

$C(b1) = c(b2) = C_2$.
As it is illustrated in Figure 1, the latter CP-net can be decomposed into two modules A and B. These modules are both synchronised on transition $tcons$*. The two shared transitions* $tcons$ *of module A and B constitute a transition fusion set.*

3 DECENTRALISED ACTIVE CONTROLLER MODEL

In this section, we present the synthesis approach of a decentralised active controller. As a plant model, we consider a decentralised DES consisting of several modules modelled by a modular CP-net. We assume that every module is structured on a set of generic processes sharing a set of resources. As our aim is to resolve the state forbidden problem, the control specification is then expressed in terms of *forbidden markings*. In a first step, we use the reachability graph to determine the *dangerous markings* as well as transitions to be disabled from firing by the local controllers, called the *forbidden transitions*. The dangerous markings are markings from which the control actions must be applied by disabling some forbidden transitions.

3.1 Architecture of a Decentralised Active Controller

A decentralised controller consists of a set of local controllers where each one is associated with one module of the plant model. The preventing of a transition from firing in a dangerous marking is ensured by the local controller associated with the module to which the transition belongs. Each local controller has to observe permanently the evolution of its associated module by tracking its current local marking. However, since a control action depends mainly on the reached global state of the plant model (the entire system), then a local controller requires to observe some evolutions of the plant model. Such a global observation is performed, when it is necessary, by communicating with the other local controllers. In summary, each derived local controller associated with a module has three main features:

- Observing the current local state of its module,
- Communicating with the other local controllers to determine the global state of the plant model when it is necessary,
- Performing control actions by preventing the firing of some transitions of its associated module.

For sake of simplicity, we propose to decompose a local controller into two submodules. The first one, called *communication submodule*, observes and diffuses the current local marking of its associated module, while the second submodule, called *executor submodule*, performs the control actions by preventing some transitions of its associated module from firing. For this purpose, the executor submodule must be able to identify when the overall plant model reaches a dangerous marking. This ability is ensured by communicating with communication submodules. It is worth noting that an executor submodule is associated with a module only when there exists at least one transition of the module to be disabled by the controller.

3.2 Determining the Admissible Behaviour

The supervisory control theory classifies the transitions into two categories. First category consists of the *controllable transitions* which may be disabled when it is necessary. In contrast, the transitions belonging to the second category, called *uncontrollable transitions*, are beyond any control procedure. Hence, we assume that the transition set of the plant model is partitioned into two disjoint subsets: the set of controllable transitions and the set of uncontrollable ones.

The disabling of a transition is performed in a *dangerous marking* from which the firing of the transition leads to a forbidden marking. So that, we have to identify the set Ω of *state-transitions* to be disabled. Every element of Ω is a couple $(M,(t,c))$ where M is a dangerous marking, and (t,c) is a controllable coloured transition, called *forbidden transition*, such that the firing of t with colour c from M yields to a forbidden marking.

The set Ω of state-transitions is computed from the reachability graph of the plant model. The basic idea is to remove forbidden nodes. Nodes becoming unreachable from the initial marking and the non coreachable markings must be removed from the admissibility graph. The identification of dangerous and forbidden nodes is performed according to the following rules:

- a marking is qualified as dangerous if it has at least one output arc where its destination is a forbidden marking,
- a marking is qualified as forbidden when it has no output arcs and it is not a final marking, or it is a dangerous marking and it has at least one output arc labelled by an uncontrollable transition,
- every forbidden marking must be removed with its input and output arcs.

In (Abid et al., 2010), an algorithm is proposed for the computation of the admissibility graph and the set Ω of state-transitions.

Example 2. *Consider the example of producer-consumer. Its reachability graph is illustrated by Fig. 2. Assume that we want to restrict the stock capacity to hold only one object. Therefore, the initially specified set of forbidden markings is*

$$FM = \{M10, M13, M14, M15, M16, M17\}.$$

By determining the admissible behaviour, we obtain the admissibility graph of Fig. 3 and the set of state-transitions $\Omega = \{(M6,(tprod,<pr,o>)),(M11,(tprod,<pr,o>)),(M12,(tprod,<pr,o>))\}$, where $M6 =< 1,1,1,1,1,0 >$, $M11 =< 1,1,1,1,0,1 >$ and $M12 =< 0,2,1,1,0,1 >$. Thus, the coloured transition $(tprod,<pr,o>)$ has to be disabled in markings $M6$, $M11$ or $M12$.

Let d be a dangerous state. We use $FT(d)$ to denote the set of coloured forbidden transitions related to d. Each element of $FT(d)$ is a coloured transition (t,c) such that the firing of (t,c) in d leads to a forbidden or inadmissible state. Let k be a module, we use FT_k to to determine the set of forbidden transitions of module k. Let DS be the set of dangerous states. FT may be viewed as an application defined as follows:

$$\begin{aligned} FT: \quad & DS \to 2^T \\ & d \mapsto \{(t,c) | (d,(t,c)) \in \Omega\} \end{aligned}$$

Basically, a local controller can only observe the behaviour of its associated module. Since each local controller has to communicate with the other local controllers in order to build and identify dangerous markings, we introduce the notion of *dangerous local marking*. A dangerous local marking of a module is a local marking from which we can build a dangerous marking.

Let M_k be a local marking of module k.
M_k is a local dangerous marking iff $\exists M' \in [M_0\rangle$: $M'_k = M_k \wedge M'$ is a dangerous marking.

For every local marking associated with a module k, we make available the following data:

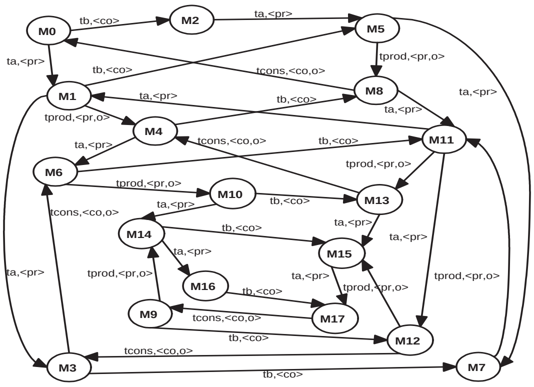


Figure 2: Example of a reachability graph.

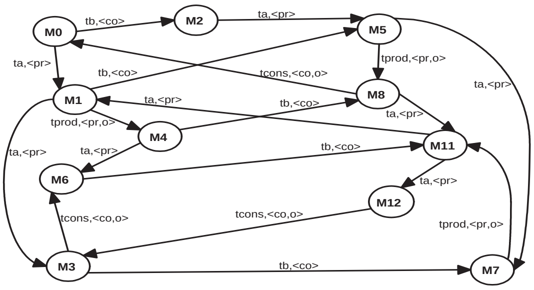


Figure 3: Example of admissibility graph.

- the set of dangerous markings,
- the set of dangerous local markings of module k,
- the set FT_k of forbidden transitions of module k and their correspondent dangerous marking.

Example 3. *From the set Ω computed in previous example, we determine the following sets:*
The set of dangerous local markings of module A is $SDLM_A = \{< 1,1,1,1 >, < 0,2,1,1 >\}$.
In module A, there exists only one forbidden transition:
$FT_A(M6) = FT_A(M11) = FT_A(M12) = \{(tprod,< pr,o >)\}$.
The set of dangerous local markings of module B is $SDLM_B = \{< 1,0 >, < 0,1 >\}$.
In module B, there is no forbidden transitions, then: $FT_B(M6) = FT_B(M11) = FT_B(M12) = \emptyset$.

4 A LOCAL CONTROLLER MODEL

Now, we describe the CP-net model of a local controller. The key idea of the decentralised active controller approach is that a local controller must be able to detect the reaching of a dangerous state. Once a

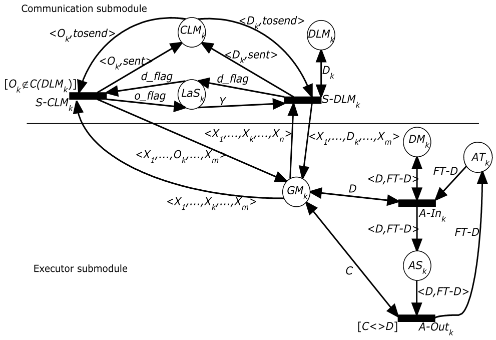


Figure 4: CP-net modelling a local controller.

dangerous marking is detected, the local controller removes appropriate authorisations in order to disable the firing of some forbidden transitions of its associated module. Basically, a local controller has to handle, permanently, information about the current local marking of its module. In contrast, a global marking is determined for a local controller through communication. Intuitively, a local controller diffuses its observed local state to the other local controllers when it detects that the observed local state may be a part of a dangerous global state.

As precited, a local controller can be seen as the composition of two submodules, namely the communication and the executor submodules. The main purpose of the first submodule is to determine and diffuse the current local marking of its associated module to other local controllers, while the executor submodule performs the control actions. More precisely, the communication submodules send the determined local markings to every executor submodule. Then, each executor module can determine whether the plant model reaches a dangerous global marking from which some transitions have to be prevented from firing.

The communication submodule diffuses the current local state of its associated module according to the following rules:

- Its associated module reaches a dangerous local state from another local state (dangerous or not),
- Its associated module reaches a non dangerous local state from a dangerous local state.

Figure 4 describes the CP-net modelling a local controller. Let us consider a DES, made up of m modules, modelled by a modular CP-net $MCPN = (\{N_k | k \in K = \{1,...,m\}\}, TF)$, where $\forall k \in K, N_k =< P_k, T_k, C_k, W_k^-, W_k^+, M_{0,k}, \Phi_k >$. In a first step, we detail the formal definition of the communication submodule. Secondly, we define the executor CP-net submodule.

Step 1: the communication submodule. Let us consider the CP-net N_k modelling module $k \in K$. The model representing module k connected to its associated communication submodule is a CP-net $N1_k$ obtained from N_k, so that $N1_k =< P1_k, T1_k, C1_k, W1_k^-, W1_k^+, M1_{0,k}, \Phi1_k >$.
$P1_k = P_k \cup \{CLM_k, DLM_k, LaS_k\}$, where:

- CLM_k represents the current local marking of module k,
- DLM_k contains the set of of dangerous local markings associated with module k,
- LaS_k indicates whether the last communicated current local marking of module k is dangerous, or not.

$T1_k = T_k \cup \{S-DLM_k, S-CLM_k\}$, where:

- $S-DLM_k$ is a synchronised transition allowing to send a dangerous local marking,
- $S-CLM_k$ is a synchronised transition. It allows to send a non dangerous current local marking when it is reached from a dangerous global marking.

Now, we introduce the following additionnal colour classes:
$C_{num} = \{1,2,\ldots,MaxInt\}$ is a colour class representing a set of finite positive integers. The elements of C_{num} are used to model the occurrences of some given tokens. We assume that $MaxInt$ is large enough to be greater than the bound of maximum occurrences of any token in a reachable marking,
$C_{status} = \{tosend, sent\}$ is a colour class where the object *sent* (resp. *tosend*) is used to identify whether a current local marking is already sent to all executor submodules (resp. not yet sent),
$C_{flag} = \{d\_flag, o\_flag\}$ where the object d_flag (resp. o_flag) is used to indicate whether the last diffused local marking is dangerous (resp. not dangerous).
Every added place is characterised by a colour domain and an initial marking defined as follows:

- Place LaS_k allows to hold information indicating whether the last communicated current local marking is dangerous or not. For this purpose,we use the colours of C_{flag}. Thus, $C(LaS_k) = C_{flag}$. Initially, place LaS_k is empty.
- Place CLM_k holds the current local marking of module k, and information whether this local marking is communicated. A local marking is represented by a tuple made up of counters. Each counter holds the information about the occurrence of tokens in a given place (according to the lexical order) among process places and the resource place. Such a representation is obtained

through a Cartesian product performed on the basis of the number l of process classes in module k, the number of places x_i per a process i and the resource class $C_{r,k}$. More precisely, a local marking is an element of the following colour class:

$$C_{LM,k} = \bigotimes_{i=1}^{l} \bigotimes_{j=1}^{x_i} C_{num} \bigotimes_{u=1}^{|C_{r,k}|} C_{num}.$$

In addition to the current local marking, place CLM_k holds information indicating whether the current local marking is already communicated or not. It is performed by using elements of the colour class C_{status}. Thus, the domain class of place CLM_k is $C(CLM_k) = C_{LM,k} \times C_{status}$.

Place CLM_k is always mono-marked and its initial marking $M_{0,k}(CLM_k)$ is determined from the initial marking of module k.

- Place DLM_k holds the set of dangerous local markings of module k. Each marking is represented in identical way as for place CLM_k. So that $C(CLM_k) = C_{LM,k}$. Initially, place DLM_k holds the set $SetDLM_k$ which represents the dangerous local markings of module k, i.e. $M_{0,k}(DLM_k) = SetDLM_k$. This place is only read accessed.

Now, we give the colour functions associated with the arcs connecting places to transitions.

- As place CLM_k holds the current local marking of module k modelled by CP-net N_k, then it has to be connected to every transition of T_k in order to keep its information up to date. This is because the firing of every transition of T_k may update the current local marking of module k. Place CLM_k is connected to every transition of T_k with an input arc (reading marking) and output arc (updating marking) as follows:
  $\forall t \in T_k$,
  $W1_k^-(CLM_k,t) =< X_{1,1},\ldots,X_{i,j},Y_1,\ldots,Y_u >= < X >$, where $X_{i,j}$ is a variable defined on C_{num} computing the number of tokens in place i of the process type j ($p_{ij} \in P_k$), and Y_u is a variable defined on C_{num} reading the occurrence of colour u in resource places.
  $\forall t \in T_k$,
  $W1_k^+(CLM_k,t) =< X'_{1,1},\ldots,X'_{i,j},Y'_1,\ldots,Y'_u >= < X >$, where $X'_{i,j}$ and Y'_u are variables defined on C_{num} and determined as follows:
  $X'_{i,j} = X_{i,j} - \chi$, where $\chi = W^+(p_{ij},t) - W^-(p_{ij},t)$,
  $Y'_u = Y'_u - \xi$, where ξ is computed as follows:
  $$\begin{cases} W_k^-(r,t) = \sum\limits_i \alpha_i.r_i, \\ W_k^+(r,t) = \sum\limits_i \alpha'_i.r_i \end{cases} \Rightarrow \xi = \alpha'_u - \alpha_i.$$

- The colour functions related to $S-DLM_k$:
  $W1_k^+(DLM_k,S-DLM_k) = W1_k^-(DLM_k,S-DLM_k) =< D_k >$;
  $W1_k^-(CLM_k,S-DLM_k) =< D_k,tosend >$;
  $W1_k^+(CLM_k,S-DLM_k) =< D_k,sent >$;
  $W1_k^-(LaS_k,S-DLM_k) =< Y >$;
  $W1_k^+(LaS_k,S-DLM_k) =< d\_flag >$;
  where $D_k \in C_{LM,k}$ and $Y \in C_{flag}$.
  The colour functions related to $S-CLM_k$:
  $W1_k^-(CLM_k,,S-CLM_k) =< O_k,tosend >$;
  $W1_k^+(CLM_k,,S-CLM_k) =< O_k,sent >$;
  $W1_k^-(LaS_k,S-CLM_k) =< d\_flag >$;
  $W1_k^+(LaS_k,S-CLM_k) =< o\_flag >$;
  where $[O_k \in C_{LM,k}]$.
  Transition $S-CLM_k$ is associated with the guard $[O_k \notin SetDLM_k]$.

Step 2: the executor submodule. Now, we present the features of the executor submodule. It is obtained from the CP-net $N1_k =< P1_k,T1_k,C1_k,W1_k^-,W1_k^+,M1_{0,k},\Phi1_k >$, modelling module k with its associated communication submodule.

The model representing a module k and its local controller is a CP-net N_k^* obtained from $N1_k$, so that $N_k^* =< P_k^*,T_k^*,C_k^*,W_k^{*-},W_k^{*+},M_{0,k}^*,\Phi_k^* >$, where:
$P_k^* = P1_k \cup \{GM_k,DM_k,AS_k,AT_k\}$, where:

- GM_k represents the global marking of the overall plant model,
- DM_k holds the set of dangerous global markings,
- AS_k represents the alert state of the local controller associated with module k,
- AT_k contains the set of authorisations for forbidden transitions of module k.

$T_k^* = T1_k \cup \{A-IN_k,A-OUT_k\} \bigcup_{i\in K\setminus\{k\}}\{S-DLM_{k,i}\} \bigcup_{i\in K\setminus\{k\}}\{S-CLM_{k,i}\}$, where:

- $A-IN_k$ allows to enter the alert state,
- $A-OUT_k$ allows to quit the alert state,
- $\bigcup_{i\in K\setminus\{k\}}\{S-DLM_{k,i}\}$ is a set of shared transitions. The firing of a transition $S-DLM_{k,i}$ allows the executor submodule k to receive a diffused dangerous local marking from the communication submodule i.
- $\bigcup_{i\in K\setminus\{k\}}\{S-CLM_{k,i}\}$ is a set of shared transitions. The firing of a transition $S-CLM_{k,i}$ allows module k to receive a non dangerous local marking from the communication submodule i.

The colour domain and the initial marking associated with every added place are as follows:

- Place GM_k holds the marking of the overall plant model. Its colour domain is a Cartesian product of colour classes representing local markings of individual modules. Thus, the colour domain of GM_k is defined as follows:

  $$C^*(GM_k) = \prod_{i \in K} C_{LM,i}$$

  Initially, the marking of place GM_k represents the initial marking of the overall plant model.

- Place DM_k maintains the set of state-transitions of module k. Its colour domain is

  $$C_k^*(DM_k) = C^*(GM_k) \times C_{FT,k}$$

  , where $C_{FT,k}$ denotes the colour class representing forbidden transitions of module k. This place is only read accessed.

- Place AS_k indicates that the controlled module k is in alert state. Its marking indicates the global marking of the controlled system and the transition to be prevented from firing.

  $$C_k^*(AS_k) = C^*(GM_k)$$

  Initially, place AS_k is empty as all forbidden transitions are authorised.

- Place AT_k is connected to every forbidden transition by one input/ output arc in order to check the presence of the associated firing authorisation: $\forall t \in FT_k, W_k^{*+}(AT_k,t) = W_k^{*-}(AT_k,t) =< Xt >$, where Xt is defined on $C_{FT,k}$ and represents the identity of the coloured forbidden transition. Further, we associate with every forbidden transition the following guard expression:

  $$[\bigvee_{c \in C(t)} [\bigwedge_{X \in Var(t) \setminus \{Xt\}} X = X(c) \wedge Xt = id\_ft(t,c)]]$$

  where id_ft is a function which maps a forbidden coloured transition (t,c) to an element of $C_{FT,k}$ representing the identity of (t,c). This associated guard expression allows to associate the appropriate authorisation for a coloured transition.

The colour functions of the arcs connecting the controller places to a subset of T_k and to the transitions $A - In_k$ and $A - Out_k$ are defined as follows:
$W_k^{-*}(DM_k, A - In_k) = W_k^{*+}(DM_k, A - In_k) =< D, FT - D >$
$W_k^{-*}(GM_k, A - In_k) = W_k^{*+}(GM_k, A - In_k) =< D >$
$W_k^{-*}(AT_k, A - In_k) =< FT - D >$
$W_k^{+*}(AS_k, A - In_k) =< D, FT - D >$
$W_k^{+*}(AS_k, A - Out_k) =< D, FT - D >$
$W_k^{-*}(GM_k, A - Out_k) =< C >$
$W_k^{+*}(AT_k, A - Out_k) =< FT - D >$.
D and $C \in C_{GM}$, $FT - D \in C_{FT,k}$.
Transition $A - Out_k$ is associated with the predicate $[C \neq D]$. For every $i \in K$, place GM_k is connected to the $S - DLM_{k,i}$ and $S - CLM_{k,i}$ allowing to update the local marking of module i in place GM_k. The associated colour functions are defined as follows:
$W_k^{-*}(GM_k, S - DLM_{k,i}) =< X_1, ..., X_i, ..., X_m >$;
$W_k^{+*}(GM_k, S - DLM_{k,i}) =< X_1, ..., X_{i-1}, D_i, ..., X_m >$;
$W_k^{-*}(GsM_k, S - CLM_{k,i}) =< X_1, ..., X_i, ..., X_m >$;
$W_k^{+*}(GM_k, S - CLM_{k,i}) =< X_1, ..., X_{i-1}, D_i, ..., X_m >$.
In summary, the controlled model, obtained from $MCPN$, is modelled by the modular CP-net $MCPN^* = (\{N_k^* | k \in K\}, TF^*)$, where $TF^* = TF \cup \{FS - CLM_k | i \in K\} \cup \{FS - DLM_k | k \in K\}$, where:

- $\forall k \in K, FS - CLM_k$ is a transition fusion set such that $FS - CLM_k = \{S - CLM_{k,i} | i \in K\}$;
- $\forall k \in K, FS - DLM_k$ is a transition fusion set such that $FS - DLM_k = \{S - DLM_{k,i} | i \in K\}$.

Remark 1. *The transitions $S - CLM_k$, $S - DLM_k$, $A - In_k$ and $A - Out_k$ must have special high priority over all the transitions of the plant model and are fired immediately when enabled.*

Example 4. *In our considered example, only $tprod$ which belongs to model A is a forbidden transition. Thus, the local controller associated with module A consists of the communication and the executor submodules, while the controller associated with module B consists only of a communication submodule. The different features of the local controller associated with module A are:*
$C_A^*(CLM_A) = C_{num} \times C_{num} \times C_{num} \times C_{num} \times C_{status}$
The first and the second elements respectively define the number of producers in places $a1$ and $a2$. The third and the fourth elements respectively handles the occurence number of f and o in place r.
$C_{FT,A} = \{tprodPro\}$ *where $tprodPro$ denotes the authorisation for the firing of transition $tprod$ with colour $< pr, o >$.*
$C_A(GM_A) = C_{num} \times C_{num} \times C_{num} \times C_{num} \times C_{num} \times C_{num}$
where the first four elements allow to handle the local marking of module A, while the fifth and the sixth elements handle the local marking of module B.
The initial markings of places are:
$M_{0,A}^*(CLM_A) =< 2, 0, 2, 0, sent >$
$M_{0,A}^*(DLM_A) =< 1, 1, 1, 1 > + < 0, 2, 1, 1 >$
$M_{0,A}^*(AT_A) = tprodPro$
$M_{0,A}^*(GM_A) =< 2, 0, 2, 0, 2, 0 >$
$M_{0,A}^*(DM_A) =< 1, 1, 1, 1, 1, 0 > + < 1, 1, 1, 1, 0, 1 > + < 0, 2, 1, 1, 0, 1 >$
$\forall t \in T_A, W_A^{*-}(CLM_A, t) =< X_{1,1}, X_{1,2}, Y_1, Y_2, Z >$
$W_A^{*+}(CLM_A, ta\{pr\}) =< X_{1,1} - 1, X_{1,2} + 1, Y_1, Y_2, tosend >$

$W_A^{*+}(CLM_A, tprod\{< pr, o >\}) =< X_{1,1} + 1, X_{1,2} - 1, Y_1 - 1, Y_2 + 1, tosend >$
$W_A^{*+}(CLM_A, tconsum\{o\}) =< X_{1,1}, X_{1,2}, Y_1 + 1, Y_2 - 1, tosend >$.

5 CONCLUSIONS

In this paper, we have investigated the problem of designing a decentralised controller based on CP-nets for the forbidden states problem. Considering global specifications, the obtained decentralised controller consists of a set of communicating local controllers, where each one is able to observe permanently the current state of its associated module. Communication between local controllers allows, when it is necessary, a local controller to determine the global state of the entire plant model. Each local controller has a fixed structure whatever the system to control. So that, the synthesis approach is reduced to the determination of the parameters related to each local controller. It is performed mainly by determining the admissible behaviour. Further, this makes the proposed approach easier to understand, and to implement in practice.

REFERENCES

Abid, C. A., Zairi, S., and Zouari, B. (2010). Supervisory control and high-level petri nets. In Pawlewski, P., editor, *Petri Nets: Applications*, chapter 14, pages 281–306. INTECH, Vienna, Austria.

Abid, C. A. and Zouari, B. (2008). Synthesis of controllers using symbolic reachability graphs. In *Proceedings of 9th International Workshop of Discrete Event Systems (WODES'08)*, pages 314–321, Göteborg.

Basile, F., Giua, A., and Seatzu, C. (2007). Supervisory control of petri nets with decentralized monitor places. In *26th American Control Conference (ACC07)*, New York, USA.

Chen, H. and Baosheng, H. (1991). Distributed control of discrete event systems described by a class of controlled petri nets. In *Preprints of IFAC International Symposium on Distributed Intelligence Systems*.

Christensen, S. and Petrucci, L. (1995). Modular state space analysis of coloured petri nets. In *Proceedings of the 16th International Conference on Application and Theory of Petri Nets*, pages 201–217, London, UK. Springer-Verlag.

Christensen, S. and Petrucci, L. (2000). Modular analysis of petri nets. *Comput. J.*, 43(3):224–242.

Ghaffari, A., Rezg, N., and Xie, X. (2003). Design of live and maximally permissive petri net controller using the theory of regions. In *Proceedings of IEEE Transactions on Robotics and Automation*, volume 19, pages 137–142, Aarhus.

Giua, A. and DiCesare, F. (1994). Petri net structural analysis for supervisory control. *IEEE Transactions on Robotics and Automation*, 10(2):185–195.

Guan, X. and Holloway, L. E. (1995). Distributed discrete event control structures with controller interactions. In *In proceedings of 1995 American Control Conference*, pages 3151–3156.

Holloway, L. E., Krogh, B. H., and Giua, A. (1997). A survey of petri net methods for controlled discrete eventsystems. *Discrete Event Dynamic Systems*, 7(2):151–190.

Jensen, K., Kristensen, L. M., and Wells, L. (2007). Coloured petri nets and cpn tools for modelling and validation of concurrent systems. *Int. J. Softw. Tools Technol. Transf.*, 9(3):213–254.

Makungu, M., Barbeau, M., and St-Denis, R. (1999). Synthesis of controllers of process modeled as coloured petri nets. *Journal Discrete Event Dynamic Systems Theory Applications Kluwer Academic Publishers*, Vol. 9(No. 2):147–169.

Ramadge, P. and Wonham, W. (1989). The control of discrete event systems. In *Proceedings of IEEE*, pages 81–98. Special Issue on Discrete Event Dynamic Systems.

Reveliotis, S., Lawley, M., and Ferreira, P. (1997). Polynomial-complexity deadlock avoidance policies for sequential resource allocation systems. *IEEE transactions on automatic control*, 42(10):1344–1357.

Rudie, K. and Wonham, W. (1992). Think globally, act locally: Decentralized supervisory control. *IEEE Trans. on Automatic Control*, 37(11):1692–1708.

A CONSTRAINED FINITE TIME OPTIMAL CONTROLLER FOR THE DIVING AND STEERING PROBLEM OF AN AUTONOMOUS UNDERWATER VEHICLE

George Nikolakopoulos, Nikolaos J. Roussos
Hellenic Navy Research Center (GETEN), Ministry of Defence, Papagos, Athens, Attika, Greece
gnikolak@ece.upatras.gr, ietan2@ath.forthnet.gr

Kostas Alexis
Department of Electrical and Computer Engineering, University of Patras, Rio Achaias, Greece
kostalexis@ece.upatras.gr

Keywords: AUV, Constrained finite time, Optimal control.

Abstract: In this paper a Constrained Finite Time Optimal Controller (CFTOC) is designed and applied to the diving and steering problem of an Autonomous Underwater Vehicle. The non–linear model of the AUV is presented and the decoupled linear models for the steering and diving motions of the vehicle are derived, based on certain modeling assumptions and simplifications, while the cruising speed of the vehicle is considered to be small and constant. The proposed control scheme has the merit to take under consideration: a) the mechanical and physical constrains of the AUV, b) uncertainties produced from modeling errors and environmental noise, c) constrains in the motors, and produce an optimal controller for the vehicle that will guarantee the stability of the closed loop system. The proposed CFTO–controller is applied to simulation studies and relevant simulation results are presented that prove the efficacy of the proposed scheme.

1 INTRODUCTION

In the last years there was a strong interest towards the development of Autonomous Underwater Vehicles (AUV) and Remotely Operated Vehicles (ROV). These classes of underwater vehicles are intended to provide simple, long–range, low–cost measurements of environmental data or surveillance studies. Although ROVs have been utilized in the past in many applications, in the recent years there has been a growing demand for the utilization of AUVs as these vehicles are superior to the ROVs, are completely autonomous and are not suffering from the demand of high operating costs, dedicated cables for carrying the data links between the vessel and the ROV and experienced crew to guide the vehicle (Yuh, 2000; E. An, 2001; Foresti, 2001).

The superiority of AUVs and their complete autonomy are generating more demands for modeling approaches and applied control algorithms as more accurate and fast control actions should be applied to the vehicle to improve the overall performance. In the utilized model of an AUV, when other factors are taken under consideration, including: a) parametric uncertainties such as added mass, hydrodynamic coefficients, lift and drag forces, b) highly and coupled non linearities, and c) environmental disturbances like ocean currents and wave effects, the problem of applying a most suitable control law is widely increased.

In the area of mathematical modeling for AUVs, there have been extended analytical approaches with the main variation being the level of association between the hydrodynamic phenomena and the underwater rigid body dynamics. In the general case the mathematical models contain hydrodynamic forces and moments expressed in terms of a set of hydrodynamic coefficients, therefore it is of paramount importance to *a priori* know the true values of these coefficients to control the AUV accurately. In most of the cases experimental measurements are needed to tune the hydrodynamic parameters of the models while the relative literature is providing sufficient references and methodologies for calculating these parameters in various types of underwater vehicles.

In the area of controlling AUVs, until now various classical approaches have been utilized. More specifically many control strategies have been appeared in the literature for the diving and steering control of an AUV such as: Optimal control (Field, 2000), Neural Networks (Kawano and Ura, 2002), Fuzzy control (Debitetto, 1995), Adaptive Sliding Mode control (Cristi et al., 1990), Proportional and Derivative (PD) control (Bjorn, 1994; Pestero, 2001), Sliding Modes Control (SMC) (Healey and Lienard, 1993; Rodrigues et al., 1996) and Linear Quadratic Gaussian (LQG) controller (Fossen, 1994b). In all these approaches, in the first stage of the controller design, decoupling can be applied to the movements of the AUV and each movement can be modeled and controlled by a different set of differential equations and different controllers.

In the current research effort the aim is to utilize a more accurate and realistic modeling approach for a torpedo like AUV, the REMUS AUV, and based on the derived decoupled model of the vehicle's motions to design a novel constrained finite time optimal control scheme for the diving and steering motions. The proposed control scheme has the advantage of taking under consideration in the design phase: a) the physical and mechanical constrains, b) the disturbances from the environmental noise, and c) the additive uncertainty on the system transfer function due to modeling errors and non–linearities. In spite of the complexity of the control design stage (off–line), the on–line controller implementation results in an exhaustive search in a multidimensional look–up table, depending on the number of the system's states and control inputs, that can be easily implemented in an on board micro–controller .

This article is structured as follows. In Section 2 the utilized modeling approach for the AUV is presented, while the design of the proposed CFTO–control scheme is presented in Section 3. The validity of the proposed scheme is provided in Section 4 by simulation results, resulting from the application of the proposed scheme to the diving and steering motions of the AUV. Finally the conclusions are drawn in Section 5.

2 AUV MODELING

The AUV under study is a torpedo like underwater vehicle and it is illustrated in Figure 1 with the relevant body–fixed and inertial coordinate systems. The modeling of the equations of motion will be derived by the utilization of a relative standard framework that has been established in (Gertler and Hagen, 1967) and revised in (Humphreys, 1976) and (Feldman, 1979) with the assumptions: 1) the vehicle is deeply submerged in a homogeneous and unbounded liquid, 2) the vehicle does not experience memory effects, 3) the simulator neglects the effects of the vehicle passing through its own wake, and 4) the vehicle does not experience underwater currents. In addition, the following assumptions for the vehicle's dynamics are also necessary: 1) the vehicle is a rigid body of constant mass, 2) the control fins do not stall regardless of angle of attack, and 3) the propulsion model treats the vehicle propeller as a source of constant thrust and torque. The vehicle equations of motion in Figure 1, consist of the kinematics, rigid-body and mechanic terms (Fossen, 1994a) and by combining the equations for the vehicle rigid–body dynamics with the equations for the forces and the moments of the vehicle, we can conclude in the following non–linear set of equations in a general framework of six degrees of freedom plane. These equations follow the SNAME convention (SNAME, 1950) for the assignment of the body–fixed vehicle coordinate system and are presented in the above for retaining clarity.

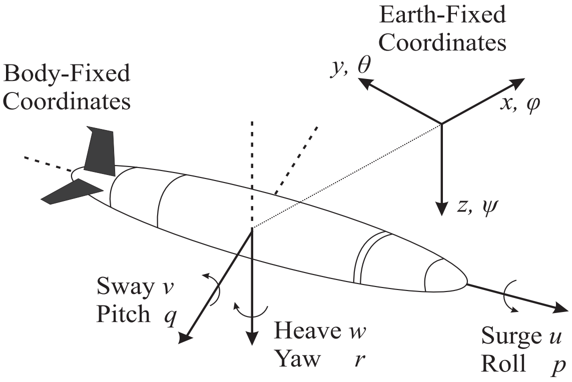


Figure 1: Body Fixed and Inertial Coordinate System of the AUV.

Eq.1 - Surge:

$$\begin{aligned}(m-X_{\dot{u}})\dot{u}+mz_g\dot{q}-my_g\dot{r}=X_{HS}+X_{u|u|}u|u|+(X_{\omega q}-m)\omega q\\+(X_{qq}+mx_g)q^2+(X_{\upsilon r}+m)\upsilon r+(X_{rr}+mx_g)r^2\\-my_g pq-mz_g pr+X\,prop\end{aligned}\tag{1}$$

Eq.2 - Sway:

$$\begin{aligned}(m-Y_{\dot{\upsilon}})\dot{\upsilon}+mz_g\dot{p}-(mx_g-Y_{\dot{r}})\dot{r}=Y_{HS}+Y_{\upsilon|\upsilon|}\upsilon|\upsilon|+Y_{r|r|}r|r|\\my_g r^2+(Y_{ur}-m)ur+(Y_{\omega p}+m)\omega p+(Y_{pq}-mx_g)pq\\+Y_{u\upsilon}u\upsilon+my_g p^2+mz_g qr+Y_{uu\delta_r}u^2\delta_r\end{aligned}\tag{2}$$

Eq.3 - Heave:

$$\begin{aligned}(m-Z_{\dot{\omega}})\dot{\omega}+my_g\dot{p}-(mx_g+Z_{\dot{q}})\dot{q}=Z_{HS}+Z_{\omega|\omega|}\omega|\omega|\\+Z_{q|q|}q|q|+(Z_{uq}+m)uq+(Z_{\upsilon p}-m)\upsilon p+(Z_{rp}-mx_g)rp\\+Z_{u\omega}u\omega+mz_g(p^2+q^2)-my_g rq+Z_{uu\delta_s}u^2\delta_s\end{aligned}\tag{3}$$

Eq.4 - Roll:

$$-mz_g\dot{\upsilon}+my_g\dot{\omega}+(I_{xx}-K_{\dot{p}})\dot{p}=K_{HS}+K_{p|p|}p|p|$$
$$-(I_{zz}-I_{yy})qr+m(uq-\upsilon p)-mz_g(\omega p-ur)+K_{prop} \tag{4}$$

Eq.5 - Pitch:

$$mz_g\dot{u}-(mx_g+M_{\dot{\omega}})\dot{\omega}+(I_{yy}-M_{\dot{q}})\dot{q}=M_{HS}+M_{\omega|\omega|}\omega|\omega|$$
$$+M_{q|q|}q|q|+(M_{uq}-mx_g)uq+(M_{\upsilon p}+mx_g)\upsilon p$$
$$+[M_{rp}-(I_{xx}-I_{zz})]rp+mz_g(\upsilon r-\omega q)+M_{u\omega}u\omega+M_{uu\delta_s}u^2\delta_s \tag{5}$$

Eq.6 - Yaw:

$$-my_g\dot{u}+(mx_g+N_{\dot{\upsilon}})\dot{\upsilon}+(I_{zz}-N_{\dot{r}})\dot{r}=N_{HS}+N_{\upsilon|\upsilon|}\upsilon|\upsilon|$$
$$+N_{r|r|}r|r|+(N_{ur}-mx_g)ur+(N_{\omega p}+mx_g)\omega p$$
$$+[N_{pq}-(I_{yy}-I_{xx})]pq-my_g(\upsilon r-\omega q)+N_{u\upsilon}u\upsilon+N_{uu\delta_r}u^2\delta_r \tag{6}$$

In these equations, the vehicle's cross products of inertia I_{xy}, I_{xz}, I_{yz} were assumed to be small and neglected. Moreover, zero value coefficients have not been included in the current formulation.

2.1 Diving Plane Motion

For deriving the equations of the diving plane motion, we should take under consideration only the body–relative surge velocity u, heave velocity ω, the pitch rate q, the earth–relative vehicle forward position x, the diving z, and the pitch angle θ. Before linearizing these equations in (1-6) we will integrate the terms for the hydrostatics, the axial and crossbow drag, the added mass, the body and fin lift and finally the moments. By assuming that the other velocities (υ, p, r) are negligible, we can result in the following linearized relationships between the body and earth fixed vehicle velocities (Prestero, 2001):

$$\begin{aligned}
(m-X_{\dot{u}})\dot{u}+mz_g\dot{q}-X_uu-X_qq-X_\theta\theta &= 0\\
(m-Z_{\dot{\omega}})\dot{\omega}-(mx_g+Z_{\dot{q}})\dot{q}-Z_\omega\omega-(mU+Z_q)q &= Z_{\delta_s}\delta s\\
mz_g\dot{u}-(mx_g+M_{\dot{\omega}})\dot{\omega}+(I_{yy}-M_{\dot{q}})\dot{q}-M_\omega\omega+&\\
(mx_gU-M_q)q-M_\theta\theta &= M_{\delta_s}\delta_s \qquad (7)
\end{aligned}$$

where at this point and for clarity in our presentation, the nomenclature that is ruling equations (1-7) is presented in Table 1.

If we assume that z_g is small compared to the other terms, we can decouple heave and pitch from surge, which results in the following set of equations:

$$\begin{aligned}
(m-Z_{\dot{\omega}})\dot{\omega}-(mx_g+Z_{\dot{q}})\dot{q}-Z_\omega\omega-(mU+Z_q)q &= Z_{\delta_s}\delta s \qquad (8)\\
-(mx_g+M_{\dot{\omega}})\dot{\omega}+(I_{yy}-M_{\dot{q}})\dot{q}-M_\omega\omega+&\\
(mx_gU-M_q)q-M_\theta\theta &= M_{\delta_s}\delta_s \qquad (9)
\end{aligned}$$

and based on the mentioned assumptions the vehicle's kinematic equations of motion are formulated as (Triantafyllou and Franz, 2003):

Table 1: Utilized parameters and their values in the linearized description of AUV's diving motion.

Par.	Name	Par.	Name
x_g	Center of gravity	z_g	Center of gravity
M_θ	Hydrostatic	m	AUV's mass
X_u	Axial drag	$X_{\dot{u}}$	Added mass
X_q	Added mass	I_{yy}	Moment of inertia
$Z_{\dot{q}}$	Heave velocity	U	Steady velocity
Z_ω	Combined term	$Z_{\dot{\omega}}$	Added mass
Z_q	Combined term	$Z_{\dot{q}}$	Added mass
M_ω	Combined term	$M_{\dot{\omega}}$	Added mass
M_q	Combined term	$M_{\dot{q}}$	Added mass
X_θ	Hydrostatic		

$$\begin{aligned}
\dot{x} &= \cos(\theta)u+\sin(\theta)\omega \qquad (10)\\
\dot{z} &= -\sin(\theta)u+\cos(\theta)\omega \qquad (11)\\
\dot{\theta} &= q \qquad (12)
\end{aligned}$$

For the linearization of equations in (10-12) it is assumed that the vehicle motion consists of small perturbations around a steady point. In this case, U represents the steady–state forward velocity of the vehicle. If heave and pitch are linearized about zero, we have $u=U+u^{'}$, $\omega=\omega^{'}$ and $q=q^{'}$. By utilizing: a) the equations in (10–12), b) the Maclaurin expansion of the trigonometric terms, c) dropping the higher order terms, and d) with the above assumption for the decoupling of heave and pitch from surge, the following linearized kinematic equations of motion are derived:

$$\begin{aligned}
\dot{z} &= \omega-U\theta \qquad (13)\\
\dot{\theta} &= q \qquad (14)
\end{aligned}$$

The combination of equations (8) and (9) with those in equations (13) and (14) results in the following matrix form for the description of the diving plane motion of the AUV:

$$\begin{bmatrix} m-X_{\dot{u}} & -(mx_g+Z_{\dot{q}}) & 0 & 0\\ -(mx_g+M_{\dot{\omega}}) & I_{yy}-M_{\dot{q}} & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix}\dot{\omega}\\ \dot{q}\\ \dot{z}\\ \dot{\theta}\end{bmatrix}-$$
$$\begin{bmatrix} Z_\omega & mU+Z_q & 0 & 0\\ M_\omega & -mx_gU+M_q & 0 & M_\theta\\ 1 & 0 & 0 & -U\\ 0 & 1 & 0 & 0 \end{bmatrix}\begin{bmatrix}\omega\\ q\\ z\\ \theta\end{bmatrix}=\begin{bmatrix}Z_{\delta_s}\\ M_{\delta_s}\\ 0\\ 0\end{bmatrix}[\delta_s] \tag{15}$$

Assuming that the heave velocities are small compared to the other terms, and that the center of gravity is equal to the buoyancy center ($x_g=0$), the equations in (15) are simplified to the following:

$$\begin{bmatrix} I_{yy}-M_{\dot{q}} & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix}\dot{q}\\ \dot{z}\\ \dot{\theta}\end{bmatrix}+\begin{bmatrix} -M_q & 0 & -M_\theta\\ 0 & 0 & U\\ -1 & 0 & 0 \end{bmatrix}\begin{bmatrix}q\\ z\\ \theta\end{bmatrix}$$
$$=\begin{bmatrix}M_{\delta_s}\\ 0\\ 0\end{bmatrix}[\delta_s] \tag{16}$$

and finally for the state space description of the linearized system, it is derived that:

$$\begin{bmatrix} \dot{q} \\ \dot{z} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{M_q}{I_{yy}-M_{\dot{q}}} & 0 & \frac{M_\theta}{I_{yy}-M_{\dot{q}}} \\ 0 & 0 & -U \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} q \\ z \\ \theta \end{bmatrix} + \begin{bmatrix} \frac{M_{\delta_S}}{I_{yy}-M_{\dot{q}}} \\ 0 \\ 0 \end{bmatrix} [\delta_s] \quad (17)$$

Given the state vector $x_1 = [q \;\; z \;\; \theta]' \in \Re^3$ and the input $u_{con_1} = \delta_s \in \Re$ we can write the matrix form in (17) as:

$$\begin{aligned} \dot{x_1} &= A_{div}x_1 + B_{div}u_{con_1} && (18) \\ y_1 &= C_{div}x_1 && (19) \end{aligned}$$

with $A_{div} = \begin{bmatrix} \frac{M_q}{I_{yy}-M_{\dot{q}}} & 0 & \frac{M_\theta}{I_{yy}-M_{\dot{q}}} \\ 0 & 0 & -U \\ 1 & 0 & 0 \end{bmatrix}$, $B_{div} = \begin{bmatrix} \frac{M_{\delta_S}}{I_{yy}-M_{\dot{q}}} \\ 0 \\ 0 \end{bmatrix}$ and $C_{div} = [1 \; 0 \; 0]$.

2.2 Steering Plane Motion

As it was presented in Section 2.2, the diving subsystem controls depth and pitch errors while the steering subsystem controls heading errors. In the presented approach it is assumed that the upper–bow rudder and the lower–bow rudder, as also the upper–stern and the lower–stern rudder of the AUV, are of identical size and shape and are receiving the deflection command equally, at the same time instant, but in an opposite direction. Moreover the following assumptions are made: 1) the center of mass of the vehicle lies below the origin (z_G is positive), 2) x_G and y_G are zero, 3) the vehicle is symmetric in its inertial properties, 4) the motions in the vertical plane are negligible ($[w_r, p, q, r, Z, \phi, \theta] = 0$), and 5) u_r equals the forward speed, U. Based on these assumptions and on the calculation of the hydrodynamic coefficients in (Fodrea, 2002), equations (1–6) are simplified to the following linearized equations:

$$\begin{aligned} m\dot{\upsilon} &= -mUr + Y_{\dot{\upsilon}}\dot{\upsilon} + Y_\upsilon \upsilon + Y_{\dot{r}}\dot{r} + Y_r r + Y_{\delta_s}\delta_r && (20) \\ I_{zz}\dot{r} &= N_{\dot{\upsilon}}\dot{\upsilon} + N_\upsilon \upsilon + N_{\dot{r}}\dot{r} + N_r r + N_{\delta_s}\delta_r && (21) \\ \dot{\psi} &= r && (22) \end{aligned}$$

where the nomenclature that is ruling the above set of equations is presented in Table 2.

Equations (20–21) in a matrix form could be written as:

$$\begin{bmatrix} m - Y_{\dot{\upsilon}} & -Y_{\dot{r}} & 0 \\ -N_{\dot{\upsilon}} & I_{zz} - N_{\dot{r}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\upsilon} \\ \dot{r} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} Y_\upsilon & Y_r - mU & 0 \\ N_\upsilon & N_r & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \upsilon \\ r \\ \psi \end{bmatrix} + \begin{bmatrix} Y_{\delta_S} \\ N_{\delta_S} \\ 0 \end{bmatrix} [\delta_r] \quad (23)$$

For the steering subsystem there are three state variables: r, ψ, ad υ. The r variable is the yaw rate of turn, ψ variable represents the heading angle, while υ represents the sway velocity. Based on the assumption made for the vehicle dynamics is that the cross coupling terms in the mass matrix are zero due to the assumed symmetry in the rudders, equation (23) can be given as:

Table 2: Utilized parameters and their values in the linearized description of AUV's Steering motion.

Par.	Name
$Y_{\dot{\upsilon}}$	Added mass is sway
$Y_{\dot{r}}$	Added mass in yaw
Y_υ	Sway force induced by side slip
Y_r	Sway force induced by yaw
$N_{\dot{\upsilon}}$	Added mass in sway
$N_{\dot{r}}$	Added mass in yaw
N_υ	Sway moment from side slip
N_r	Sway moment from yaw
Y_{δ_S}	Linearized rudder action force
N_{δ_S}	Linearized rudder action force

$$\begin{bmatrix} m - Y_{\dot{\upsilon}} & 0 & 0 \\ 0 & I_{zz} - N_{\dot{r}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\upsilon} \\ \dot{r} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} Y_\upsilon & Y_r - mU & 0 \\ N_\upsilon & N_r & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \upsilon \\ r \\ \psi \end{bmatrix} + \begin{bmatrix} Y_{\delta_S} \\ N_{\delta_S} \\ 0 \end{bmatrix} [\delta_r] \quad (24)$$

where δ_r is the control signal applied to both rudders. The state space description of the linearized system will be:

$$\begin{bmatrix} \dot{\upsilon} \\ \dot{r} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \frac{Y_\upsilon}{m-Y_{\dot{\upsilon}}} & \frac{Y_r - mU}{m-Y_{\dot{\upsilon}}} & 0 \\ \frac{N_\upsilon}{I_{ZZ}-N_{\dot{r}}} & \frac{N_r}{I_{ZZ}-N_{\dot{r}}} & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \upsilon \\ r \\ \psi \end{bmatrix} + \begin{bmatrix} \frac{Y_{\delta_S}}{m-Y_{\dot{\upsilon}}} \\ \frac{N_{\delta_S}}{I_{ZZ}-N_{\dot{r}}} \\ 0 \end{bmatrix} [\delta_r] \quad (25)$$

Given the state vector $x_2 = [\upsilon \;\; r \;\; \psi]' \in \Re^3$ and the input $u_{con_2} = \delta_r \in \Re$, the matrix form in Eq. (25) can be written as:

$$\begin{aligned} \dot{x_2} &= A_{steer}x_2 + B_{steer}u_{con_2} && (26) \\ y_2 &= C_{steer}x_2 && (27) \end{aligned}$$

where

$A_{steer} = \begin{bmatrix} \frac{Y_\upsilon}{m-Y_{\dot{\upsilon}}} & \frac{Y_r - mU}{m-Y_{\dot{\upsilon}}} & 0 \\ \frac{N_\upsilon}{I_{ZZ}-N_{\dot{r}}} & \frac{N_r}{I_{ZZ}-N_{\dot{r}}} & 0 \\ 0 & 1 & 0 \end{bmatrix}$, $B_{steer} = \begin{bmatrix} \frac{Y_{\delta_S}}{m-Y_{\dot{\upsilon}}} \\ \frac{N_{\delta_S}}{I_{ZZ}-N_{\dot{r}}} \\ 0 \end{bmatrix}$ and $C_{steer} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$.

3 CONSTRAINED FINITE TIME OPTIMAL CONTROLLER SYNTHESIS

In the proposed control strategy, the aim is to design a CFTOC–scheme for the decoupled diving and steering motion of the AUV as it is presented in Figure 2. At this point it should be mentioned that the speed

$u \in \Re$ of the AUV was considered to be constant and no control action has been considered for this motion while the control actions are applied to the linear model of the AUV after the ZOH. Prior to the design of the control algorithm, it is necessary to model and take under consideration in the controller's synthesis, the mechanical constrains of the utilized AUV, the disturbances that are introduced from the onboard sensors, the additive uncertainties due to modeling errors and the non–linearities.

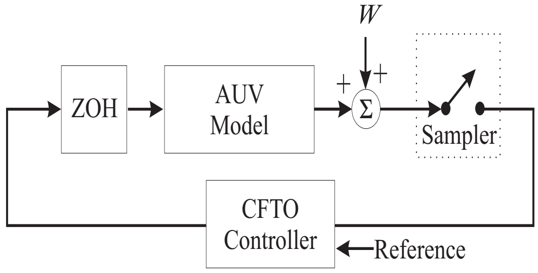


Figure 2: The Proposed Control Scheme for the AUV's Depth and Steering Motions

The derived linearized decoupled models in equations (18-19) and (26-27) are valid only for small values of pitch, yaw and fin angles around the linearization points. Moreover, the control actions u_{con_1}, u_{con_2} should also be bounded due to physical constraints applied to the motors. We consider the state vector $x = [x_1,\ x_2]^T \in \Re^6$ and the control vector $u = [u_{con_1},\ u_{con_1}]^T \in \Re^2$.

Let the matrix H_i be a zeroed 2×8 matrix except for its i–th column, which is equal to $[1, -1]^T$ and with $i \in [1, 2, \cdots, 16]$, i.e.:

$$H_i = \begin{bmatrix} 0 & \dots & 0 & 1 & 0 & \dots & 0 \\ 0 & \dots & 0 & -1 & 0 & \dots & 0 \end{bmatrix} \tag{28}$$

The bounds can be cast in a more compact form as:

$$\begin{bmatrix} H_1 \\ \vdots \\ H_{12} \\ -- \\ H_{13} \\ \vdots \\ H_{16} \end{bmatrix}_{16\times 8} \cdot \begin{bmatrix} x \\ -- \\ u \end{bmatrix}_{8\times 1} \leq \begin{bmatrix} x_1^{\max} \\ x_1^{\min} \\ \vdots \\ x_6^{\max} \\ x_6^{\min} \\ -- \\ u_1^{\max} \\ u_1^{\min} \\ u_2^{\max} \\ u_2^{\min} \end{bmatrix}_{16\times 1} \tag{29}$$

where the notation x_i corresponds to the ith element of the vector x.

Due to factors such as noise, accuracy of measurements and round–off errors, the onboard measurements are not ideal. These inaccuracies in the measurements, in the presented approach are considered as additive disturbances to the system models. Moreover we will consider uncertainty in the state space matrices, due to the existence of modelling simplifications and errors. If we consider a sampling time $T_s \in \Re^+$, the combined discrete time version of equations (18-19) and (26-27) with the effects of the additive noise, can be cast as piecewise affine (PWA) system:

$$\dot{x} = \begin{bmatrix} A_{div}^{*,i} & 0_{3\times 3} \\ 0_{3\times 3} & A_{steer}^{*,i} \end{bmatrix} x + \begin{bmatrix} B_{div}^{*,i} & B_{steer}^{*,i} \end{bmatrix} u + W \tag{30}$$

with the constraints in (29). The notation $(\cdot)^*$ represents the discrete time value of the corresponding matrix, while $W \in \Re^8$ is an additive and of a zero mean white noise, bounded by the set $\mathcal{W} \subseteq \Re^6$. Moreover $j \in \mathcal{S}$, with $\mathcal{S} \triangleq \{1, 2, \cdots, s\}$, is a finite set of indexes and $s \in \mathbb{Z}^+$ denotes the number of affine sub–systems in (30). For polytopic uncertainty, Ω is the polytope defined as:

$$\Omega = Co\{[A_{div}^{*,1}\, A_{steer}^{*,1}\ B_{div}^{*,1}\ B_{steer}^{*,1}], \cdots, [A_{div}^{*,s}\ A_{steer}^{*,s}\ B_{div}^{*,s}\ B_{steer}^{*,s}]\}, \tag{31}$$

where Co denotes the convex hull and $[A_{div}^{*,i}\ \ A_{steer}^{*,i}\ \ B_{div}^{*,i}\ \ B_{steer}^{*,i}]$ are vertices of the convex hull. Any $[A_{div}^{*}\ \ A_{steer}^{*}\ \ B_{div}^{*}\ \ B_{steer}^{*}]$ within the convex set Ω is a linear combination of the vertices:

$$[A_{div}^{*}\ A_{steer}^{*}\ B_{div}^{*}\ B_{steer}^{*}] = \sum_{j=1}^{L} a_j [A_{div}^{*}\ A_{steer}^{*}\ B_{div}^{*}\ B_{steer}^{*}] \tag{32}$$

with $\sum_{j=1}^{L} a_j = 1,\ 0 \leq a_j \leq 1$.

The CFTOC-design problem consists of computing the optimum control vector sequence $\tilde{u} = [u_k\ u_{k+1}\ \dots\ u_{k+N-1}]^T$, where N corresponds to the prediction horizon that minimizes the following cost function:

$$\begin{aligned} J_N(x_k, x^r) \quad = \quad & [x_{k+N} - x^r)]^T \tilde{P} [x_{k+N} - x^r] + \\ & \sum_{m=0}^{N-1} [u_{k+m}]^T R [u_{k+m}] + [x_{k+m} - x^r]^T Q [x_{k+m} - x^r] \end{aligned}$$

where $\tilde{P} \in \Re^{6\times 6}$ with $\tilde{P} = \tilde{P}^T \geq 0$, $R \in \Re^{2\times 2}$ with $R = R^T > 0$ and $Q \in \Re^{6\times 6}$ with $Q = Q^T \geq 0$, are full column rank weighting matrices penalizing the corresponding optimization variables i.e predicted states, control effort and the desired final state, respectively, while x^r is the reference set–point.

The solution to the CFTOC problem (Borelli et al., 2003; Grieder et al., 2004; Kvasnica et al., 2004) is a continuous control action of the form:

$$u_k \quad = \quad F_l x_k + G_l \text{ if } x_k \in \tilde{R}_l \tag{33}$$

where $\tilde{R}_l$, $l \in \{1, \dots l^{\max}\}$ corresponds to a convex polyhedron $(\tilde{R}_l \in \Re^6)$ calculated by the algorithm, and $F_l \in \Re^{2\times6}$, $G_l \in \Re^{2,1}$. The $l^{\max}$–number of polyhedra is similarly specified by the algorithm. In general the higher the number of the PWA–systems along with the large dimension of the state vector and the number of constraints, the more complicated the solution is.

This complexity increases significantly with the value of the prediction horizon N, and the number $l^{\max}$ of the convex polyhedra (regions) grows up (usually) exponentially. Certain techniques have been devised to merge the various regions into larger ones without significantly compromising the validity of the solution. For the on–line implementation of the controller the number $l^{\max}$ of the regions is not only a measure of the controller's complexity but also affects its implementation typically precomputed in a look–up table.

4 SIMULATION STUDIES

The proposed CFTO–control scheme has been applied in simulation studies on the model of the REMUS AUV. Based on experimental results in (Prestero, 2000) the parameters presented in Tables I and II were tuned and the continuous time state space matrices for the diving and steering motion of the AUV are:

$$A_{div} = \begin{bmatrix} -0.82 & 0 & -0.69 \\ 0 & 0 & -1.54 \\ 1 & 0 & 0 \end{bmatrix},\ B_{div} = \begin{bmatrix} -4.16 \\ 0 \\ 0 \end{bmatrix} \tag{34}$$

$$A_{steer} = \begin{bmatrix} -1.01 & -0.68 & 0 \\ -0.54 & -0.82 & 0 \\ 0 & 1 & 0 \end{bmatrix},\ B_{steer} = \begin{bmatrix} 0.22 \\ -1.19 \\ 0 \end{bmatrix} \tag{35}$$

The constraints on the control inputs and the outputs have been arbitrary set to: $u_{con_{1_{min}}} = -10 \le u_{con_1}(t) \le 10 = u_{con_{1_{max}}}$, $u_{con_{2_{min}}} = -60 \le u_{con_2}(t) \le 60 = u_{con_{2_{max}}}$ and $y_{1_{min}} = -50^0 \le y_1(t) \le 50^0 = y_{1_{max}}$, $y_{2_{min}} = -70^0 \le y_2(t) \le 50^0 = y_{1_{max}}$ respectively. The constrains for the states have been set as:

$$\begin{array}{rcl} -30^0 \le & x_1 & \le 30^0 \\ -20^0 \le & x_2 & \le 20^0 \\ -360 \le & x_3 & \le -360 \\ -50^0 \le & x_4 & \le 50^0 \\ -40^0 \le & x_5 & \le 40^0 \\ -360 \le & x_3 & \le -360 \end{array}$$

For the state space matrices A_{div}, B_{div}, A_{steer}, B_{steer}, we assume that there is an additive corrupting uncertainty of 1%, while the additive disturbances have been set to $0.01 \cdot I_{6\times1}$. The selection that has been made on the penalizing matrices for the CFTOC cost was $P = 10^3 \cdot I^{6\times6}$, $R = I^{2\times2}$ and $Q = 10^3 \cdot I^{6\times6}$. The output set–point was selected for the diving motion as $Y_{1_{ref}} = 5^0$ and $Y_{2_{ref}} = 5^0$ for the steering motion. The initial augmented state vector was $X_{init} = O_{6\times1}$. For the controllers formulation the $2-$ vector norm case was tested and the discretization has been made with a sampling period of $T_s = 1sec$.

The resulting controllers' partitions for the diving and steering motion are presented in Figures 3 and 4, while the responses of the diving and steering motions are displayed in Figures 5 and 6 respectively, while the controllers' response for the cases of the diving and steering motion are presented in Figures 7 and 8 respectively. Finally in Figure 9 the 3–D combined movement of the AUV (including displacement with a constant speed (U=1.54m/sec) is displayed.

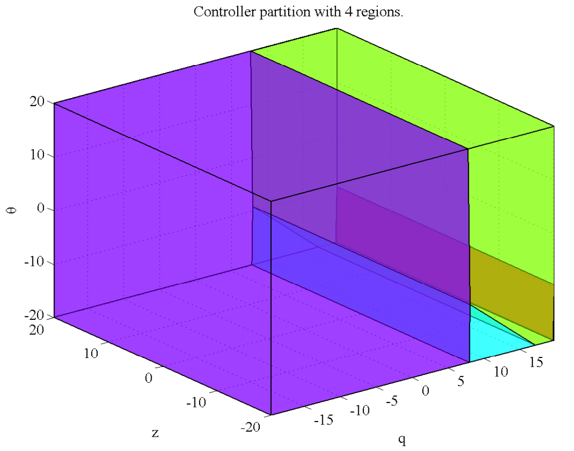


Figure 3: CFTO–Controller Partitioning for the Diving Motion.

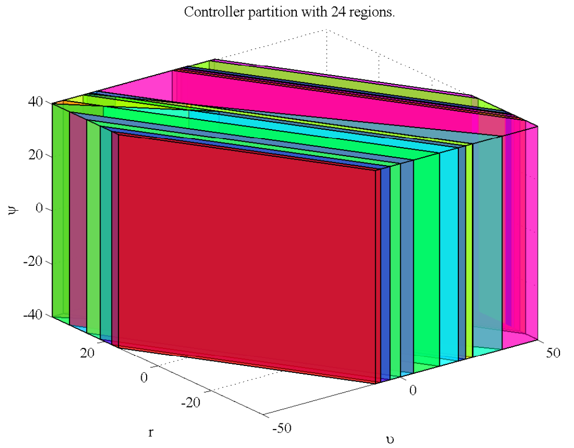


Figure 4: CFTO–Controller Partitioning for the Steering Motion.

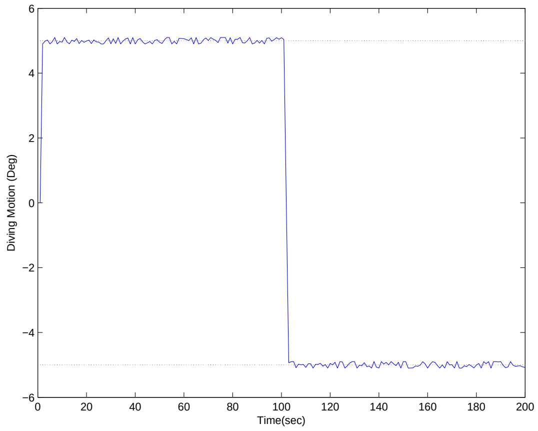


Figure 5: Diving Motion Time Response.

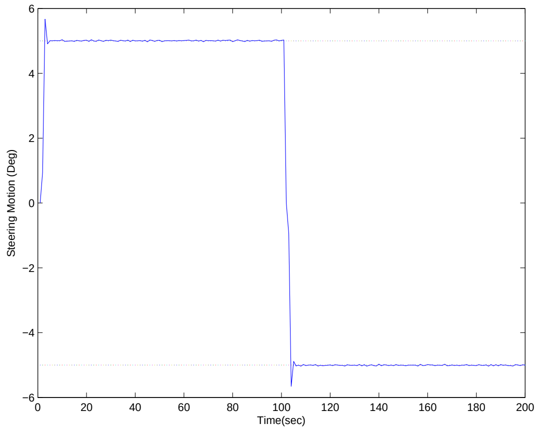


Figure 6: Steering Motion Time Response.

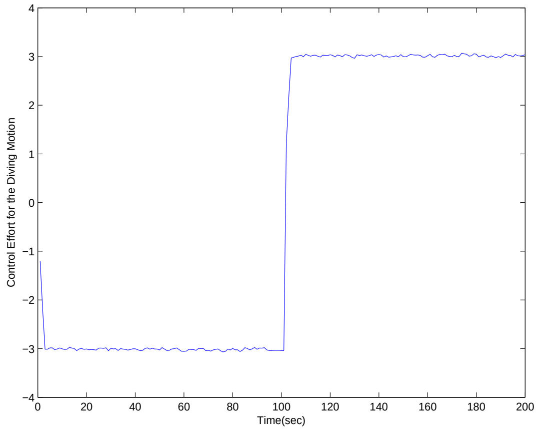


Figure 7: Controller Effort for the Diving Motion.

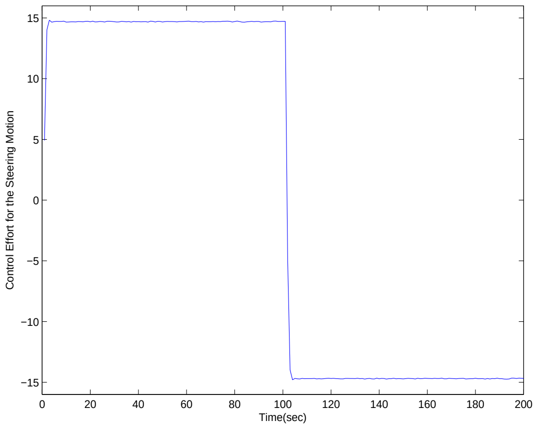


Figure 8: Controller Effort for the Steering Motion.

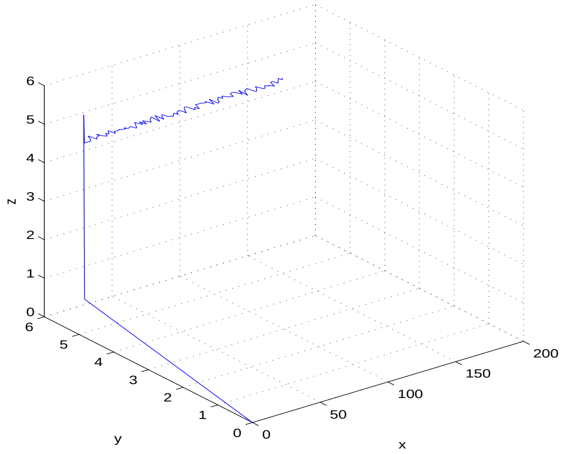


Figure 9: 3–D Combined movement of the AUV.

5 CONCLUSIONS

In this paper a constrained finite time optimal controller for the diving and steering motion of an AUV has been presented. The utilized proposed scheme was developed based on the decoupled equations of motion for the diving and steering of the AUV. The derived Constrained Finite Time Optimal control had the ability to take under consideration: a) the constrains on the inputs, outputs, and the states of the model, b) the corrupting disturbances due to the existence of the noise in the measurements, and c) the uncertainty in the modeling procedures. Simulation results have been presented that prove the validity of the proposed scheme.

REFERENCES

Bjorn, J. (1994). The NDRE-AUV Flight Control System. *IEEE journal of Oceanic Engineering*, 19(4).

Borelli, F., Baotic, M., Bemporad, A., and Morari, M. (2003). An Efficient Algorithm for Computing the State Feedback Optimal Control Law for Discrete Time Hybrid Systems.

Cristi, R., Papoulias, A., and Healey, A. J. (1990). Adaptive Sliding Mode Control of Autonomous Undewater Vehicle in the Dive Plane. *IEEE Journal of Oceanic Engineering*, 15(3).

Debitetto, A. (1995). Fuzzy logic for depth control of unmanned undersea vehicles. *IEEE journal of Oceanic Engineering*, 20(3).

E. An, R. Dhanak, L. S. S. S. J. L. (2001). Coastal oceanography using a small auv. *Journal of Atmospheric and Ocean Technology*, (18):215–234.

Feldman, J. (1979). Revised standard submarine equations of motion report dtnsrdc/spd-0393-09. Technical report, David W. Taylor Naval Ship Research and Development Center, Bethesda, MD.

Field, A. I. (2000). Optimal control of an autonomous underwater vehicle. In *World Automatic Congress*, Maui, Hawaii.

Fodrea, L. R. (2002). *Obstactle Avoidance Control for the REMUS Autonomous Underwater Vehicle*. PhD thesis, Naval Postgraduate School, Monterey, California.

Foresti, G. (2001). Visual inspection of sea bottom structures by an autonomous underwater vehicle. *IEEE Transactions on Systems, Man and Cybernetics - Part B: cybernetics*, 31(5).

Fossen, T. (1994a). *Guidance and control of ocean vehicles*. John Wiley & Sons Ltd.

Fossen, T. I. (1994b). *Guidance and Control of an Autonomous Underwater Vehicle*. John Wiley and Sons.

Gertler, M. and Hagen, G. (1967). Standard equations of motion for submarine simulation report dtnsrdc 2510. Technical report, David W. Taylor Naval Ship Research and Development Center, Bethesda.

Grieder, P., Borelli, F., Torrisi, F., and Morari, M. (2004). Computation of the Constrained Infinite Time Linear Quadratic Regulator. *Automatica*, 40(4):701–708.

Healey, A. and Lienard, D. (1993). Multivariable sliding mode control for autonomous diving and steering of unmanned underwater vehicles. In *IEEE Journal of Oceanic Engineering*, volume 18.

Humphreys, D. (1976). Development of the equations of motion and transfer functions for underwater vehicles. Technical report, Naval Coastal Systems Laboratory, Panama City, FL.

Kawano, H. and Ura, T. (2002). Fast reinforcement learning algorithm for motion planning of non-holonomic unmanned underwater vehicles. In *Int. Conference on Intelligent Robots and Systems IEEE/RSJ*, Lausanne, Switzerland.

Kvasnica, M., Grieder, P., Baotic, M., and Morari, M. (2004). Multi–Parametric Toolbox (MPT). *Hybrid Systems: Computation and Control*, (2993):448–462.

Pestero, T. (2001). Verification of six–degree of freedom simulation model for the remus auv. Master's thesis, Massachusetts Institute of Technology.

Prestero, T. (2000). Development of a six–degree of freedom simulation model for the remus autonomous underwater vahicle. In *In Proceedings of MTS-IEEE Oceans 2000,Providence, Rhode Island*.

Prestero, T. (2001). Verification of a six-degree of freedom simulation model for the remus autonomous underwater vehicle. Master's thesis, Massachusetts Institute of Technology and Woods Hole Oceanographic Institution, Cambridge, Massachusetts.

Rodrigues, L., Tavares, P., and Prado, M. (1996). Sliding mode control of an auv in the diving and steering planes. In *OCEANS MTS/IEEE*.

SNAME (1950). Nomenclature for treating the motion of a submerged body through a fluid. *The Society of Naval Architects and Marine Engineers, Technical and Reserach Bulletin*, (1-5):1–15.

Triantafyllou, M. and Franz, S. (2003). Hover maneuvering and control of marine vehicles. *Lectures*.

Yuh, J. (2000). Design and control of autonomous underwater robots: A survey. *Autonomous Robots*, 8(1):7–24.

A NEW PREDICTOR/CORRECTOR PAIR TO ESTIMATE THE VISUAL FEATURES DEPTH DURING A VISION-BASED NAVIGATION TASK IN AN UNKNOWN ENVIRONMENT
A Solution for Improving the Visual Features Reconstruction During an Occlusion

A. Durand Petiteville, M. Courdesses and V. Cadenat
CNRS, LAAS, 7 avenue du colonel Roche, F-31077 Toulouse, France
Université de Toulouse, UPS, INSA, INP, ISAE, LAAS, F-31077 Toulouse, France
{adurandp, courdess, cadenat}@laas.fr

Keywords: Visual servoing, Visual data estimation, Predictor/estimator pair.

Abstract: This papers deals with the problem of estimating the visual features during a vision-based navigation task when a temporary total occlusion occurs. The proposed approach relies on an existent specific algorithm. However, to be efficient, this algorithm requires highly precise initial values for both the image features and their depth. Thus, our objective is to design a predictor/estimator pair able to provide an accurate estimation of the depth value, even when the visual data are noisy. The obtained results show the efficiency and the interest of our technique.

1 INTRODUCTION

In the past decades, many works have addressed the problem of using information provided by a vision system to control a robot. Such techniques are commonly known as Visual Servoing (Corke, 1996), (Chaumette and Hutchinson, 2006). Visual servoing is roughly classified into two main categories: Image based visual servoing (IBVS) and Position based visual servoing (PBVS) (Chaumette and Hutchinson, 2006). In the first approach, the goal to be reached is expressed only in the image space, whereas in the second one, it is given in terms of a desired camera pose (Corke, 1996). A complete survey can be found in (Chaumette and Hutchinson, 2006).

We focus in the sequel on the first kind of control. In this case, the control law depends only on the visual features. Therefore, if they are lost because of an occlusion or any other unexpected event, the desired task cannot be realized anymore. Here, we are working in a mobile robotics context. In this context, the realization of a vision-based navigation task in a given environment requires to preserve not only the image data visibility, but also the robot safety. In that case, techniques allowing to avoid simultaneously collisions and visual data losses appear to be limited, because they are restricted to missions where an avoidance motion exists without leading to local minima (Folio and Cadenat, 2008). As many robotic tasks cannot be performed if the visual data loss is not tolerated, it is necessary to provide methods which accept that occlusions may effectively occur without leading to a task failure.

Folio has proposed such an approach (Folio and Cadenat, 2008). The idea is to reconstruct the visual features whenever necessary. The developed algorithm is based upon the vision/motion link which relates the variation of the visual features in the image to the camera motion. However, this method needs accurate initial values for the visual features and the depth. If the first ones can be obtained from the last image available before the occlusion occurs, determining a precise initial value for the depth sufficiently rapidly to correctly handle the occlusion remains a challenging problem. Different approaches are proposed in the literature. See for instance the works by Matthies who derived and compared several algorithms based on a Kalman filter (Matthies et al., 1989). However, a correct depth value can be obtained only if the camera motion respects some very particular constraints. These solutions are not suitable for our particular case. It would be also possible to use the epipolar geometry (Ma et al., 2003), stereovision (Cervera et al., 2002), or even structure from motion techniques (Jerian and Jain, 1991). However, this kind of approaches are time-consuming and cannot be used for our purpose. Finally, (De Luca et al., 2008) have proposed to estimate the depth using a non linear observer. But, unfortunately, the convergence time seems to be too large to be useful in our specific context.

In this paper, we have developed a non recursive algorithm based on a predictor/corrector pair allowing to provide an accurate value of the depth using visual data. This value will then be used to feed D. Folio's algorithm in order to improve its efficiency.

The paper is organized as follows. Section II is dedicated to some preliminaries regarding the system modelling, the visual servoing control law design and the description of D. Folio's algorithm. Section III details and analyzes the predictor/estimator pair which has been developed. Finally, simulation results validating our approach are shown in section IV.

2 PRELIMINARIES

2.1 System Modelling

We consider the system presented in figure 1(a), which consists of a robot equipped with a camera mounted on a pan-platform. We describe the successive frames : $F_O(O, \vec{x}_O, \vec{y}_O, \vec{z}_O)$ attached to the world, $F_M(M, \vec{x}_M, \vec{y}_M, \vec{z}_M)$ linked to the robot, $F_P(P, \vec{x}_P, \vec{y}_P, \vec{z}_P)$ attached to the platform, and $F_C(C, \vec{x}_C, \vec{y}_C, \vec{z}_C)$ linked to the camera. Let θ be the direction of the robot wrt. $\vec{x}_O$, ϑ the direction of the pan-platform wrt. $\vec{x}_M$, P the pan-platform centre of rotation and D_x the distance between the robot reference point M and P. Defining vector $q = (l, \theta, \vartheta)^T$ where l is the robot curvilinear abscissa, the control input is given by $\dot{q} = (\upsilon, \omega, \varpi)^T$, where υ and ω are the cart linear and angular velocities, and ϖ is the pan-platform angular velocity wrt. F_M. For such a robot, the kinematic model is classically given by the following relations:

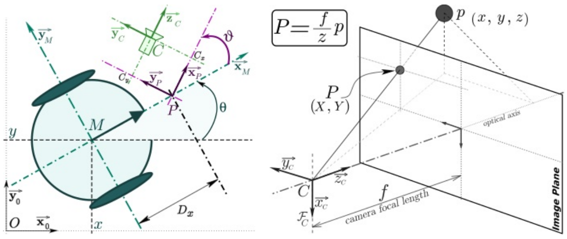
$$\begin{cases} \dot{M}_x(t) = \upsilon(t)\cos(\theta(t)) \\ \dot{M}_y(t) = \upsilon(t)\sin(\theta(t)) \\ \dot{\theta}(t) = \omega(t) \\ \dot{\vartheta}(t) = \varpi(t) \end{cases} \tag{1}$$

where $\dot{M}_x(t)$ is the speed of M wrt. $\vec{x}_O$ and $\dot{M}_y(t)$ wrt. $\vec{y}_O$.

The camera motion can be described by the kinematic screw T_{C/F_O}:

$$T_{C/F_O} = \left[(V_{C/F_O})^T (\Omega_{F_C/F_O})^T)\right]^T \tag{2}$$

where V_{C/F_O} and Ω_{F_C/F_O} are the camera translation and rotation speeds wrt. the frame F_O. For this specific mechanical system, T_{C/F_O} is related to the control input by the robot jacobian J : $T_{C/F_O} = J\dot{q}$. As the camera is constrained to move horizontally, it is sufficient to consider a reduced kinematic screw $T_r = (V_{\vec{y}_C}, V_{\vec{z}_C}, \Omega_{\vec{x}_C})^T$, and a reduced jacobian matrix J_r as follows:



(a) The robotic system. (b) The camera pinhole model.

Figure 1: System modeling.

$$T_r = J_r \dot{q} \tag{3}$$

Defining C_x and C_y as the coordinates of C along axes $\vec{x}_P$ and $\vec{y}_P$ (see figure 1(a)), J_r is given by:

$$J_r = \begin{pmatrix} -\sin(\vartheta(t)) & D_x\cos(\vartheta(t)) + C_x & C_x \\ \cos(\vartheta(t)) & D_x\sin(\vartheta(t)) - C_y & -C_y \\ 0 & -1 & -1 \end{pmatrix} \tag{4}$$

It should be noted that J_r can be inverted ($det(J_r) = D_x$).

2.2 The Vision-based Navigation Task

The vision-based navigation task consists in positioning the camera with respect to a given static landmark. We assume that this landmark can be characterized by n interest points which are extracted by our image processing. Therefore, the visual data are represented by a $2n$-dimensional vector s made of the coordinates (X_i, Y_i) of each point P_i, in the image plane as shown on figure 1(b). For a fixed landmark, the variation of the visual signal $\dot{s}$ is related to the reduced camera kinematic screw T_r thanks to the interaction matrix $L_{(s,z)}$ as shown below (Espiau et al., 1992):

$$\dot{s} = L_{(s,z)} T_r = L_{(s,z)} J_r \dot{q} \tag{5}$$

In the case of n points, $L_{(s,z)} = [L_{(P_1)}^T, \ldots, L_{(P_n)}^T]^T$ where $L_{(P_i)}$ is classically given by (Espiau et al., 1992):

$$L_{(P_i)} = \begin{pmatrix} L_x(s_i, z_i) \\ L_y(s_i, z_i) \end{pmatrix} = \begin{pmatrix} 0 & \frac{X_i}{z_i} & \frac{X_i Y_i}{f} \\ -\frac{f}{z_i} & \frac{Y_i}{z_i} & f + \frac{Y_i^2}{f} \end{pmatrix} \tag{6}$$

where z_i represents the depth of the projected point p_i, and f is the camera focal (see figure 1(b)). At the beginning of the visual servoing, z_i is not available yet. Thus, only an approximation of the interaction matrix noted $\hat{L}_{(s^*,z^*)}$ computed at the desired position will be used to build the control law as in (Chaumette and Hutchinson, 2006).

To perform the desired vision-based task, we apply the visual servoing technique given in (Espiau et al., 1992) to mobile robots as in (Pissard-Gibollet and Rives, 1995). The proposed approach relies on the task function formalism (Samson et al., 1991) and consists in expressing the visual servoing task by the following task function to be regulated to zero:

$$e = C(s - s^*) \tag{7}$$

where s^* represents the desired value of the image data. Matrix C, called combination matrix, allows to take into account more visual features than available degrees of freedom. Several choices are possible: see for example (Comport et al., 2004). A classical idea consists in defining $C = \hat{L}^+_{(s^*,z^*)} = (\hat{L}^T_{(s^*,z^*)}\hat{L}_{(s^*,z^*)})^{-1}\hat{L}^T_{(s^*,z^*)}$ because the expression of the control law is simplified (see (8) hereafter). Now, it remains to determine a controller allowing to make e vanish. Such a controller is classically designed by imposing an exponential decrease, that is $\dot{e} = -\lambda e$, where λ is a positive scalar or a positive definite matrix. Then, the visual servoing controller can be written as follows:

$$\dot{q}_{(s)} = -(C\hat{L}_{(s^*,z^*)}J_r)^{-1}\lambda C(s - s^*) = J_r^{-1}\lambda C(s - s^*) \tag{8}$$

2.3 Visual Data Estimation

However, the above controller can only be used if the visual data are available. If they are not because of a landmark occlusion or a camera failure for example, the task cannot be realized anymore. To remedy this critical situation, (Folio and Cadenat, 2008) has recently proposed to solve the dynamic system (5) to obtain the expression of the visual data. However, the latter depends not only on s but also on depth z which must then be determined. As our robot is not equipped with any sensor able to measure this data, we have to reconstruct it. After some computations (see (Folio and Cadenat, 2008) for a detailed proof), it can be shown that, for any $t \in [t_{k-1}, t_k]$, X_i, Y_i and z_i express as:

$$\begin{cases} X_i(t) = \frac{z_i(k-1)X_i(k-1)}{z_i(t)} \\ Y_i(t) = \frac{f}{z_i(t)}\Big\{D_x\sin(\vartheta(t)) + \frac{\upsilon(k-1)}{\omega(k-1)}\cos(\vartheta(t)) - C_y \\ \qquad +c_1\cos(A(\dot{q}(k-1))(t-t_{k-1})) \\ \qquad -c_2\sin(A(\dot{q}(k-1))(t-t_{k-1})\Big\} \\ z_i(t) = -D_x\cos(\vartheta(t)) + \frac{\upsilon(k-1)}{\omega(k-1)}\sin(\vartheta(t)) - C_x \\ \qquad +c_1\sin(A(\dot{q}(k-1))(t-t_{k-1})) \\ \qquad +c_2\cos(A(\dot{q}(k-1))(t-t_{k-1})) \end{cases} \tag{9}$$

where:

$$\begin{cases} A(\dot{q}(k-1)) = \omega(k-1) + \varpi(k-1) \\ c_1 = \frac{Y_i(k-1)z_i(k-1)}{f} - D_x\sin(\vartheta(k-1)) \\ \qquad -\frac{\upsilon(k-1)}{\omega(k-1)}\cos(\vartheta(k-1)) + C_y \\ c_2 = z_i(k-1) + D_x\cos(\vartheta(k-1)) \\ \qquad -\frac{\upsilon(k-1)}{\omega(k-1)}\sin(\vartheta(k-1)) + C_x \end{cases}$$

Thanks to (9), Folio has developped a recursive algorithm able to estimate X_i, Y_i and z_i provided that $\vartheta(t)$ has been previously determined[1]. However, it should be noted that initial conditions, namely $X_i(k-1)$, $Y_i(k-1)$ and $z_i(k-1)$, are required to determine $X_i(k)$, $Y_i(k)$ and $z_i(k)$. So, this algorithm cannot be used to properly estimate $z_i(k)$ without a precise initial value of $z_i(k-1)$.

3 METHODOLOGY

In this paper, we propose to estimate the depth by building a predictor/estimator pair using data from m images, with $m \in \mathbb{N}^*$ to repair a too small signal/noise ratio (Durand Petiteville et al., 2009). Our first objective is to express a predictor $\hat{X}_i(k|k-j), \hat{Y}_i(k|k-j)$ of one point P_i at instant k using the image at $k-j$, with $j \in [1, ..., m]$. To this aim, we rewrite equation (9) to relate $z_i(k-1|k-1)$ and $z_i(k|k-1)$. We obtain:

$$\hat{z}_i(k-1|k-1) = \frac{\hat{z}_i(k|k-1) - \beta}{\alpha_i} \tag{10}$$

where α_i and β are given in the appendix. Denoting by $\tilde{X}_i(k-1)$ and $\tilde{Y}_i(k-1)$ the visual data measured at instant $k-1$, we use (10) in (9) to obtain the following predictor for the visual features:

[1] As one can see, solution (9) requires the determination of $\vartheta(t)$. This angle can be simply computed by integrating $\dot{\vartheta} = \varpi$ between t_{k-1} and t. A straightforward calculus leads to $\vartheta(t) = \varpi(k-1)(t - t_{k-1}) + \vartheta(k-1)$, where $\vartheta(k-1)$ is the pan-platform angular value at t_{k-1}, which is usually provided by the embedded encoder.

$$\left\{\begin{array}{l} \hat{X}_i(k|k-1) = \frac{\hat{z}_i(k|k-1)\tilde{X}_i(k-1)-\beta\tilde{X}_i(k-1)}{\hat{z}_i(k|k-1)\alpha_i} \\ \hat{Y}_i(k|k-1) = \frac{f}{\hat{z}_i(k|k-1)} \\ \qquad (\frac{\tilde{Y}_i(k-1)\hat{z}_i(k|k-1)}{f\alpha_i}\cos(A(\dot{q}(k-1))T) \\ \qquad -\frac{\hat{z}_i(k|k-1)}{\alpha_i}\sin(A(\dot{q}(k-1))T)+\kappa_i) \end{array}\right. \tag{11}$$

where κ_i is given in the appendix $T = t_k - t_{k-1}$. As shown by (11), the obtained predictor depends only on the last image. To use more than one image and improve the accuracy, a first natural solution is to recursively use equations (10) and (11). However, this would lead to highly complex relations. This is the reason why we propose to find how image $k-j$ can be transformed into image k. Defining $\mathcal{X}(k) = [M_x(k), M_y(k), \theta(k), \vartheta(k)]^T$ as the system state at k, we propose to compute the smallest sequence of control inputs allowing to reach state at k starting from state at $k-j$. To this aim, we first need to verify the controllability of the corresponding nonlinear discrete system $\mathcal{X}(k+1) = g(\mathcal{X}(k), \dot{q}(k))$ where $g(\mathcal{X}(k), \dot{q}(k))$ is obtained by analytically solving equation (1). Its expression is given by:

$$g: \left(\begin{array}{l} M_x(k) = M_x(k-1) \\ \qquad +\frac{\upsilon(k-1)}{\omega(k-1)}(\sin(\theta(k-1)+\omega(k-1)*T) \\ \qquad -\sin(\theta(k-1))) \\ M_y(k) = M_y(k-1) \\ \qquad -\frac{\upsilon(k-1)}{\omega(k-1)}(\cos(\theta(k-1)+\omega(k-1)*T) \\ \qquad -\cos(\theta(k-1))) \\ \theta(k) = \theta(k-1)+\omega(k-1)*T \\ \vartheta(k) = \vartheta(k-1)+\varpi(k-1)*T \end{array}\right) \tag{12}$$

when $\omega \neq 0$ (the problem is straightforward if $\omega = 0$). Such a system is controllable in p steps if the following matrix P is full rank (Djeridane, 2004).

$$P = \left[\begin{array}{c} \frac{\partial g(\mathcal{X}(p-1),\dot{q}(p-1))}{\partial \dot{q}(p-1)} \\ \frac{\partial g(\mathcal{X}(p-1),\dot{q}(p-1))}{\partial \mathcal{X}(p-1)} \frac{\partial g(\mathcal{X}(p-2),\dot{q}(p-2))}{\partial \dot{q}(p-2)} \\ \ldots \\ \frac{\partial g(\mathcal{X}(p-1),\dot{q}(p-1))}{\partial \mathcal{X}(p-1)} \\ \ldots \\ \frac{\partial g(\mathcal{X}(1),\dot{q}(1))}{\partial \mathcal{X}(1)} \frac{\partial g(\mathcal{X}(0),\dot{q}(0))}{\partial \dot{q}(0)} \end{array}\right]^T \tag{13}$$

It can be shown that P is not full rank for $p = 1$. For $p = 2$, this property is fulfilled if $\omega \neq 2\eta\pi$ for $\eta \in \mathbb{N}$. Now, thanks to (12), we can compute the two control inputs allowing to reach the system state at k from the one at $k-j$. The first one aims at positionning the robot at $[M_x(k), M_y(k)]$.

The second one orientates the robot and the platform towards $[\theta(k), \vartheta(k)]$. We obtain the following equations:

$$\left\{\begin{array}{l} \upsilon_{e1} = \frac{\omega_{e1}}{2\sin(\frac{\omega_{e1}*T}{2})} * R \\ \omega_{e1} = \frac{-2\theta(k-j)}{T} + 2\arctan\left(\frac{(M_y(k)-M_y(k-j))}{(M_x(k)-M_x(k-j))}\right) \\ \varpi_{e1} = 0 \end{array}\right.$$
$$\left\{\begin{array}{l} \upsilon_{e2} = 0 \\ \omega_{e2} = \frac{\theta(k)-(\theta(k-j)+\omega_{e1}T)}{T} \\ \varpi_{e2} = \frac{\vartheta(k)-\vartheta(k-j)}{T} \end{array}\right. \tag{14}$$

where:
$R = \sqrt{(M_x(k)-M_x(k-j))^2+(M_y(k)-M_y(k-j))^2}$.

Now, thanks to the two control inputs $\dot{q}_{e1} = (\upsilon_{e1}, \omega_{e1}, \varpi_{e1})^T$ and $\dot{q}_{e2} = (\upsilon_{e2}, \omega_{e2}, \varpi_{e2})^T$ given by (14), we are able to reach the image at instant k from any image at $k-j$. It should be noted that the robot trajectory in the world frame computed with $[\dot{q}_{e1}, \dot{q}_{e2}]$ is not the same as the one calculated with the sequence $[\dot{q}(k-j), ..., \dot{q}(k-1)]$. Therefore we have to introduce an intermediate state $\mathcal{X}(j')$ to compute our predictor. It corresponds to the system state which has been reached at the virtual instant j' by applying $\dot{q}_{e1}$. Using (10) recursively, we obtain the following results:

$$\begin{array}{l} \hat{z}_i(k-j'|k-j) = \hat{z}_i(k-j|k-j)\phi_i + \varphi_i \\ \hat{z}_i(k|k-j') = \hat{z}_i(k-j'|k-j)\phi'_i + \varphi'_i \\ \hat{z}_i(k|k-j) = \hat{z}_i(k-j|k-j)\mu_i + \nu_i \end{array} \tag{15}$$

The different parameters $\phi_i, \varphi_i, \phi'_i, \varphi'_i, \mu_i, \nu_i$ and κ_i involved in these equations are given in the appendix. Now, using (15) and (11), we express a predictor using image at $k-j$ as follows:

$$\left\{\begin{array}{l} \hat{X}_i(k|k-j) = \frac{\tilde{X}_i(k-j)\hat{z}_i(k|k-j)-\nu_i}{\hat{z}_i(k|k-j)\mu_i} \\ \hat{Y}_i(k|k-j) = f\left\{\frac{\tilde{Y}_i(k-j)\cos(A(\dot{q}_{e1})T)\cos(A(\dot{q}_{e2})T)}{f\mu_i}\right. \\ \qquad -\frac{\sin(A(\dot{q}_{e1})T)\cos(A(\dot{q}_{e2})T)}{\mu_i} \\ \qquad \left.-\frac{\sin(A(\dot{q}_{e2})T)}{\phi'_i}+\frac{\gamma_i}{\hat{z}_i(k|k-j)})\right\} \end{array}\right. \tag{16}$$

Once the predictors have been obtained, in a second step, we address the estimators determination problem. To this aim, we propose to minimize the following criterion which represents the error (for one point P_i) between the above predictors and the corresponding measures at instant t_k. We get:

$$C^* = \sum_{j=1}^{m} (\hat{X}_i(k|k-j)-\tilde{X}_i(k))^2 + (\hat{Y}_i(k|k-j)-\tilde{Y}_i(k))^2 \tag{17}$$

Derivating this cost function with respect to the depth leads to:

$$\frac{\partial C^*}{\partial \hat{z}_i(k|k-j)} = \sum_{j=1}^{m} \Big\{ 2(\hat{X}_i(k|k-j) - \tilde{X}_i(k)) \frac{\partial \hat{X}_i(k|k-j)}{\partial \hat{z}_i(k|k-j)} + 2(\hat{Y}_i(k|k-j) - \tilde{Y}_i(k)) \frac{\partial \hat{Y}_i(k|k-j)}{\partial \hat{z}_i(k|k-j)} \Big\} \quad (18)$$

where:

$$\frac{\partial \hat{X}_i(k|k-j)}{\partial \hat{z}_i(k|k-j)} = \frac{\tilde{X}_i(k-j)\nu i}{\hat{z}_i^2(k|k-j)\mu_i} \quad (19)$$

and

$$\frac{\partial \hat{Y}_i(k+2|k)}{\partial \hat{z}_i(k|k-j)} = \frac{-f\gamma_i}{\hat{z}_i^2(k|k-j)} \quad (20)$$

Our estimator $\hat{z}_i(k|k)$ is then given by:

$$\hat{z}_i(k|k) = \frac{\sum_{j=1}^{m} Num_i^j}{\sum_{j=1}^{m} Den_i^j} \quad (21)$$

with

$$Num_i^j = \frac{\nu_i^2 \tilde{X}_i^2(k-j)}{\mu_i^2} + f^2\gamma^2 \quad (22)$$

and

$$\begin{aligned} Den_i^j = {} & \Big(\frac{\tilde{X}_i(k-j)}{\mu_i} - \tilde{X}_i(k)\Big)\frac{\tilde{X}_i(k-j)\nu_i}{\mu_i} \\ & - \Big\{ \frac{\tilde{Y}_i(k-j)\cos(A(\dot{q}_{e1})T)\cos(A(\dot{q}_{e2})T)}{\mu_i} \\ & - \frac{f\sin(A(\dot{q}_{e1})T))\cos(A(\dot{q}_{e2})T)}{\mu_i} \\ & - \frac{f\sin(A(\dot{q}_{e2})T)}{\phi_i'} - \tilde{Y}_i(k) \Big\} f\gamma_i(k) \end{aligned} \quad (23)$$

We have then computed an estimator for the depth using data provided by m previous images. This depth will be used as an initial condition in Folio's algorithm to reconstruct the visual data.

4 SIMULATION

We have simulated the proposed estimation method during a visual servoing task using Matlab$^{\text{TM}}$software. We present hereafter the obtained results. The considered landmark has been characterized by four interest points. The initial robot configuration has been defined by $\mathcal{X} = (5, 5, -2.35, 0)^T$ and the reference visual features s^* have been determined for the configuration $\mathcal{X} = (1.2, 0, \pi, 0)^T$. For all the presented simulations, the control input $\dot{q}$ given by (8) has been computed using $\lambda = 0.7$ and $T = 0.1s$. $z_i(k)$ is reconstructed using equation (21). The corresponding results for one point are represented on figure 2.

In the first simulation, all the data are supposed to be perfectly known. As one can see, the obtained estimation of the depth of each point p_i is perfect, which validates estimator (21) in the ideal case. Notice that a good estimated value of depth is immediately obtained, which is not the case of the method proposed in (De Luca et al., 2008). Moreover, we can see the depth estimation using Folio's algorithm. The initial condition is not the real value. The error between them is preserved during the entire simulation.

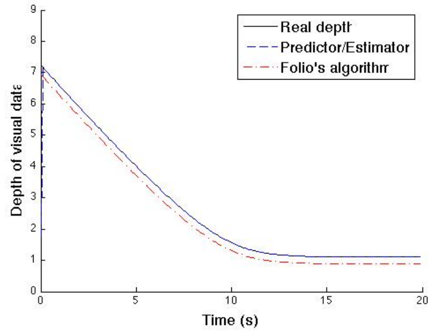


Figure 2: Evolution of both estimated and real depths of one point P_i.

Now, we aim at validating estimator (21) when the visual data are noisy. A one pixel noise has been added on $\tilde{X}_i$ and $\tilde{Y}_i$. In this case the estimator uses at most $m = 20$ images. The corresponding results for one point are represented on figure 3. In a noisy context, estimator (21) converges towards the real depth value within an acceptable time. Indeed the estimated depth value is correct after about 2 s. The number of images m used in the estimation process can be tuned to fit the performances of the considered testbed. Finally, to show that our algorithm provides an adequate value of z sufficiently rapidly, we have coupled it to D. Folio's estimation method. Thus, in the same conditions as previously, we have simulated a loss of visual data between the seventh and ninth seconds during a visual servoing task. As shown in figure 4, the estimated depth values allow to correctly reconstruct the visual features. The navigation task can then be correctly realized, although the controller has been computed with the estimated data instead of the real ones. This last result demonstrates the efficiency of the proposed approach in a noisy context. D. Folio's algo-

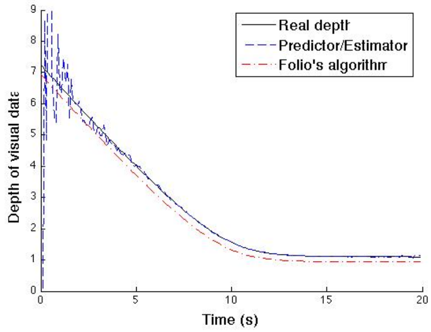


Figure 3: Evolution of both estimated and real depths.

rithm accuracy is then significantly improved thanks to our approach.

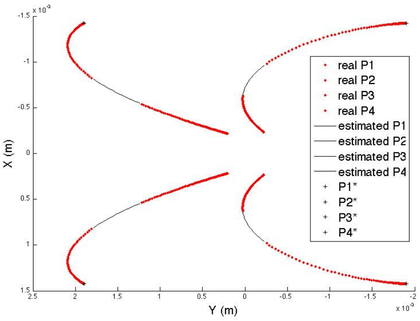


Figure 4: Evolution of estimated and real visual features.

5 CONCLUSIONS

In this paper, we have presented a method allowing to estimate the depth z_i during a vision-based navigation task. The proposed approach relies on a predictor/estimator pair able to provide an estimation of z_i, even when the visual data are noisy. The advantage of the proposed approach is that it relies on a parameterizable number of images, which can be adjusted depending on the computation abilities of the considered processor. The reconstructed depth value is then used to feed Folio's algorithm, increasing its accuracy. The obtained results have proven the efficiency of our technique in a noisy context. Up to now, we have only used the estimated value of z_i to improve Folio's work. In the future, we plan to benefit from this value at two different levels. The first one concerns the control law design with the computation of $L_{(s,z)}$. The approximations classically made in the visual servoing area could then be overcome. The second level is related to the determination of the reference visual signals s^*. This term is computed either experimentally by taking an image at the desired position or theoretically by means of models. These solutions significantly reduce autonomy. We believe that a precise estimation of the depth can be very helpful to automatically on-line compute the value of s^*, suppressing the above mentioned drawbacks. Finally, another challenging aspect of our future work will consist experimenting our approach on a real robot.

REFERENCES

Cervera, E., Martinet, P., and Berry, F. (2002). Robotic manipulation with stereo visual servoing. *Robotics and Machine Perception, SPIE International Group Newsletter*, 1(1):3.

Chaumette and Hutchinson (2006). Visual servo control, part 1 : Basic approaches. *IEEE Robotics and Automation Magazine*, 13(4).

Comport, Pressigout, Marchand, and Chaumette (2004). Une loi de commande robuste aux mesures aberrantes en asservissement visuel. In *Reconnaissance des Formes et Intelligence Artificielle*, Toulouse, France.

Corke, P. (1996). *Visual control of robots : High performance visual servoing*. Research Studies Press LTD.

De Luca, Oriolo, and Giordano (2008). Features depth observation for image based visual servoing: theory and experiments. *Int. Journal of Robotics Research*, 27(10).

Djeridane (2004). *Sur la commandabilité des systèmes non linéaires à temps discret*. PhD thesis, Université Paul Sabatier - Toulouse III.

Durand Petiteville, A., Courdesses, M., and Cadenat, V. (2009). Reconstruction of the features depth to improve the execution of a vision-based task. In *9th International workshop on Electronics, Control, Modelling, Measurement and Signals 2009*, Mondragon, Spain.

Espiau, Chaumette, and Rives (1992). A new approach to visual servoing in robotics. *IEEE Trans. Robot. Automat.*, 8:313–326.

Folio and Cadenat (2008). *Computer Vision*, chapter 4. Xiong Zhihui; IN-TECH.

Jerian, C. and Jain, R. (1991). Structure from motion: a critical analysis of methods. *IEEE Transactions on systems, Man, and Cybernetics*, 21(3):572–588.

Ma, Y., Soatto, S., Kosecka, J., and Sastry, S. (2003). *An invitation to 3-D vision: from images to geometric models*. New York: Springer-Verlag.

Matthies, Kanade, and Szeliski (1989). Kalman filter-based algorithms for estimating depth in image sequences. *Int, Journal of Computer Vision*, 3(3):209–238.

Pissard-Gibollet and Rives (1995). Applying visual servoing techniques to control a mobile handeye system. In *IEEE Int., Conf. on Robotics and Automation*, Nagoya, Japan.

Samson, Borgne, and Espiau (1991). Robot control : The task function approach. *Oxford science publications*.

APPENDIX

Parameters for equation (11):

$$\left\{\begin{array}{l}
\alpha_i = \frac{\tilde{Y}_i(k-1)}{f}\sin(A(\dot{q}(k-1))T)+\cos(A(\dot{q}(k-1))T) \\
\beta = \Big\{-D_x\sin(\vartheta(k-1))-\frac{\upsilon(k-1)}{\omega(k-1)}\cos(\vartheta(k-1))+C_y\Big\} \\
\qquad \sin(A(\dot{q}(k-1))T)+ \\
\qquad \Big\{D_x\cos(\vartheta(k-1))-\frac{\upsilon(k-1)}{\omega(k-1)}\sin(\vartheta(k-1))+C_x\Big\} \\
\qquad \cos(A(\dot{q}(k-1))T) \\
\qquad -D_x\cos(\vartheta(k))+\frac{\upsilon(k-1)}{\omega(k-1)}\sin(\vartheta(k))-C_x \\
\kappa_i = \Big\{\frac{-\tilde{Y}_i(k-1)\beta}{f\alpha_i}-D_x\sin(\vartheta(k-1))-\frac{\upsilon(k-1)}{\omega(k-1)}\cos(\vartheta(k-1))+C_y\Big\} \\
\qquad \cos(A(\dot{q}(k-1))T)- \\
\qquad \Big\{\frac{-\beta}{\alpha_i}+D_x\cos(\vartheta(k-1))-\frac{\upsilon(k-1)}{\omega(k-1)}\sin(\vartheta(k-1))+C_x\Big\} \\
\qquad \sin(A(\dot{q}(k-1))T)+D_x\sin(\vartheta(k))+\frac{\upsilon(k-1)}{\omega(k-1)}\cos(\vartheta(k))-C_y
\end{array}\right.$$

Parameters for equations (20) and (21):

$$\left\{\begin{array}{l}
\vartheta(j') = \vartheta(k-j)+\varpi_{e1}T \\
\vartheta(k) = \vartheta(k-j)+(\varpi_{e1}+\varpi_{e2})T
\end{array}\right.$$

$$\left\{\begin{array}{l}
\phi_i = \frac{\tilde{Y}_i(k-j)}{f}\sin(A(\dot{q}_{e1})T)+\cos(A(\dot{q}_{e1})T) \\
\varphi_i = \Big\{-D_x\sin(\vartheta(k))-\frac{\upsilon_{e1}}{\varpi_{e1}}\cos(\vartheta(k-j))+C_y\Big\}\sin(A(\dot{q}_{e1})T) \\
\qquad +\Big\{D_x\cos(\vartheta(k-j))-\frac{\upsilon_{e1}}{\varpi_{e1}}\sin(\vartheta(k-j))+C_x\Big\}\cos(A(\dot{q}_{e1})T) \\
\qquad -D_x\cos(\vartheta(j'))+\frac{\upsilon_{e1}}{\varpi_{e1}}\sin(\vartheta(j'))-C_x
\end{array}\right.$$

$$\left\{\begin{array}{l}
\phi'_i = \frac{\tilde{Y}_i(k-j)}{f\phi}\cos(A(\dot{q}_{e1})T)\sin(A(\dot{q}_{e2})T)-\frac{\sin(A(\dot{q}_{e1})T)\sin(A(\dot{q}_{e2})T)}{\phi} \\
\qquad +\cos(A(\dot{q}_{e2})T) \\
\varphi'_i = \Big[\Big\{\frac{\tilde{Y}_i(k-j)\varphi_i}{f\phi_i}-D_x\sin(\vartheta(k-j))-\frac{\upsilon_{e1}}{\omega_{e1}}\cos(\vartheta(k-j))+C_y)\Big\} \\
\qquad \cos(A(\dot{q}_{e1})T) \\
\qquad -\Big\{\frac{-\varphi_i}{\phi_i}+D_x\cos(\vartheta(k-j))-\frac{\upsilon_{e1}}{\omega_{e1}}\sin(\vartheta(k-j))+C_x\Big\} \\
\qquad \sin(A(\dot{q}_{e1})T) \\
\qquad +(\frac{\upsilon_{e1}}{\omega_{e1}}-\frac{\upsilon_{e2}}{\omega_{e2}})\cos(\vartheta(j'))\Big]\sin(A(\dot{q}_{e2})T) \\
\qquad +\Big\{D_x\cos(\vartheta(j'))-\frac{\upsilon_{e2}}{\omega_{e2}}\sin(\vartheta(j'))+C_x\Big\}\cos(A(\dot{q}_{e2})T) \\
\qquad -D_x\cos(\vartheta(k))-\frac{\upsilon_{e2}}{\omega_{e2}}\sin(\vartheta(k))+C_x
\end{array}\right.$$

$$\left\{\begin{array}{l}
\mu_i = \phi_i\phi'_i \\
\nu_i = \varphi_i\phi'_i+\varphi'_i
\end{array}\right.$$

$$\left\{\begin{array}{l}
\gamma_i = \Big[\Big\{\frac{-\tilde{Y}_i(k-j)\nu_i}{f\mu_i}-D_x\sin(\vartheta(k-j))-\frac{\upsilon_{e1}}{\omega_{e1}}\cos(\vartheta(k-j))+C_y\Big\} \\
\qquad \cos(A(\dot{q}_{e1})T) \\
\qquad -\Big\{\frac{-\nu_i}{\mu_i}+D_x\cos(\vartheta(k-j))-\frac{\upsilon_{e1}}{\omega_{e1}}\sin(\vartheta(k-j))+C_x\Big\} \\
\qquad \sin(A(\dot{q}_{e1})T) \\
\qquad +(\frac{\upsilon_{e1}}{\omega_{e1}}-\frac{\upsilon_{e2}}{\omega_{e2}})\cos(\vartheta(j'))\Big]\cos(A(\dot{q}_{e2})T) \\
\qquad -\Big\{\frac{-\varphi'_i}{\phi'_i}+D_x\cos(\vartheta(j'))-\frac{\upsilon_{e2}}{\omega_{e2}}\sin(\vartheta(j'))+C_x\Big\} \\
\qquad \sin(A(\dot{q}_{e2})T) \\
\qquad +D_x\sin(\vartheta(k))+\frac{\upsilon_{e2}}{\omega_{e2}}\cos(\vartheta(k))-C_y
\end{array}\right.$$

NORMAL FLAT FORMS FOR A CLASS OF 0-FLAT AFFINE DYNAMICAL SYSTEMS AND ITS APPLICATION TO NONHOLONOMIC SYSTEMS

S. Bououden, D. Boutat
PRISME - ENSI de Bourges, 10 boulevard Lahitolle, 18020, Bourges cedex, France
soriabouden@yahoo.fr, driss.boutat@ensi-bourges.fr

F. Abdessemed
Faculty of Engineering, Electronic Departement, Batna University, 05000, Algeria
fodil_a@hotmail.com

Keywords: Flatness, Nonlinear systems, Normal form, Involutivity, Nonholonomic systems.

Abstract: In this contribution normal flat forms are used to achieve stable tracking control for nonlinear flat systems. Our approach is based on a nonlinear transformations in order to derive two 0-flat normal forms for a class of non-linear systems, a dynamical control law is then proposed to achieve stable trajectory tracking. Finally, This method is generalized to analysis and control a class of a 0-flat affine nonlinear multi-input dynamical systems for which we can build flat outputs to give structural normal flat forms. The computer simulations are given in the paper to demonstrate the advantages of the method.

1 INTRODUCTION

The control of nonholonomic dynamic systems has received considerable attention during the last years and become a popular subject in the nonlinear control. One a reasons for this, in real world, the nonholonomic systems are frequently used to describe some pratical control systems such as mobile robot, car-like vehicule, and under-actuated satellites, can all be modeled as nonholonomic control systems or nonholonomic maneuvers. Hence, control problems involve them have attracted attention in the control community.

Different methods have been applied to solve motion control problems. (Kanayama et al., 1991) propose a stable tracking control method for nonholonomic vehicule using a Lyapunov function. (Lee et al., 1998) solved tracking control using backstepping and in (Lee and Tai, 2001) with saturation constraints. Furthermore, most reported designs rely on intelligent control approaches such as Fuzzy Logic Control (Pawlowski et al., 2001), (Tsai et al., 2004), Neural Networks (Song and Sheen, 2000), and (Chwa, 2004) used a sliding mode control to the tracking control problem. (Fierro and Lewis, 1995) propose a dynamical extention that makes possible the integration of kinematic and torque controller for a nonholonomic mobile robot. (Fukao et al., 2000), introduces an adaptive tracking controller for the dynamic model of mobile robot with unknown parameters using backstepping. However the field of control of such systems is still open to develop other control strategie.

Application of flatness to problems of engineering interest have grown steadily in recent years. Michel Fliess et al. (Fliess et al., 1992), (Fliess et al., 1995) introduced the concept of flat outputs, these outputs guarantee that the problem will be put in term of control algorithm for motion planning, trajectory generation and stabilization. A limitation of flatness is that there does not exist necessary and sufficient conditions to determine if a general system is differentially flat and there no algorithm to compute the flat outputs. Nevertheless, it is well-known that all controllable linear systems can be shown to be flat. Indeed, any system that can be transformed into a linear system by changes of coordinates, static feedback transformations, or dynamic feedback transformations is also flat (Jakubczyk and Respondek, 1980), (Hunt et al., 1983).

We present in this paper two normal flat forms, It deals with sufficient geometrical conditions which enable us to conclude if a given nonlinear controllable dynamical system can be transformed, by means of change of coordinates, to one of these normal forms.

In the same way it gives an algorithm to compute the flat outputs. As an illustration to the proposed approach, a trajectory tracking of a nonholonomic uniaxial vehicule is simulated. As we will show with this example, for this particular class, our method presents a new direction to solve the flatness problem.

The paper is organized as follows. In section 2 we address notations, definitions and our problem statement, we describes the classes of 0-flat systems study. The necessary and sufficient geometrical conditions for affine dynamical systems are presented in Section 3. In sections 4 provides illustrative examples and simulations. Some conclusions are presented in section 5.

2 DEFINITION AND PROBLEM STATEMENT

Let us consider the following class of multivariable nonlinear systems described in state space form by equations of the following kind In the next section, we will recall the concept of 0-flat systems

$$\dot{x} = f(x) + \sum_{i=1}^{m} g_i(x)u_i \tag{1}$$

where $x \in \chi \subseteq \Re^n, u \in U \subseteq \Re^m$ and f is a smooth function on $\chi \times U$.

Definition 1. The dynamical system (1) is flat if there exist m functions $y = (y_1, ..., y_m)$ called the flat outputs such that:

1- $y = F(x, u, \dot{u}, ..., u^{(r_1)})$ is a function of the state x, the input u, and its derivatives $u^{(i)}$

2- $x = \varphi(z, \dot{z}, ..., z^{(r_2)})$ is a function of the flat outputs and their derivatives.

3- $u = \gamma(z, \dot{z}, ..., z^{(r_2+1)})$ is a function of the flat outputs and their derivatives.

In this paper, we will deal with multi-input affine dynamical systems, without loss of generality, we will assume within this work that:

Assumption 1. The vector field $G = [g_1, ..., g_m]$ is of rank m.
However, we don't assume that $\Delta = span(G)$ is involutive as it is the case in many researches which try to compute the inverse dynamics. We will characterize a class of dynamical systems for which the flat outputs are only functions of states x. This dynamical systems is called 0-flat: (Pereira, 2000)

$$y_i = F_i(x);\ where\ 1 \leq i \leq m \tag{2}$$

2.1 A Class of Structurally 0-Flat Dynamical System

It is well-known that the codimension one $(m = n - 1)$ controllable dynamical systems are 0-flat (Charlet and Lévine, 1989). Our result concerns a normal flat forms of affine dynamical systems with n states and $m = n - 2$ inputs (codimension 2). Our objective is to introduce our formulation.

2.2 Main Result

In the sequel we introduce the following notation $z_{ji} = z_{j,i} = y_j$ with $j = 1 : m$ relative to the flat output y_j, which means $i - 1$ derivation of the output y_j.

2.2.1 First 0-Flat Normal Form

Let us consider the following proposition written in terms of the z_{j1}, where $j = 1 : m$ state variables. Let us consider the following proposition

Proposition 1. The following dynamical system is 0-flat

$$\begin{aligned}
\dot{z}_{11} &= z_{12} + \sum_{i=3}^{n-2} \beta_{11}^{i}(z)u_i \\
\dot{z}_{12} &= \alpha_{12}(z) + u_1 + \beta_{12}^{2}(z)u_2 + ... + \beta_{12}^{m}(z)u_m \\
\dot{z}_{21} &= z_{22} + \sum_{i=3}^{n-2} \beta_{21}^{i}(z)u_i \\
\dot{z}_{22} &= \alpha_{22}(z) + \beta_{22}^{1}(z)u_1 + u_2 + ... + \beta_{22}^{m}(z)u_m \\
\dot{z}_{j1} &= \alpha_{j1} + u_j + \sum_{i=3, i\neq j}^{n-2} \beta_{j1}^{i}(z)u_i
\end{aligned} \tag{3}$$

with z_{11}, z_{21} as flat outputs, and $m = n - 2$.

Proof 1. Now we will show that the equations (3) represent a locally dynamical 0-flat system. Consider the following m equations :

$$E_1 = \dot{z}_{1,1} - z_{1,2} - \sum_{i=3}^{m} \beta_{11}^{i}(z)u_i = 0 \tag{4}$$

$$E_2 = \dot{z}_{2,1} - z_{2,2} - \sum_{i=3}^{m} \beta_{21}^{i}(z)u_i = 0 \tag{5}$$

$$E_j = \dot{z}_{j,1} - \alpha_{j,1} - u_j - \sum_{i=3, i\neq j}^{m} \beta_{j1}^{i}(z)u_i = 0 \tag{6}$$

where $3 \leq j \leq m$

Let $v = (z_{1,2}, z_{2,2}, u_3, ..., u_m)$ be the vector of unknown system variables and let us compute the following partial derivative

$$\frac{\partial(E_1, ..., E_m)}{\partial v} = -I_m + O(v) \tag{7}$$

Where I_m is the identity matrix and $o(v)$ represents the order one of the v variables. From the equations (4, 5, 6) and the fact that $\frac{\partial(E_1,\ldots,E_m)}{\partial v}$ is locally invertible then the implicite function theorem, allows us to conclude that there exists $\varphi_k()$ and $\gamma_k()$ functions such that

$$z_{k,2} = \varphi_k(y_j,\dot{y}_j, j = 1:m, k = 1:2) \quad (8)$$

$$u_k = \gamma_k(y_j,\dot{y}_j, j = 1:m, k = 3:m) \quad (9)$$

By replacing (8) and (9) in the second and the fourth dynamic equation of (3) we can get the inputs u_1 and u_2 as functions of $(y_j,\dot{y}_j)$ for $j = 1:m$ and their second derivatives $y_1^{(2)}$, $y_2^{(2)}$. In the next subsection we will give a slightly different 0-flat normal form which is related to more drastic conditions.

2.2.2 Second 0-Flat Normal Form

The second canonical system gives the missing variables from the successive derivation of the same flat output written in terms of the variables $z_{j1} = y_j, 1 \leq j \leq (n-2)$:

Proposition 2. The following dynamical system is 0-flat

$$\begin{aligned}
\dot{z}_{11} &= z_{12} + \sum_{i=2}^{n-2} \beta_{11}^i(z)u_i \\
\dot{z}_{12} &= z_{13} + \sum_{i=2}^{n-2} \beta_{12}^i(z)u_i \\
\dot{z}_{13} &= \alpha_{13}(z) + u_1 + \sum_{i=2}^{n-2} \beta_{13}^i(z)u_i \\
\dot{z}_{j1} &= \alpha_{j1(z)} + u_j + \sum_{i=2}^{n-2} \beta_{j1}^i(z)u_i
\end{aligned} \quad (10)$$

Where $2 \leq j \leq m$, and $m = n-2$.

Proof 2. The main difference from (3) concerns the fact that we assume the variable z_{13} not present in the dynamics $\dot{z}_{j1}$, for $1 \leq j \leq (n-2)$. Then we can conclude that

Condition 1. z_{13} must not be present in $\beta_{j1}^i(z)$ for $j = 1:m$, and $i = 2:m$

Condition 2. z_{13} must not be present in α_{j1} for $j = 2:m$.

Under these conditions we can use the same procedure as the canonical form (3) to solve the dynamical system (10). So let us consider the m equations:

$$E_1 = \dot{z}_{11} - z_{12} - \sum_{i=2}^{n-2} \beta_{11}^i(z)u_i = 0 \quad (11)$$

$$E_j = \dot{z}_{j1} - \alpha_{j1(z)} - u_j - \sum_{i=2}^{n-2} \beta_{j1}^i(z)u_i = 0 \quad (12)$$

We can put $v = (z_{12}, u_2, \ldots, u_m)$ the vector of unknown system variables. Let us compute the following partial derivative

$$\frac{\partial(E_1,E_2\ldots,E_m)}{\partial v} = -I_m + o^1(v) \quad (13)$$

Where I_m is the identity matrix and $o^1(v)$ represents the order one of the v variables. From the equations (11, 12) and the fact that $\frac{\partial(E_1,E_2,\ldots,E_m)}{\partial v}$ is locally inversible then the implicite function theorem, allows us to conclude that there exists $\varphi_1()$ and $\gamma_k()$ functions such that

$$z_{1,2} = \varphi_1(y_j,\dot{y}_j, j = 1:m) \quad (14)$$

$$u_k = \gamma_k(y_j,\dot{y}_j, j = 1:m, k = 2:m) \quad (15)$$

By replacing (14) and (15) in the second and the third dynamic equation of (10) we can get the variable z_{13} as a function of $\dot{y}_1$, $y_1^{(2)}$ and y_i for $i = 1:m$. Also we get the input variable u_1 as a function of $\dot{y}_1$, $y_1^{(2)}$, $y_1^{(3)}$and y_i for $i = 1:m$.

3 TRANSFORMATIONS

Now, we give some conditions for a class of nonlinear systems, for which we can transform a nonlinear dynamical system (1) in a new 0-flat normal forms. So we distinguished two cases:

Case 1. The controllability distribution has the following vector field:

$\Delta_1 = span\{g_1, g_2, \ldots, g_{n-2}, ad_f g_1, ad_f g_2\}$, with $dim(\Delta_2) = n$.

Case 2. The controllability distribution has the following:

$\Delta_2 = span\{g_1, g_2, \ldots, g_{n-2}, ad_f g_1, ad_f^2 g_1\}$, with $dim(\Delta_2) = n$.

3.1 Case 1

Proposition 1. If the distribution $\overline{\Delta_1} = span\{g_1, g_2\} \subseteq \Delta_1$ is involutive, then the dynamical system (1) is 0-flat.

Proof 1. As $dim(\Delta_1) = n$ and $\overline{\Delta_1}$ is a 2-dimensional involutive distribution, there exist $n-2 = m$ independent functions of states x, $(y_i = z_{i,1}), 1 \leq i \leq m$ such that:

- $\overline{\Delta_2} = \bigcap_{i=1}^m \ker dy_i$ where $\ker dy_i$ means the kernel of the differential of the function y_i.

- $L_{[g_k,f]}z_{k,1} = 1$ for $k = 1:2$
- $L_{g_k}z_{k,1} = 1$ for $3 \le k \le m$

Now let us consider for $k = 1:2$ the following new variables: $z_{k,2} = L_f z_{k,1}$ where L_f is the Lie derivative in the direction of f. Therefore the set of the n variables: $(z_{i,1}), 1 \le i \le m$, $z_{1,2}$ and $z_{2,2}$ form a new coordinate system. For this the derivative of these variables give structural 0-flat normal form (3).

3.2 Case 2

Now let us describe the above conditions **(1), (2)** in (Proof 2) geometrically. For this one remarks that in (10) we have $g_1 = \frac{\partial}{\partial z_{1,3}}$. Therefore the independence of the others input directions $(g_i)_{2 \le i \le m}$ from the variables $z_{1,3}$ can be described by the fact that:

$$[g_1, g_k,] \in span\{g_1, ad_f g_1\}.$$

Indeed $ad_f g_1 = \frac{\partial}{\partial z_{1,3}} + h(z)\frac{\partial}{\partial z_{1,2}}$. For second condition, $\alpha_{j,1} = L_f z_{j,1}$ for all $j \ge 2$. Then, if we want this function independent of the variable $z_{1,3}$, then we must have $L_{g_1}L_f z_{j,1} = 0$. As u_1 is not present in the last equations of (10) then $L_{g_1}z_{j,1} = 0$. Therefore, $L_{g_1}L_f z_{j,1} = 0$ is equivalent to $L_{ad_f g_1}z_{j,1} = 0$. Thus we can conclude that the distribution $\overline{\Delta_2} = \{g_1, ad_{f g_1}\}$ is involutive.

Proposition 3. If the distribution $\overline{\Delta_2} = \{g_1, ad_{f g_1}\} \subseteq \Delta_2$ is involutive and $[g_1, g_2] \in span\{g_{1,}ad_{f g_1}\}$ then the dynamical system (1) is 0-flat.

4 ILLUSTRATIVE EXAMPLE

In order to verify the performance of proposed methodology, as an illustration, we used a nonholonomic system. A nonlinear transformation is made in order to derive a 0-flat normal forms, the results obtained with our proposed control based on 0-flat normal forms of codimension 2 are used to control the nonlinear system in the aim to show its usefulness.

4.1 Application to a Nonholonomic Uniaxal Vehicule

The example we study is the kinematic model of a mobile car (see Figure 1), this system can be represented by the following set of equations:

$$\begin{cases} \dot{x}_1 &= vcos\theta \\ \dot{x}_2 &= vcos\theta \\ \dot{\theta} &= \omega \end{cases} \tag{16}$$

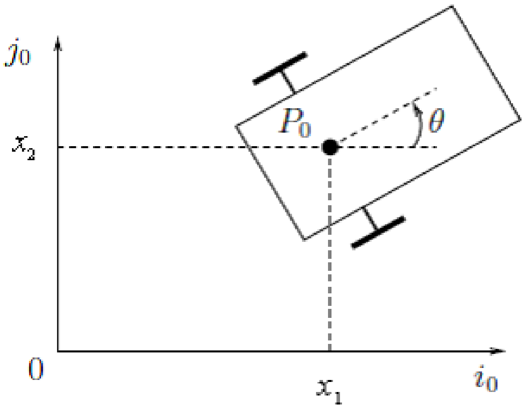


Figure 1: Coordinates system of car.

where
(x_1, x_2) represents the cartesian position of the centre of mass of the car,
θ its inclination with respect to the horizontal axis,
(v, ω) its forward and angular velocities respectively.
the model (16) is not static feedback linearizable. However, the problem can be solved by introducing the new state $x_4 = \sqrt{\dot{x_1}^2 + \dot{x_2}^2}$.
Resulting in the extended static feedback linearizable system described by:

$$\begin{cases} \dot{x}_1 &= x_4 cos(x_3) \\ \dot{x}_2 &= x_4 sin(x_3) \\ \dot{x}_3 &= u_1 \\ \dot{x}_4 &= u_2 \end{cases} \tag{17}$$

The outputs are the states variables: $x^T = (x_1, x_2, x_3, x_4)$, the above equations can be written in the following form:

$$\dot{x} = f(x) + g_1(x)u_1 + g_2(x)u_2; \tag{18}$$

where $f(x), g_1(x), g_2(x)$ are the smooth vector fields.
We will transform the nonlinear dynamical system (17) in a structural 0-flat normal form, the distribution
$\overline{\Delta_1} = \{g_1, g_2\}$ has dimension 2.
The bracket $[g_1, g_2] = 0$ which means that $\overline{\Delta_1}$ is involutive. Then there exists two functions $y_1(x)$ and $y_2(x)$ such that:

$$dy_1(x).\overline{\Delta}_1 = 0 \tag{19}$$

$$dy_2(x).\overline{\Delta}_1 = 0 \tag{20}$$

From (19) and (20) we can conclude that $\frac{\partial y_i}{\partial x_3} = 0$ and $\frac{\partial y_i}{\partial x_4} = 0 \; \forall x_1, x_2$. Then it is enough to set $y_1(x)$ a function of x_1 and $y_2(x)$ any function in terms of x_2. Let us consider $y_1 = x_1$ and $y_2 = x_2$.
Now let us consider for $k = 1:2$ the following variables: $z_{k,1} = y_k$ and the new coordinate variables:
$z_{1,2} = L_f z_{1,1}$ and $z_{2,2} = L_f z_{2,1}$.

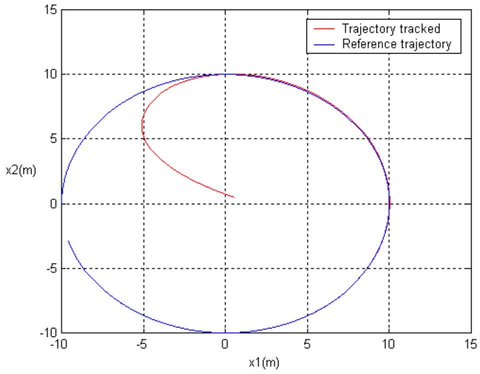


Figure 2: Trajectory tracked by uniaxal vehicule.

For this the derivative of these variables give structural 0-flat normal form (3).

$$
\begin{aligned}
\dot{z}_{11} &= z_{12}+\sum_{i=3}^{n-2}\beta_{11}^{i}(z)u_i \\
\dot{z}_{12} &= \alpha_{12}(z)+u_1+\beta_{12}^{2}(z)u_2+\ldots+\beta_{12}^{m}(z)u_m \\
\dot{z}_{21} &= z_{22}+\sum_{i=3}^{n-2}\beta_{21}^{i}(z)u_i \qquad (21)\\
\dot{z}_{22} &= \alpha_{22}(z)+\beta_{22}^{1}(z)u_1+u_2+\ldots+\beta_{22}^{m}(z)u_m
\end{aligned}
$$

The system is codimension 2, the canonical form can be expressed as follows:

$$
\left\{
\begin{aligned}
\dot{z}_{11} &= z_{12} \\
\dot{z}_{12} &= \alpha_{12}(z)+\beta_{11}^{1}(z)u_1+\beta_{12}^{2}(z)u_2 \\
\dot{z}_{21} &= z_{22} \\
\dot{z}_{22} &= \alpha_{22}(z)+\beta_{22}^{1}(z)u_1+\beta_{22}^{2}(z)u_2
\end{aligned}
\right. \qquad (22)
$$

$$
\begin{aligned}
&\alpha_{12}=0,\alpha_{22}=0 \\
\text{where}\quad &\beta_{12}^{1}(z)=x_4 sin(x_3);\beta_{12}^{2}(z)=x_4 cos(x_3) \\
&\beta_{22}^{1}(z)=cos(x_3);\beta_{22}^{2}(z)=sin(x_3)
\end{aligned}
$$

The main objective of the flatness based controller is to obtain the asymptotic tracking of a desired trajectory, let the system output be $y_1=z_{11}$, $y_2=z_{21}$, from (22) we can obtain the expressions of u_1, u_2 in terms of $(y_1,\dot{y}_1,y_2,\dot{y}_2)$, and the second derivatives of two first variables $\ddot{y}_1$, $\ddot{y}_2$. Let $\ddot{y}_1=v_1,\ddot{y}_2=v_2$ two new inputs control such that:

$$
\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} \ddot{y}_{d_1}+k_d(\dot{y}_{d_1}-\dot{y}_1)+k_p(y_{d_1}-y_1) \\ \ddot{y}_{d_2}+k_d(\dot{y}_{d_2}-\dot{y}_2)+k_p(y_{d_2}-y_2) \end{bmatrix}
$$

where $k_d,k_p>0$ are control gains chose carefully to ensure exponential stability, and $\ddot{y}_d,\dot{y}_d,y_d$ are prescribed reference trajectories.
The control law:

$$
\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} \beta_{11}^{1}(z) & \beta_{12}^{2}(z) \\ \beta_{22}^{1}(z) & \beta_{22}^{2}(z) \end{bmatrix}^{-1} \begin{bmatrix} v_1-\alpha_{12}(z) \\ v_2-\alpha_{22}(z) \end{bmatrix}
$$

The controller gains are chosen to be:

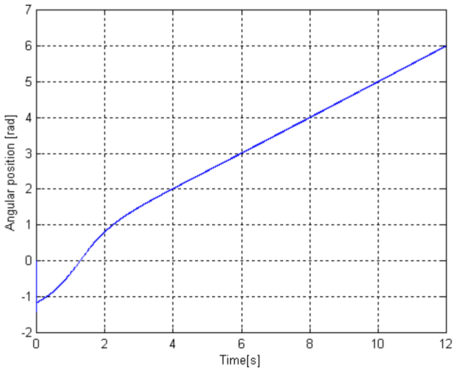


Figure 3: Unicycle orientation x_3.

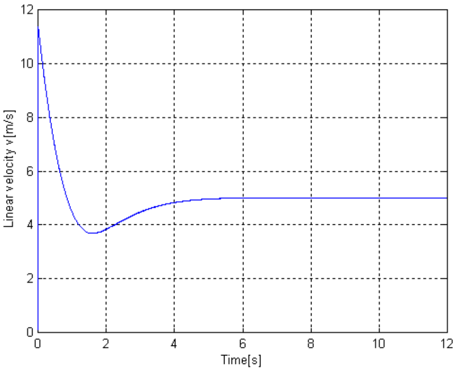


Figure 4: Linear velocity of unicycle vehicule.

$$k_p=diag[1000,1000], k_d=diag[1000,1000].$$

We consider the following continuously derivable desired trajectories:

$y_{d1}=rsin\frac{v_0}{r}t$, $y_{d2}=-rcos\frac{v_0}{r}t$ is assigned, which described a circular movement with radius $r=10$ at the constant speed $v_0=5$.

The closed loop system was simulated using the initial conditions $x^T(0)=(0.5;0.5;0;0.1)$ for system (17).

The trajectory tracked (see Fig. 2) is very close to the desired one, achieving by thus the control objective. Finally, angular position and linear velocity are depicted in the Fig. 3 and Fig. 4. It can be seen that our control scheme achieves satisfactory performances.

5 CONCLUSIONS

We described the development of a tracking controller based on normal flat forms. This method is generalized to analysis and control a class of a 0-flat affine nonlinear multi-input dynamical systems for which we can build flat outputs to give structural normal flat forms. Simulation results show an acceptable performance under the studied cases.

REFERENCES

Fliess, M., Lévine, J., Martin, P. and Rouchon, P., 1992. Sur les systmes non linaires diffrentiellement plats. C. R. Acad. Sci. Paris Sr. I Math., 315: pp 619-624.

Fliess, M., Lévine, J., Martin, P. and Rouchon, P., 1995. Flatness defect of non-linear systems introductory theory and examples. Internat. J. Control, 61: pp 1327-1361.

Chwa, D., 2004. Sliding- Mode Tracking Control of nonholonomic Wheeled Mobile robotin polar coordinates, IEEE Trans. On Control Syst. Tech. Vol. 12, No. 4,pp 633-644.

Fierro, R., Lewis, F. L., 1995. Control of Nonholonomic Mobile Robot: Backstepping Kinematics into Dynamics, Proc. 34th Conf. On Decision & Control, New Orleans, LA.

Fukao, T., Nakagawa, N. Adachi, 2000. Adaptive Tracking Control of a Nonholonomic Mobile Robot. IEEE Trans, On Robotics and Automation, Vol. 16, No. 5, pp609-615.

Kanayama, Y., Kimura, Y., Miyazaki, F., Noguchi, T. 1991. A stable Tracking control method for a Non-Holonomic Mobile Robot. Proc. IEEE/RSJ Int. Workshop on Intelligent Robots and Systems, Osaka, Japan, pp 1236-1241.

Lee, T-C., Lee, C.H., Teng, C-C. 1998. Tracking Control of Mobile Robots Using the Backsteeping Technique, Proc. 5th. Int. Conf. Contr., Automat Robot. Vision, Singapore, pp 1715-1719.

Lee, T-C., Tai, K., 2001. Tracking Control of Unicycle Modeled Mobile robots using a saturation feedback controller, IEEE Trans. On Control Systems Technology, Vol. 9, No. 2, pp 305-318.

Pawlowski, S., Dutkiewicz, K. Kozlowski, Wroblewski, W., 2001. Fuzzy logic implementation in Mobile robot Control, 2th Workshop On Robot Motion and Control, pp 65-70.

Tsai, C-C., Lin, H-H, Lin, C-C., 2004. Trajectory Tracking control of a Laser-Guided wheeled mobile robot, Proc. IEEE Int. Conf. On Control Application, Taipei, Taiwan, pp 1055-1059.

Song, K. T., Sheen, L. H., 2000. Heuristic fuzzy-neural Network and its application to reactive navigation of a mobile robot, Proc. IEEE Int. Conf. On Control Application, Taipei, Taiwan, pp 1055-1059.

Pereira da Silva., 2000. Flatness of nonlinear control systems a Cartan-Kähler approach, In Proc. Mathematical Theory of Networks et Systems MTNS'2000, pages 1 10, Perpignan, Jun. 19-23.

Jakubczyk, B., Respondek, W., 1980. On linearization of control systems, Bull. Acad. Pol. Sc., Ser. Sci. Math., 28: pp 517-522.

Hunt, L. R., Su, R. and Meyer, G., 1983. Global transformations of nonlinear systems, IEEE Trans. Automat. Control, 28:24-31.

Charlet, B., Lévine. J., On dynamic feedback linearization, Systems Control Lett., 13: pp 143-151.

BEARING-ONLY SAM USING A MINIMAL INVERSE DEPTH PARAMETRIZATION
Application to Omnidirectional SLAM

Cyril Joly and Patrick Rives

INRIA Sophia Antipolis Méditerranée, 2004 route des Lucioles, Sophia Antipolis Cedex, France

{*cyril.joly, patrick.rives*}*@sophia.inria.fr*

Keywords: Simultaneous localization and mapping (SLAM), Smoothing and mapping (SAM), Extended Kalman filter (EKF), Bearing-only, Inverse depth representation.

Abstract: Safe and efficient navigation in large-scale unknown environments remains a key problem which has to be solved to improve the autonomy of mobile robots. SLAM methods can bring the map of the world and the trajectory of the robot. Monucular visual SLAM is a difficult problem. Currently, it is solved with an Extended Kalman Filter (EKF) using the inverse depth parametrization. However, it is now well known that the EKF-SLAM become inconsistent when dealing with large scale environments. Moreover, the classical inverse depth parametrization is over-parametrized, which can also be a cause of inconsistency. In this paper, we propose to adapt the inverse depth representation to the more robust context of smoothing and mapping (SAM). We show that our algorithm is not over-parameterized and that it gives very accurate results on real data.

1 INTRODUCTION

Simultaneous localization and mapping (SLAM) is a fundamental and complex problem in mobile robotics research which has mobilized many researchers since its initial formulation by Smith and Cheeseman ((Smith and Cheeseman, 1987)). The robot moves from an unknown location in an unknown environment and proceeds to incrementally build up a navigation map of the environment, while simultaneously using this map to update its estimated position. Traditional methods are based on the extended Kalman filter (EKF). It is now well known that EKF based methods yield inconsistencies ((Julier and Uhlmann, 2001; Bailey et al., 2006a)). This is essentially due to linearization errors. Furthermore, these algorithms suffer from computational complexity ($O(N^2)$ where N is the number of landmarks in the map).

The FastSLAM method, introduced by Thrun and Montemerlo ((Montemerlo et al., 2003)), is based on the particle filter. Each position is approximated by a set of M random particles and one map is associated to each particle. It can be shown that $M \log N$ complexity is achievable. However, the algorithm is sensitive to the number of particles chosen. Furthermore, the issue of particles diversity can make the FastSLAM inconsistent ((Bailey et al., 2006b)).

In the above-mentioned methods, SLAM is solved as a filtering problem: the state-space contains only the state of the robot at the current time and the state of the map. SLAM can also be stated as a smoothing problem: the whole trajectory is taken into account (Thrun and Montemerlo, 2006; Dellaert and Kaess, 2006). This approach is referred as *simultaneous smoothing and mapping* (SAM). Although it is based on the linearization of the equations, the approach gives better results than EKF because all the linearization points are adjusted during the optimization.

A very interesting problem is the monocular visual SLAM which is also referred as Bearing-Only SLAM. Since camera does not measure directly the distance between the robot and the landmark, there is an observability issue. In standard approach, landmarks are initialized with a delay. Recent approaches try to avoid this and propose undelayed formulation. The method presented in (Civera et al., 2008) consists in changing the parametrization of the landmarks. This parametrization includes the location of the first position where the landmark was observed and the inverse depth between the landmark and this position.

Also the latter algorithm produces good results in real situations, it has some drawbacks. First, the parametrization of the landmarks is not minimal. Then, this parametrization is often associated with the EKF (although it is inconsistent).

In this paper, we propose a new way to implement the inverse depth parametrization by using a SAM approach. First, the underlying hypothesis of the SLAM and the general equations are given in section 2. Then, section 3 presents the SAM algorithm and the particularities of the inverse depth parametrization with SAM. Section 4 gives some elements of comparison between the EKF and the SAM approach. Finally, results are presented and discussed on real data acquired by an omnidirectional camera on board of a mobile robot.

2 BEARING-ONLY SLAM

2.1 Notation and Hypotheses

In the following, we denote:

- $\mathbf{x}_t$ the robot state at time t.
- $\mathbf{u}_t$ the control inputs of the model at time t.
- $\mathbf{m}_{(i)}$ the i^{th}-landmark state. Concatenation of all landmarks state is $\mathbf{m}$
- $\mathbf{z}_{t(i)}$ the measurements of landmark i at t. Concatenation for all the landmark is $\mathbf{z}_t$

In this paper, we propose to compare two methods for computing the robot and landmarks states: a new implementation of the SAM algorithm and the classical EKF. Both methods use the same hypotheses:

- the robot state $\mathbf{x}_t$ is modeled as a Markov chain,
- conditioned to all the trajectory and the map, measurements at time t depend only on $\mathbf{x}_t$ and $\mathbf{m}$ and are independent of measurements at other time.

Finally, the densities functions for the prediction of the state of the robot and the observations are assumed to be Gaussian:

$$\left\{ \begin{array}{rcl} p\left(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{u}_t\right) & = & \mathcal{N}\left(\mathbf{f}\left(\mathbf{x}_{t-1},\mathbf{u}_t\right),\mathbf{Q}_t\right) \\ p\left(\mathbf{z}_t|\mathbf{x}_t,\mathbf{m}\right) & = & \mathcal{N}\left(\mathbf{h}\left(\mathbf{x}_t,\mathbf{m}\right),\mathbf{R}_t\right) \end{array} \right. \qquad (1)$$

where $\mathbf{f}$ and $\mathbf{h}$ are the prediction and observation functions, $\mathbf{Q}_t$ and $\mathbf{R}_t$ the covariance matrices of the model and the measurements.

2.2 Camera Motion

Let us consider a standard 6DOFs BO-SLAM configuration, i.e. the robot moves in the 3D space and observes 3D-landmarks with a camera. No more sensor is considered (no odometry and no IMU). The robot position at time t is given by its Euclidian coordinates ($\mathbf{r}_t = [x_t, y_t, z_t]^T$) and its orientation by a quaternion ($\mathbf{q}_t$).

Since we do not measure the linear and angular velocities, they are estimated in the state. Linear velocity is defined in the global frame as $\mathbf{v}_t = [v_{xt}, v_{y_t}, v_{zt}]^T$ and angular velocity is defined in the camera frame as $\Omega_t = [\omega_{xt}, \omega_{y_t}, \omega_{zt}]^T$.

Furthermore, we assume that constant linear and angular accelerations act as an input on the system between t and $t+1$. It produces increments $\mathbf{V}_k$ and Ω_k. They are modeled as Gaussian white noise with zero mean. However, the velocity is not exactly constant between two time steps, so that the position and rotation are not exactly given by the integration of the velocities defined in the state. We take that into account by adding two error terms (ν_v and ν_ω) in the model (they are taken as Gaussian white noise).

Finally, the function $\mathbf{f}$ which stands for the camera motion is given by:

$$\mathbf{x}_{k+1} = \begin{bmatrix} \mathbf{r}_{k+1} \\ \mathbf{q}_{k+1} \\ \mathbf{v}_{k+1} \\ \Omega_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{r}_k + (\mathbf{v}_k + \mathbf{V}_k + \nu_v)\Delta t \\ \mathbf{q}_k \times \mathbf{q}\left((\Omega_k + \Omega_k + \nu_\omega)\Delta t\right) \\ \mathbf{v}_k + \mathbf{V}_k \\ \Omega_k + \Omega_k \end{bmatrix} \qquad (2)$$

where $\mathbf{q}(\mathbf{u})$ is the quaternion defined by the rotation vector $\mathbf{u}$.

2.3 Inverse Depth Representation

It is well known that several observations are necessary to estimate a landmark. Indeed, a camera gives a measurement of the direction of the landmarks, but not their distance. So, landmarks initialisation is delayed when using the classical euclidian representation (a criterion based on the parallax is generally used). However, landmarks can bring bearing information when the depth is unknown. An interesting solution was given by *Civera et al* ((Civera et al., 2008)). It consists in using a new landmark representation based on the *inverse depth*. In this scheme, a 3D point (i) can be defined by a 6D vector:[1]

$$\mathbf{y}_{(i)} = \begin{bmatrix} x_{(i)} & y_{(i)} & z_{(i)} & \theta_{(i)} & \phi_{(i)} & \rho_{(i)} \end{bmatrix}^T \qquad (3)$$

where $x_{(i)}$, $y_{(i)}$, $z_{(i)}$ are the euclidian coordinates of the camera position from which the landmark was observed for the first time; $\theta_{(i)}$, $\phi_{(i)}$ the azimuth and elevation (expressed in the world frame) definining unit directional vector $\mathbf{d}(\theta_{(i)}, \phi_{(i)})$ and $\rho_{(i)}$ encodes the inverse of the distance between the first camera position and the landmark. So, we can write:

$$\mathbf{m}_{(i)}^{\textit{Euclidian state}} = \begin{bmatrix} x_{(i)} \\ y_{(i)} \\ z_{(i)} \end{bmatrix} + \frac{1}{\rho_{(i)}}\mathbf{d}(\theta_{(i)}\phi_{(i)}) \qquad (4)$$

$$\mathbf{d} = \begin{bmatrix} \cos\phi_{(i)}\cos\theta_{(i)} & \cos\phi_{(i)}\sin\theta_{(i)} & \sin\phi_{(i)} \end{bmatrix}^T$$

[1] in the following, (i) stands for the landmark number. Parenthesis are used for landmarks in order to avoid confusion between the time index and landmarks index.

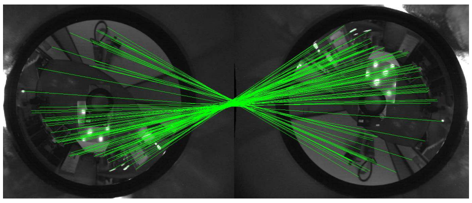


Figure 1: Data association for a loop closure. The angle between the two views is close to π. This case shows the advantage of the omnidirectional camera. Such loop closure would have been impossible with a perspective camera looking forward.

The main advantage of this representation is that it satisfies linearity condition for measurement equation even at very low parallax. According to (Civera et al., 2008),landmarks can be initialized with only one measurement.

2.4 Measurement Equation

2.4.1 Specificity of the Omnidirectional Camera Model

3D points are observed by an omnidirectional camera, which is the combination of a classical perspective camera and a mirror. For the low-level image processing, we use algorithms specifically designed to deal with omnidirectional images (see (Hadj-Abdelkader et al., 2008) for more details). An example of data association in omnidirectional images is given on Fig 1. First, Harris points are extracted from the current image. Data association with the points extracted from the previous image is performed thanks to a Sift descriptor. Finally, we obtain a set of matched points of interest in pixels coordinates.

The points in omnidirectional images are linked to the landmarks coordinates by the unified projection model (Barreto, 2003; Mei, 2007). It is done in the four steps which are recalled below. Let $\mathcal{X}$ be the euclidian coordinates of a point in the camera frame:

1. First, $\mathcal{X}$ is projected onto the unit sphere: $\mathcal{X}_s = \frac{\mathcal{X}}{\|\mathcal{X}\|} = [X_s \; Y_s \; Z_s]$
2. The third coordinate of $\mathcal{X}_s$ is shifted by ξ (mirror parameter): $\tilde{\mathcal{X}_s} = [X_s \; Y_s \; Z_s + \xi]$
3. Then, the result is projected onto the unit plane: $\mathbf{u} = [\frac{X_s}{Z_s+\xi} \; \frac{Y_s}{Z_s+\xi} \; 1]$
4. Finaly, the result is projected onto the image by $\mathbf{p} = \mathbf{Ku}$ where $\mathbf{K}$ is the calibration matrix of the system.

In the following, let us write $\mathbf{h}_1$ the function that returns $\mathbf{p}$ from $\mathcal{X}$:

$$\mathbf{p} = \mathbf{h}_1(\mathcal{X}) \tag{5}$$

2.4.2 Final Measurement Equation

It is shown in (Civera et al., 2008) that the Euclidian coordinates of landmark (i) in camera frame at time t are proportional to:

$$\mathbf{h}_2(\mathbf{y}_{(i)}, \mathbf{r}_t, \mathbf{q}_t) = \mathbf{R}_t^{CW} \times \\ \times \left(\rho_{(i)} \left(\begin{bmatrix} x_{(i)} \\ y_{(i)} \\ z_{(i)} \end{bmatrix} - \mathbf{r}_k \right) + \mathbf{d}(\theta_{(i)}, \phi_{(i)}) \right)$$

where $\mathbf{R}_t^{CW}$ is the rotation matrix relative to camera frame and world frame. It is the transpose of the rotation matrix associated to $\mathbf{q}_t$.

Finally, the measurement equation $\mathbf{h}$ is given by:

$$\mathbf{h} = \mathbf{h}_1 \circ \mathbf{h}_2 \tag{6}$$

3 SMOOTHING AND MAPPING

3.1 SAM Strategy

In this paper, we propose a SAM method for computing the states of the robot and the landmarks. So, we are trying to obtain $p(\mathbf{x}_{0:t}, \mathbf{m} | \mathbf{z}_{0:t}, \mathbf{u}_{1:t})$, which is the joint probability of the trajectory and the map when all the measurements and inputs are known. So, the state contains all the trajectory of the robot and the landmarks. However, SAM methods can be implemented in real-time due to the block structure of the information matrix and the use of re-ordering rules ((Dellaert and Kaess, 2006)).

3.2 SAM Equations

With the previous hypotheses, and assuming that no prior knowledge is available for the landmarks, the *pdf* of the trajectory and landmarks can be formulated as:

$$p(\mathbf{x}_{0:t}, \mathbf{m} | \mathbf{z}_{0:t}, \mathbf{u}_{1:t}) \propto p(\mathbf{x}_0)\, p(\mathbf{z}_0 | \mathbf{x}_0, \mathbf{m}) \times \\ \prod_{k=1}^{t} [p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_k)\, p(\mathbf{z}_k | \mathbf{x}_k, \mathbf{m})] \tag{7}$$

Taking the logarithm of (7), and assuming that the functions $\mathbf{f}$ and $\mathbf{h}$ can be linearized around the approximated trajectory defined by $\mu^T = [\mu_{0:t}^{\mathbf{x}}{}^T, \mu^{\mathbf{m}T}]^T$, $\log p(\mathbf{x}_{0:t}, \mathbf{m} | \mathbf{z}_{0:t}, \mathbf{u}_{1:t})$ yields:

$$
\begin{aligned}
&\log p\left(\mathbf{x}_{0:t},\mathbf{m}|\mathbf{z}_{0:t},\mathbf{u}_{1:t}\right) = \mathrm{cst} - \tfrac{1}{2}\mathbf{x}_0^T\,\mathcal{I}_0\mathbf{x}_0 + \mathbf{x}_0^T\,\xi_0 \\
&+\Sigma_{k=1}^{t}\left\{-\tfrac{1}{2}\begin{bmatrix}\mathbf{x}_k\\ \mathbf{x}_{k-1}\end{bmatrix}^T\begin{bmatrix}\mathbf{I}\\ -\mathbf{G}_k^T\end{bmatrix}\mathbf{Q}_k^{-1}\begin{bmatrix}\mathbf{I} & -\mathbf{G}_k\end{bmatrix}\begin{bmatrix}\mathbf{x}_k\\ \mathbf{x}_{k-1}\end{bmatrix}\right. \\
&\quad\left.+\begin{bmatrix}\mathbf{x}_k\\ \mathbf{x}_{k-1}\end{bmatrix}^T\begin{bmatrix}\mathbf{I}\\ -\mathbf{G}_k^T\end{bmatrix}\mathbf{Q}_k^{-1}\left(\mathbf{f}_k\left(\mu_{k-1}^{\mathbf{x}},\mathbf{u}_k\right)+\mathbf{G}_k\mu_{k-1}^{\mathbf{x}}\right)\right\} \\
&+\Sigma_{k=0}^{t}\left\{-\tfrac{1}{2}\begin{bmatrix}\mathbf{x}_k\\ \mathbf{m}\end{bmatrix}^T\begin{bmatrix}(\mathbf{H}_k^{\mathbf{x}})^T\\ (\mathbf{H}_k^{\mathbf{m}})^T\end{bmatrix}\mathbf{R}_k^{-1}\begin{bmatrix}\mathbf{H}_k^{\mathbf{x}} & \mathbf{H}_k^{\mathbf{m}}\end{bmatrix}\begin{bmatrix}\mathbf{x}_k\\ \mathbf{m}\end{bmatrix}\right. \\
&\quad\left.+\begin{bmatrix}\mathbf{x}_k\\ \mathbf{m}\end{bmatrix}^T\begin{bmatrix}(\mathbf{H}_k^{\mathbf{x}})^T\\ (\mathbf{H}_k^{\mathbf{m}})^T\end{bmatrix}\mathbf{R}_k^{-1}\left(\begin{array}{c}\mathbf{z}_k-\mathbf{h}(\mu_k^{\mathbf{x}},\mu^{\mathbf{m}})+\\ \begin{bmatrix}\mathbf{H}_k^{\mathbf{x}} & \mathbf{H}_k^{\mathbf{m}}\end{bmatrix}\begin{bmatrix}\mu_k^{\mathbf{x}}\\ \mu^{\mathbf{m}}\end{bmatrix}\end{array}\right)\right\}
\end{aligned}
\tag{8}
$$

where $\mathbf{G}_k$ is the Jacobian of $\mathbf{f}$ with respect to $\mathbf{x}_k$, $\mathbf{H}_k^{\mathbf{x}}$ (resp. $\mathbf{H}_k^{\mathbf{m}}$) is the Jacobian of $\mathbf{h}$ with respect to $\mathbf{x}_t$ (resp. $\mathbf{m}$) (the Jacobian are computed at μ). $\mathcal{I}_0$ and ξ_0 are the information matrix and vector associated to $\mathbf{x}_0$. If no prior information is available on $\mathbf{x}_0$, we set these variables to zero. Equation (8) shows that $\log p\left(\mathbf{x}_{0:t},\mathbf{m}|\mathbf{z}_{0:t},\mathbf{u}_{1:t}\right)$ is quadratic in $\left[\mathbf{x}_{0:t}^T\ ,\ \mathbf{m}^T\right]^T$. Therefore, $p\left(\mathbf{x}_{0:t},\mathbf{m}|\mathbf{z}_{0:t},\mathbf{u}_{1:t}\right)$ is Gaussian, and the computation of the information parameters are provided by (8). This yields to a sparse matrix $\mathcal{I}$[2] which simplifies the recovering of the mean and covariance.

3.3 SAM Implementation

SAM is a filtering algorithm which uses all the data to compute all the trajectory of the robot and the map. It needs a prediction of all the trajectory and the positions of the landmarks. To simulate real time conditions, we decide to use it incrementally (instead of using once all the measurements). Let us assume that we have already an estimation of $\mu_{1:t-1}^{\mathbf{x}}$ and $\mu^{\mathbf{m}}$. When the inputs and measurements are available, we update our estimation in two steps:

1. First, we compute a prediction of $\mu_t^{\mathbf{x}}$ by applying $\mathbf{f}$
2. Then, we compute $\mathcal{I}$ and ξ with Eq. (8)
3. μ is given by $\mathcal{I}^{-1}\xi$

Usually, the computation of $\mathcal{I}$ and ξ is done several times up to convergence. However, we add only one pose at each step. Thus, one iteration is enough in practice with our implementation.

3.4 Minimal Landmark Parametrization

3.4.1 Definition of the Parametrization

Let us discuss about the expression of the vector $\mathbf{m}$ that is estimated in the SAM algorithm. We use an inverse depth representation in order to initialize $\mathbf{m}$ at the first iteration. However, the inverse depth representation is not minimal since it uses a 6D vector to encode a 3D point. We can arbitrarily fix the three first components and compute the three others. Thus, the 3 last components are enough to represent a landmark in the state if we fix the first ones. Now, let us assume that 3 first components of $\mathbf{y}_{(i)}$ are *fixed* to $\tilde{x}_{(i)}$, $\tilde{y}_{(i)}$ and $\tilde{z}(i)$, so that we have $\mathbf{m}_{(i)} = [\theta_{(i)}\ \phi_{(i)}\ \rho_{(i)}]^T$.

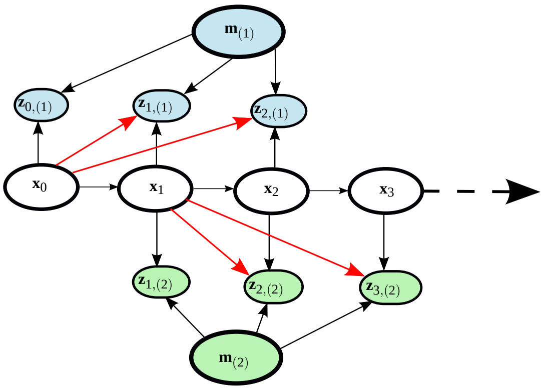
However, we can not choose an arbitrary value for $\tilde{x}_{(i)}, \tilde{y}_{(i)}$ and $\tilde{z}(i)$ if the goal is to insert $\mathbf{m}_{(i)}$ in the filter at the first iteration. Indeed, the linearity constraints has to be respected, which implies that the approximated trajectory has to be close enough to the true one. To do that, $\tilde{x}_{(i)}$, $\tilde{y}_{(i)}$ and $\tilde{z}_{(i)}$ are fixed to $\mu_k^{\mathbf{x}}$ (assuming that first observation of landmark (i) is at time k). Choosing any other value would lead to delay the initialization since we can not guarantee that the angles will be close enough.

Finally, we can notice that the values chosen for $\tilde{x}_{(i)}$, $\tilde{y}_{(i)}$ and $\tilde{z}(i)$ are valid for one iteration of the SAM algoritnm. Indeed, imagine that a part of the estimated trajectory has changed significantly after several time steps (because of a loop closure for example). The values of the fixed coordinates have to be update for the next step in the same way than μ^x. Reader should note that this is not an “ estimation ” (in a filter point of view) of $\tilde{x}_{(i)}, \tilde{y}_{(i)}$ and $\tilde{z}_{(i)}$ but a shift of equilibrium point in order to optimize the validity of the linearization. We do not compute the jacobian of $\mathbf{h}$ with respect to these fixed components.

3.4.2 Discussion

The parametrization presented in the previous paragraph takes advantage of the inverse depth representation but remains minimal since only 3 parameters are estimated. Using a minimal representation reduce the risk of “ falling in local minima ”. Furthermore, we can show that keeping the six variables in the estimation would lead to severe inconsistencies in the case of SAM. Assume for example that landmark (i) was first observed at time k and that we use the six variables in $\mathbf{m}_{(i)}$. At time k, the three first variables of $\mathbf{m}_{(i)}$ are fully correlated with $\mathbf{r}_k$. Let us examine what happens for each new observation of $\mathbf{m}_{(i)}$ for $t > k$:

[2] Sparseness results from the classical assumption that measurement of landmark (i) is independent of measurement of landmark $(j) \neq (i)$ ($\mathbf{R}_k$ matrix is block diagonal).



(a)

Figure 2: Bayesian network in the case of 2 landmarks. Black links indicates normal links of the network. The red links indicates spurious links due to over-parametrization for the EKF case (they are not present with the SAM).

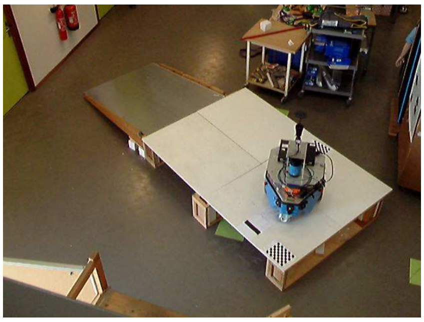


Figure 3: Photo of the experiment.

1. First, the observation will give information about the current variable $\mathbf{x}_k$ and $\mathbf{m}_{(i)}$. This gain of information implicitly improves the estimation of all the other variables of the filter by inference, including $\mathbf{r}_{(k)}$
2. Then, we should remember that the first 3 components of $\mathbf{m}_{(i)}$ are fully correlated with $\mathbf{r}_k$ which implies that the information added on $x_{(i)}, y_{(i)}$ and $z(i)$ are directly propagated up to $\mathbf{r}_k$.

The second source of information gathered by $\mathbf{r}_k$ is spurious information. In fact, the full inverse depth parametrization adds shorcuts in the Bayesian Graph associated to $\mathbf{x}$ and $\mathbf{m}$ (Figure 2). With this model, the measurement of landmark (i) depends directly on the current robot pose, the landmark and the first position of observation. **The second hypothesis given in paragraph 2.1 is not satisfied**. In practice, such implementation of the SAM algorithm will lead to divergence.

Finally, an intuitive way to understand the theoretical differences between the EKF formulation and the minimal SAM representation is to examine the basic properties of the measurement function:

1. in the case of EKF, $\mathbf{h}$ is a function of $\mathbf{r}_k$, $\mathbf{x}_t$ and $\mathbf{m}_{(i)}$. So, we have $\mathbf{z}_t = \mathbf{h}(\mathbf{r}_k, \mathbf{x}_t, \mathbf{m}_{(i)})$,
2. in the case of minimal SAM representation, $\mathbf{h}$ is a function of $\mathbf{x}_t$ and $\mathbf{m}_{(i)}$ parameterized by $\mu_k^{\mathbf{x}}$. So, we have $\mathbf{z}_t = \mathbf{h}_{\mu_k^{\mathbf{x}}}(\mathbf{x}_t, \mathbf{m}_{(i)})$.

4 COMPARISON WITH THE EKF

In this section, we propose to compare some aspects of the EKF-SLAM with the SAM. First, it is now well known that the EKF algorithm is inconsistent. It was shown in simulations by Tim Bailey (Bailey et al., 2006a). Moreover, Simon Julier made a proof of the inconsistency in the case of a static Robot (Julier and Uhlmann, 2001).

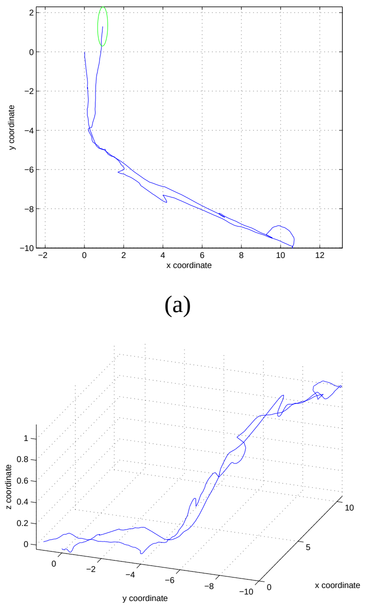
More recently Huang shown the general inconsistency in the case of 3 degrees of freedom SLAM by an analysis of observability ((Huang et al., 2008)). Indeed, an observability of the 3DOF-SLAM shows that the system is unobservable and that the observability matrix has a nullspace of dimension 3. Intuitively, it corresponds to a rigid motion of the whole system (2 translations and one rotation). However, the authors of (Huang et al., 2008) shown that the observability matrix associated to the system linearized by the EKF has a nullspace of dimension 2. This corresponds to spurious information gain in rotation.

They also present an other filter derivated from the EKF (the *The First-Estimates Jacobian EKF SLAM (FEJ-EKF)*) which *does not update the linearization points of the landmarks*. This new filter respects the observabilty nullspace dimension. So, Huang claims that the root of the EKF inconsistency is the updates of equilibrium points. **We can notice that the formulation of the SAM algorithm guarantees that the linearization points for the landmarks are the same for all the observations**. In consequence, we can guess that the good properties of the FEJ-EKF are kept by the SAM algorithm.

5 RESULTS

5.1 Experimental Testbed

Both algorithms were tested with true data acquired by a mobile robot in an indoor environment. The trajectory length is about 15m. **All the results were obtained in the case of pure bearing-only SLAM (no odometry).**



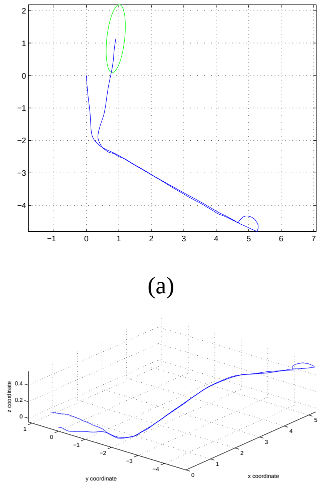
(a)

(b)

Figure 4: Trajectory provided by the EKF-SLAM.

In order to test the capability of the algorithms to estimate the 6 DOF of the robot, we made a plateform at 35cm high from the ground (Figure 3). To access to this plateform, the robot is driven on a 180cm long ramp. This gives us a ground truth concerning the angle of the trajectory of the robot during its “ climb ”:

$$\gamma = \arcsin(35/180) \approx 11.2deg \qquad (9)$$

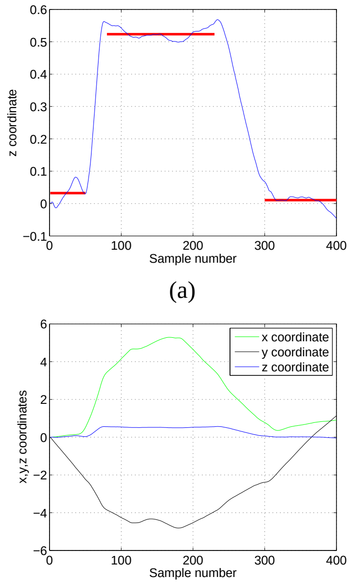


(a)

(b)

Figure 5: Trajectory provided by the SAM algorithm.

Furthermore, the robot is submitted to high changes in rotation velocity. Particulary, the robot makes a turn just before climbing on the plateform. Then, the robot rotates on itself (without translation) once it is on the plateform. Some pictures of the experiment are proposed on Figure 3. As the camera is not mounted on the center of rotation of the robot, we expect to see a semicircle in the trajectory. The end of the trajectory consists in going down to the ground and moving in the plane. We also used the fact that the robot came back close to the first position to force a loop closure (the loop closure correspond to the data association shown on Fig 1). Both algorithms are tested with exactly the same data (the image processing is done off-line). Finally, even if we do not have a ground truth on all the trajectory, we have 4 parameters which can be tested:

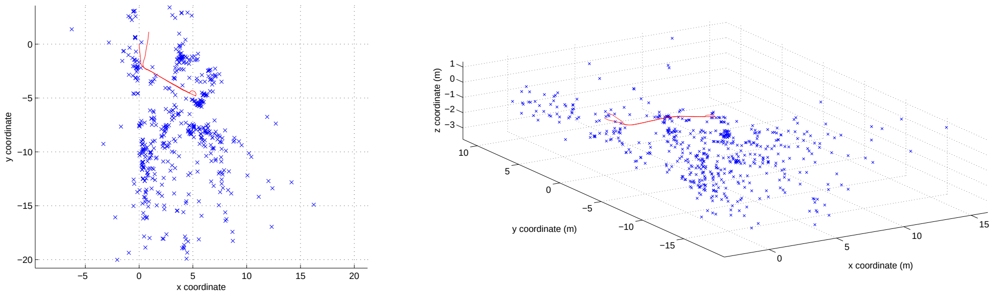


(a)

(b)

Figure 6: SAM solution — (a) z coordinate – (b) x, y and z coordinates: we see the relevance of z with respect to the main scale (x and y).

1. The robot is on the ground during the first part of the trajectory. So, we expect the z coordinate of the trajectory to be zero,
2. The angle of the ramp is known. We can use it as a ground truth,
3. After the climb, the robot movement is parallel to the ground. So, we expect the z coordinate to be a constant,
4. The robot is on the ground during the last part of the trajectory.



(a)

(b)

Figure 7: Map provided by the SAM algorithm. The trajectory is also plotted in red.

5.2 Results on the EKF

The trajectory given by the EKF is given on Figure 4. The trajectory starts from the origin. The green ellipsis on Figure 4a shows the uncertainty of the last robot pose. We can see that there are many discontinuities in the trajectory at the end due to the loop closure. This is due to the fact that when the robot observe an old landmark, the drift accumulated force the filter to make a discontinuity in the solution. Finally, even if the trajectory looks fine (we see the different heights and the semicircle), results of the EKF are not much impresive.

5.3 Results on the SAM

The results provided by the SAM algorithm are given on Figure 5 to 7. They look like much interesting than the EKF-SLAM results. Indeed, the estimation of the trajectory is very smooth comparing to the one provided by the EKF. Moreover, we can see that the uncertainty ellipsis is bigger in the case of the SAM algorithm. This is certainly a manifestation of the EKF inconsistency: the EKF is over-confident.

Moreover, we were able to compute an approximation of the angle γ with the trajectory by using the coordinates at the beginning and at the end of the climb. We got 10.1deg (the ground truth is 11.2deg). We have just an **error of 1.1deg**. This confirms the excellent results provided by the SAM algorithm.

We also checked the relevance of the z coordinate (Figure 6). It appears that the mean value during the first part of the trajectory is 0.03m with a standard deviation of 0.03m (first red bar on Figure 6a). During the second part of the trajectory (after the climb), the mean value of z is 0.52m (standard deviation: 0.015m). Finally, the mean value of z during the last part of the trajectory is 0.01 (standard deviation: 0.02m). These results are good: we get the two parts were z is close from 0 and see a near constant z during the second part of the trajectory. Then, z values are scaled with the values of x and y on Figure 6b: we can see that displacements in z are much smaller than displacement in x,y. However, the shape of the z is well estimated, which shows the good precision of the algorithm. We got similar results with the EKF algorithm (but with a higher level of noise).

Finally, the map provided by the SAM algorithm is given on Figure 7. It is quite difficult to interpret since we do not have any ground truth (the map is made by interest points selected during the video sequence). Nevertheless, we can see on the left part a wall (close to the zero vertical axis) which seems to be coherent.

6 CONCLUSIONS

In this paper, we proposed an accurate method to solve the monocular SLAM problem. To do it, we used the inverse depth representation for landmarks and the SAM algorithm for the filtering part. However, we do not use the whole inverse depth parametrization as proposed by the original authors. Indeed, we decide to fix the three first components of the representation by using the linearization points provided by the SAM algorithm. Thus, we solved the over-parametrization problem introduced by the EKF-version of the algorithm. In consequence, two origins of inconsitencies are avoided: the one due to the inherent inconsistency of EKF, and the one due to the over-parametrization. This makes our algorithm more robust than the EKF proposed by *Civera et al.*

We shown that we had a very good estimation without any odometry. First, we recovered the angle of the ramp if good precision. Then the z coordinate estimation was quite relevant in spite of the small vertical displacements comparing to the x and y coordinates.

Our future work will focus on optimizing the algorithm and implementing a strategy to switch from inverse depth coordinates to classical euclidian coordinates. Indeed, inverse depth parametrization is very relevant when parallax is small. Then, classical coordinates can be used. The authors of (Civera et al., 2007) propose a test to switch from inverse depth coordinates to global ones. We will focus on adapting this strategy to our algorithm.

REFERENCES

Bailey, T., Nieto, J., Guivant, J., Stevens, M., and Nebot, E. (2006a). Consistency of the ekf-slam algorithm. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Bailey, T., Nieto, J., and Nebot, E. (2006b). Consistency of the fastslam algorithm. In *IEEE International Conference on Robotics and Automation*.

Barreto, J. P. (2003). *General Central Projection Systems, modeling, calibration and visual servoing*. PhD thesis, Department of electrical and computer engineering.

Civera, J., Davion, A. J., and Montiel, J. M. (2008). Inverse Depth Parametrization for Monocular SLAM. *IEEE Transactions On Robotics*, 24(5).

Civera, J., Davison, A. J., and Montiel, J. (2007). Inverse depth to depth conversion for monocular slam. In *International Conference on Robotics and Automation (ICRA)*.

Dellaert, F. and Kaess, M. (2006). Square Root SAM: Simultaneous Location and Mapping via Square Root Information Smoothing. *Journal of Robotics Research*.

Hadj-Abdelkader, H., Malis, E., and Rives, P. (2008). Spherical Image Processing for Accurate Odometry with Omnidirectional Cameras. In *The Eighth Workshop on Omnidirectional Vision*.

Huang, G., Mourikis, A., , and Roumeliotis, S. (2008). Analysis and improvement of the consistency of extended kalman filter based slam. In *Proceedings of the 2008 IEEE International Conference on Robotics and Automation (ICRA)*.

Julier, S. J. and Uhlmann, J. K. (2001). A counter example to the theory of simultaneous localization and map building. In *International Conference on Robotics and Automation (ICRA)*.

Mei, C. (2007). *Laser-Augmented Omnidirectional Vision for 3D Localisation and Mapping*. PhD thesis, Ecole des Mines de Paris, INRIA Sophia Antipolis.

Montemerlo, M., Thrun, S., Koller, D., and Wegbreit, B. (2003). FastSLAM 2.0: An Improved Particle Filtering Algorithm for Simultaneous Localization and Mapping that Provably Converges. In *International Joint Conference on Artificial Intelligence*, pages 1151–1156.

Smith, R. and Cheeseman, P. (1987). On the representation and estimation of spatial uncertainty. *International Journal of Robotics Research*, 5(4):56–68.

Thrun, S. and Montemerlo, M. (2006). The GraphSLAM Algorithm with Applications to Large-Scale Mapping of Urban Structures. *The International Journal of Robotics Research*, 25(5-6):403–429.

A FLEXIBLE ROBOTICS AND AUTOMATION SYSTEM
Parallel Visual Processing, Realtime Actuator Control and Task Automation for Limp Object Handling

Thomas Müller, Binh An Tran and Alois Knoll
Robotics and Embedded Systems, Faculty of Informatics, Technische Universität München
Blotzmannstr. 3, 85748 Garching, Germany
muelleth@cs.tum.edu, tran.binhan@mytum.de, knoll@cs.tum.edu

Keywords: Flexible automation, Parallel processing, Realtime actuator control, Limp object handling.

Abstract: In this paper, an intrinsically parallel framework striving for increased flexibility in development of robotic, computer vision, and machine intelligence applications is introduced. The framework comprises a generic set of tools for realtime data acquisition, robot control, integration of external software components and task automation. The primary goal is to provide a developer- and user-friendly, but yet efficient base architecture for complex AI system implementations, be it for research, educational, or industrial purposes. The system therefore combines promising ideas of recent neuroscientific research with a blackboard information storage mechanism, an implementation of the multi-agent paradigm, and graphical user interaction.

Furthermore, the paper elaborates on how the framework's building blocks can be composed to applications of increasing complexity. The final target application includes parallel image processing, actuator control, and reasoning to handle limp objects and automate handling-tasks within dynamic scenarios.

1 INTRODUCTION

These days, robots are common in industrial production setups. They accompany assembly lines all over the world, as they have interesting properties for production processes: they never tire, provide high accuracy, and are able to work in environments not suitable for humans. Still, todays industrial robots are often limited to very specific repetitive tasks, as they must be statically programmed ("teached") in advance. But industrial applications nowadays tend to require greater flexibility, as the manufactured products become highly customized and thus the production scenarios become more complex. Also, automation engineers strive to advance to production fields like handling of limp or deformable objects, that have not been considered before. In this context sensorimotor integration, visual servoing, and adaptive control are some of the most prominent buzz-words being investigated in the recent past by academics.

The flexible robotics and automation framework presented in Section 3 refers to these topics and provides a generic, configurable, and interactive; but nevertheless sound and efficient foundation for such tasks. Section 4 then shows, how the proposed framework can be used to build an application for robot-assisted, semi-automated limp object handling.

2 RELATED WORK

This section elaborates on how the proposed framework fits into findings / results of existing recent research in related fields. Furthermore, common robotics frameworks are mentioned and their relation to the presented system is discussed.

2.1 Related Research

We find relevant sophisticated approaches primarily within the area of cognitive and blackboard architectures, or multi-agent systems.

Cognitive architectures originate from psychology and by definition try to integrate *all* findings from cognitve sciences into a general framework from which intelligence may arise. Multiple systems have been proposed to fulfill this requirement, including Act-R (Newell, 1994; Anderson, 2007) and Soar (Lehman et al., 2006). Although these approaches may be biologically plausible and have the potential to form the foundation of some applications in reality, they all face the problem of being a mere scientific approach to cognition. We argue that, in order to develop applications also suitable for industrial automation, a framework for intelligent robotics and sensorimotor integration has to be designed keeping this scope in mind. Furhtermore, additional requirements like robustness and repeatability, generic interfacing, user-friendlyness and graphical interaction have to be taken into account with high priority. Still, we are impressed by the incredible performance of biological cognitive systems and, although we do not propose a cognitive architecture, try to integrate certain aspects where we find them useful and appropriate.

The principle theory considering **blackboard architectures** is based on the assumption, that a common database of knowledge, the "blackboard", is filled with such by a group of experts (Erman et al., 1980). The goal is to solve a problem using contributions of multiple specialists. We adopt the basic ideas of this concept in the implementation of the information storage of the proposed framework (see Section 3.3). Nevertheless, a drawback with traditional blackboard systems is the single expert, i.e., a processing thread, that is activated by a control unit (Corkill, 2003). There is no strategy for concurrent data access in parallel scenarios. Futhermore, there is no space for training an expert over time, e.g., applying machine learning techniques, or even exchanging a contributor with another one in an evolutionary fashion. We deal with these shortcomings within the proposed framework and present our approach in Section 3.2 and 3.3.

Finally, a **multi-agent system** (MAS) is a system composed of a group of agents. According to a common definition by Wooldridge (Wooldridge, 2009) an agent is a software entity being *autonomous*, acting without intervention from other agents or a human; *social*, interacting and communicating with other agents; *reactive*, perceiving the environment and acting on changes concerning the agent's task; and *proactive*, taking the initiative to reach a goal. Most existing implementations (e.g., JADE (Bellifemine et al., 2003)) use a communication paradigm based on FIPA's agent communication language (FIPA, 2002), which is designed to exchange text messages, not complex data items. Thus we instead implement the blackboard paradigm which is capable of maintenance of complex items. Still, we acknowledge the above definition and the fact, that agents may concurrently work on a task and run in parallel. The processing units of our framework are hence implemented according to these MAS paradigms.

2.2 Related Systems

The following paragraphs discuss some of the most widespread robotics frameworks with respect to applicability for vision-based limp object handling and automation in the target scenario.

Orca[1] adopts a component-based software engineering approach without applying any additional architectural constraints. The framework is open-source, but uses the commercial Internet Communications Engine (ICE)[2] for (distributed) communication and definition of programming language independent interfaces. ICE provides a Java-based graphical user interaction/control instance, but capabilities to incrementally compose applications by connecting the executable nodes at runtime is very limited. One may start/stop nodes, but the communication is established using a publish/subscribe mechanism implicitly by implementing the corresponding interfaces. So while providing facilities for efficient distribution of tasks on multiple computers, Orca lacks the required flexibility when trying to automate tasks without time-consuming reprogramming (runtime reconfiguration).

The **Robot Operating System**[3] (ROS) is another open-source software framework striving to provide a (robot-)platform independent operating system. The framework uses peer-to-peer technology to connect multiple executables (nodes) at runtime in a publish/subscribe way. A master module is instantiated for the required node-lookup. An interesting feature of ROS is the logging and playback functionality (Quigley et al., 2009) allowing for replication of recorded data-streams for later usage, e.g., in a simulation environment. Still, the framework lacks a facility to record an application configuration in order to conveniently replicate an application setup for a specific task, e.g., load a pick- and place task description and execute it on various robotic setups. **Microsoft's Robotics Studio**[4], released in 2007, is a Microsoft Windows specific .NET-based software library for robotic software applications. The framwork utilizes the Coordination and Concurrency Runtime (CCR) and the Decentralized Software Services Protocol (DSSP) for message processing and passing. It includes a simulation environment and, most notably, a convenient facility to develop applications in a user-friendly graphical way. Still, it is closed source, and in addition to that, "not an ideal platform for real-time systems" (Jackson, 2007), since there is no guarantee, that a service is not interrupted.

[1] http://orca-robotics.sourceforge.net

[2] http://www.zeroc.com

[3] http://ros.sourceforge.net

[4] http://www.microsoft.com/Robotics

The open-source **Player/Stage**[5] project provides another popular and widespread framework for robotics. Here, a system typically comprises a "player", which implements a TCP-server on the robot, and "clients", the user control programs, such as a joystick controller or a keyboard device, that may connect to the server, send messages, and control the robot (Vaughan and Gerkey, 2007). Although being powerful, in the eye of the authors, Player/Stage is rather complex to setup and application development is not user-friendly. I.e., one needs to implement a character device and a interface/driver model atop the Player Abstract Device Interface (PADI), combine it with the Player Protocol and finally implement the application semantics, even for a simple pick- and place task. The flexible robotics and automation system introduced in this paper strives to overcome the drawbacks of the aforementioned systems. Furthermore, it incorporates the useful features of the above architectures to form a foundation for user-friendly limp object handling applications.

3 SYSTEM OVERVIEW

From a software engineering perspective, an application built atop of the proposed system is built from three building blocks shown in Figure 1.

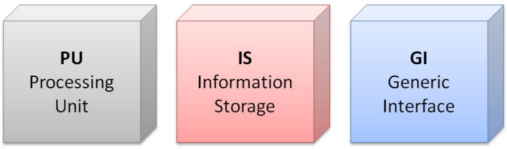


Figure 1: Building blocks of the flexible automation system.

These building blocks, namely the *information storage*, *processing units*, and *generic interfaces*, are introduced briefly in the following paragraphs.

3.1 Generic Interfaces (GI)

The first building block provides an easy-to-use interface abstraction for accessing external hardware components such as the realtime connectors for robot control, grippers and servos, or image sensors, user IO-devices (mouse, keyboard, etc.), or other sensory devices, e.g. force-torque sensors (FTS).

But implementing a generic interface is not limited to IO-devices, but indeed one can write an interface to virtually any external component, be it software or hardware. For instance considering software libraries, at the time of publication, the framework already provides predefined interfaces to the robust model-based realtime tracking library *OpenTL* (Panin et al., 2008) and the library underlying the efficient EET (exploring / exploiting tree) planner (Rickert et al., 2008) for advanced industrial robot control. Additionally, in order to support seamless integration with external applications, interfaces for accessing socket connectors for remote control, data exchange, and remote procedure calls are supplied by the framework, as well as an interface for running arbitrary executables.

3.2 Processing Units (PU)

The base class for processing data, the PU, provides a configuration, control and feedback facility and a possibility to share information with others. Furthermore, each processing unit is designed as a thread and supplies a description of the action it performs to the automation system (see Section 3.4).

Typically, an application comprises a set of PUs, where PUs perform their action in parallel, either continuously or they are triggered by a start event. While most hardware interfaces need a cyclic, continuous update / retrieval (such as the robot joint values or the camera interface), higher-level actions wrapped into a PU, e.g. moving the end-effector from A to B, handing a workpiece over to the next robot, or finding a grasping point in a visual scene, most commonly need a trigger-event in an assembly workflow.

3.3 Information Storage (IS)

In order to map a complex assembly workflow, exchanging data between PUs is essential. For example the input device unit maps to the target pose generation unit, which again maps to the joint values corresponding to a robotic end-effector.

The singleton information storage supplied by the framework is the building block designed for this purpose. Figure 2 shows the workflow for data registration, storage, and retrieval modalities as a diagram.

As shown in the figure, after registering a data-item, synchronous and asynchronous data access is possible, i.e. PUs can either listen for the event generated, when a data item has changed in the storage, or poll the data only when needed.

3.4 Task Automation

Tasks are generally described as a set of connected actions, where each processing unit defines a such action. In order to combine actions to a complex task,

[5] http://playerstage.sourceforge.net

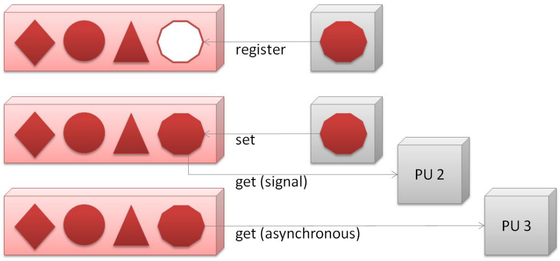


Figure 2: Dataflow for a data item in the information storage.

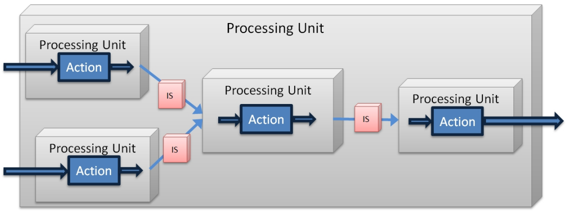


Figure 3: Processing units can be generated online to combine a set of arbitrary actions into a more complex one for automation.

a generic action description comprising an identifier, IO and configuration parameters, etc., was developed. Utilizing this action description, it is possible to specify a complex task by combination of primitive actions (e.g., operations like “grasp”, “screw”, “move-to”).

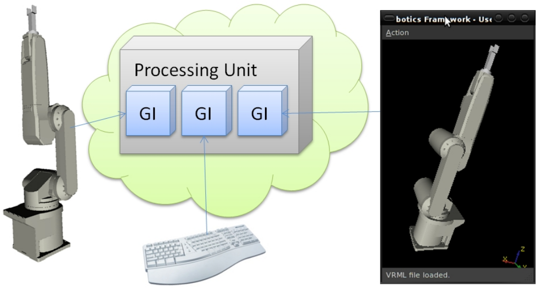
Furthermore, the framework supplies an automation unit, which enables a user to transform a task record into an action and generate a new processing unit (see Figure 3) from it. Increasingly complex tasks can be automated, as these composite actions are provided to the user interface for further combination. Thus it is easy to quickly and conveniently adapt, e.g., a production system to a novel product variant requiring different assembly steps.

4 TOWARDS LIMP OBJECT HANDLING

The following paragraphs show step-by-step, how the application for limp object handling is implemented using the framework’s building blocks.

4.1 A Simple Example

In a very basic, single threaded application example, the PU shown in Figure 4 is designed for processing data from and sending data to the robot and interaction with a PC keyboard.



VRML file loaded.

Figure 4: Application scheme for a simple keyboard controlled robot.

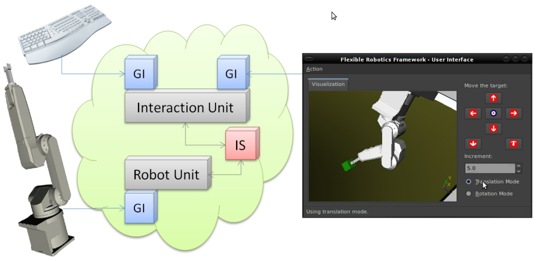
The actual hardware components are accessed utilizing the generic interfaces. The PU polls the robot interface and the keyboard interface continuously in an endless while-loop. Then, corresponding to the key pressed by the user, a robot action is computed and sent via the robot interface to the actual hardware. Finally, a widget provides for feedback visualization of robot movements. This widget is statically embedded in a Qt-based graphical user interface, which is updated cyclically by the PU.

Notably, no communication with other PUs is implemented here, so the application does not need to initialize an information storage.

Clearly, integration of all hardware interfaces and computations into a single PU is non-optimal considering load distribution and parallelization. Therefore, the next example shows, how this can be improved using multiple PUs and the information storage.

4.2 Multiple PUs and Information Sharing

Using the IS to share and exchange processing data enables the application developer to decompose a problem into semantically independent tasks and distribute them to multiple task-specific PUs.

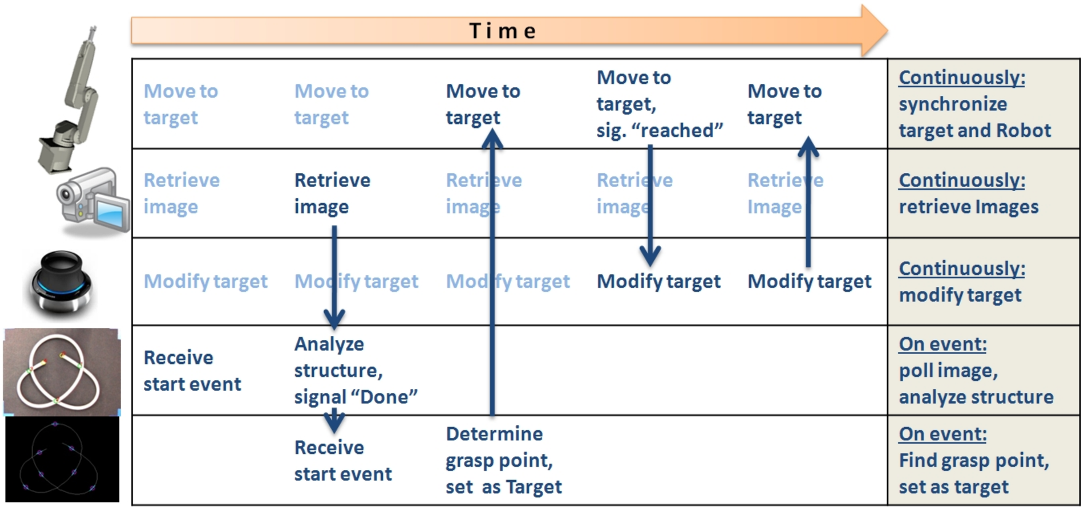


Increment: 5.0

Using translation mode.

Figure 5: Multithreaded application for keyboard control of a robot.

Figure 5 shows a control system, where keyboard / mouse events are handled by the GI of the upper PU,



Move to target	Move to target	Move to target	Move to target, sig. “reached”	Move to target	Continuously: synchronize target and Robot
Retrieve image	Retrieve image	Retrieve image	Retrieve image	Retrieve Image	Continuously: retrieve Images
Modify target	Modify target	Modify target	Modify target	Modify target	Continuously: modify target
Receive start event	Analyze structure, signal “Done”				On event: poll image, analyze structure
	Receive start event	Determine grasp point, set as Target			On event: Find grasp point, set as target

Figure 6: A workflow detail for automatically grasping a limp object from the table and handing it over at a pose defined through user-interaction.

also offering some configuration options in the interaction widget (shown in the right part of the screenshot). The PU in this case posts a data item containing the user’s selection for the new target position of the robot’s end-effector. The robot control unit implements a listener on that new target position. It computes a trajectory to move the robot to the target and sends the corresponding commands via the robot GI.

Note, that in the example the listener is not implemented event based (i.e., it does not implement a slot to the new target signal), but in an asynchronous manner. This ensures, that the robot smoothly completes a motion. Only after the movement has finished the next trajectory may thus be computed.

4.3 Limp Object Handling

Concluding the application section, the following paragraphs describe a rather complex limp object handling application. Here, all the proposed concepts from Section 3 are applied.

The application features a 6 DOF mouse-device PU and a realtime robot control unit sending new position commands every seventh millisecond and simultaneously sharing feedback information (joint angles and Cartesian pose of the end-effector). A separate GUI-unit provides for visual feedback, additional manual robot control through buttons and display of the live-image and (semi-)processed images from the IEEE1394 camera. In the application, a focus of attention (FoA) unit implements the attention condensation mechanism (Müller and Knoll, 2009). This approach speeds up visual processing by performing a relevance evaluation on the visual field as it creates regions of interest for salient areas appropriately.

A detail of the workflow for manipulation of a limp object in the workspace of the robot is scetched in Figure 6.

The figure shows three PUs performing continuous actions, the robot control unit, the camera unit and the mouse unit. Furthermore, two event-based units are shown, one for analyzing the object structure (ends and intersections) and one for determining a suitable point for grasping the object (see Figure 7 for a detail of the process).

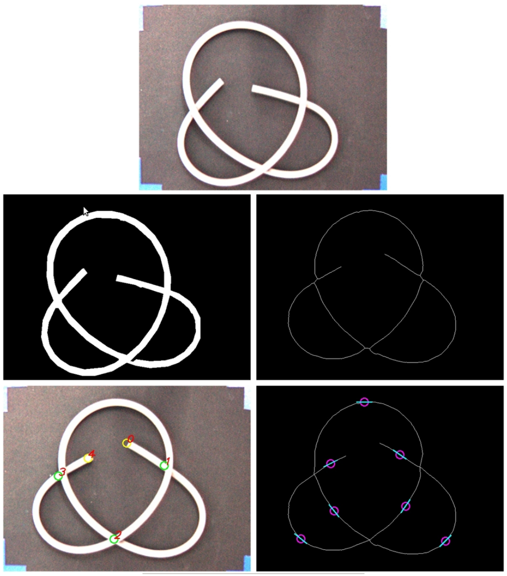


Figure 7: From top to bottom: The region of interest, background segmentation, thinning, structure analysis and possible grasping points.

The application comprises four more units, which are not shown. These are the FoA unit and the GUI unit mentioned before, and an event-based gripper unit. Using the forth one, the automation unit, the user is able to compose an arbitrary workflow (the application task), e.g., the one shown in Figure 6, by connecting the IO-parameters of different units and then transforming the task into a new PU, e.g., a "find-and-grasp-object" unit.

5 CONCLUSIONS

The paper introduces a flexible robotics and automation system for parallel, asynchronous and decoupled processing on hardware architectures like multicore or multiprocessor systems.

A typical modular application developed with the framework is an unordered set of instances of the building blocks. The set may comprise a structure of processing units; a blackboard for storage and exchange of data; and multiple interfaces to hardware or external software components. Furthermore, the proposed system incorporates facilities to generate automation functions, for example from telepresent teach-in or predefined action primitives.

As an example, the paper shows, how the robotics system can be used to develop an application for limp object handling, incorporating parallel visual processing and realtime control of the actuator in a user-friendly way.

ACKNOWLEDGEMENTS

This work is supported by the German Research Foundation (DFG) within the Collaborative Research Center SFB 453 on "High-Fidelity Telepresence and Teleaction".

REFERENCES

Anderson, J. R. (2007). *How Can the Human Mind Occur in the Physical Universe?* Oxford University Press.

Bellifemine, F., Caire, G., Poggi, A., and Rimassa, G. (2003). JADE: A White Paper. Technical report, Telecom Italia Lab and Universita degli Studi di Parma.

Corkill, D. D. (2003). Collaborating Software: Blackboard and Multi-Agent Systems & the Future. In *Proc. of the International Lisp Conference*.

Erman, L. D., Hayes-Roth, F., Lesser, V. R., and Reddy, D. R. (1980). The Hearsay-II Speech-Understanding System: Integrating Knowledge to Resolve Uncertainty. *Computing Surveys*, 12(2):213–253.

FIPA (2002). ACL Message Structure Specification. Technical report, Foundation for Intelligent Physical Agents.

Jackson, J. (2007). Microsoft Robotics Studio: A Technical Introduction. *IEEE Robotics and Automation Magazine*, 14(4):82–87.

Lehman, J. F., Laird, J., and Rosenbloom, P. (2006). A Gentle Introduction to Soar, an Architecture for Human Cognition. 14-Sept-2009, http://ai.eecs.umich.edu/soar/sitemaker/docs/misc/ /GentleIntroduction-2006.pdf.

Müller, T. and Knoll, A. (2009). Attention Driven Visual Processing for an Interactive Dialog Robot. In *Proc. of the 24th ACM Symposium on Applied Computing*.

Newell, A. (1994). *Unified Theories of Cognition*. Harvard University Press.

Panin, G., Lenz, C., Nair, S., Roth, E., in Wojtczyk, M., Friedlhuber, T., and Knoll, A. (2008). A Unifying Software Architecture for Model-based Visual Tracking. In *Proc. of the 20th Annual Symposium of Electronic Imaging*.

Quigley, M., Gerkey, B., Conley, K., Faust†, J., Foote, T., Leibs, J., Berger, E., Wheeler, R., and Ng, A. (2009). ROS: an open-source Robot Operating System. In *ICRA 2009 Workshop on Open Source Software in Robotics*.

Rickert, M., Brock, O., and Knoll, A. (2008). Balancing Exploration and Exploitation in Motion Planning. In *Proc. of the IEEE International Conference on Robotics and Automation*.

Vaughan, R. T. and Gerkey, B. P. (2007). *Software Engineering for Experminetal Robotics*, chapter Reusable Robot Code and the Player / Stage Project, pages 267–289. Springer tracts on Advanced Robotics. Springer.

Wooldridge, M. J. (2009). *An Introduction to MultiAgent Systems*. John Wiley & Sons, 2nd edition.

BIOMIMETIC CONTROL ALGORITHM FOR THE BALANCE AND LOCOMOTION OF WALKING SYSTEMS

Nicu George Bîzdoacă, Anca Petrişor
University of Craiova, Faculty of Automation, Computers and Electronics, Caiova, Romania
nicu@robotics.ucv.ro, apetrisor@em.ucv.ro

Hani Hamdan
SUPELEC, Department of Signal Processing and Electronic Systems, Gif-sur-Yvette, France
Hani.Hamdan@supelec.fr

Khalid Al Mutib
King Saud University, College of Computer and Information Sciences, Riyadh, Kingdom of Saudi Arabia
muteb@ksu.edu.sa

Keywords: Robotics, Biomimetic approach, Control strategy, Stable state transition.

Abstract: Implementation of active control has much potential to contribute to the creation and construction of innovative structures. This paper summarizes recent research of the authors that is the study of biomimetic control solutions regarding balance and locomotion of robotic systems. A first goal of the work consists in identifying solutions necessary to balance the individual systems. Research has been focused both on systems with a single foot, but also on biped, tripods, quadrupeds, hexapods and octopods. Static balance is achieved by a proper mechanical design (Bizdoaca and Petrisor, 2009), but also by a corresponding load / tensioning actuators systems that can compensate for inertial elements that can lead to system stability limit. Theoretical studies have been focused on developing an efficient stepping algorithm in environment with strong uncertainties, known as SSTA algorithm. The article present a series of experiments made with servo actuated and smart actuated (based on shape memory alloy, especially) walking biomimetic structures.

1 INTRODUCTION

Biomimetics (or bionics, biognosis, etc.) is an abstract of “good design from nature”. Roughly speaking, biomimetics is the concept of taking ideas from nature and implementing them in another technology.This concept is actually very old, for example, the Chinese wanted to make artificial silk 3,000 years ago. Some biomimetic processes have been in use for years. An example is the artificial synthesis of certain vitamins and antibiotics. More recently, the biomimetic concepts, ideas and applications are increasingly reported.

For example, the latest new biomimetic study reported in the journal Nature, according to the current picks of biomimetic issues, is actually from studying how ants avoid traffic jams, which has numerous implications for many scientists to rely on the behavior of ants or other natural systems to give them clues as to how to design computer systems that avoid overcrowded networks. Another biomimetic example, as commented by Philip Ball in the 26 February 2004 issue of Nature in (Hong, and Bruce, 2004), is on the use of microbes in waste-water that could make a handy household battery.

In a more general setting, according to (Whatis.com), biomimetic refers to human-made processes, substances, devices, or systems that imitate nature. The art and science of designing and building biomimetic apparatus is called biomimetics, and is of special interest to researchers in robotics, artificial intelligence (AI), nanotechnology, the medical industry, and the military. Other possible applications of biomimetics include nanorobot antibodies that seek and destroy disease-causing bacteria, artificial organs, artificial arms, legs, hands, and feet, and various electronic devices. One of the more intriguing ideas is the so-called biochip, a microprocessor that grows from a starter crystal in much the same way that a seed grows into a tree, or a fertilized egg grows into an embryo.

Biomimetics is now not at the stage in generating new concepts and ideas because the mother nature has already provided numerous models for us to imitate. The key is the implementation and development which is gathering momentum only recently because the science base can cope with the advanced techniques in various areas such as biology, materials, electronics, computing, communication and control etc. The idea of extrapolating designs from nature and copying them has entered into many areas of applied science, most notably the synthesis of new materials. So, it is no surprise that people tend to regard the biomimetics as an interdisciplinary field in materials science, engineering, and biology.

2 STUDY OF BIOMIMETIC CONTROL SOLUTIONS BASED ON THE BALANCE OF INDIVIDUAL SYSTEMS

Studies of the authors have been focused to identify the necessary solutions to ensure the individual systems balance. The research have been directed both on systems with a single foot, which made a movement by jumping, but also on biped, tripod, quadruped, hexapod and octopod systems.

Static balance is achieved by a proper mechanical design [CMSM'2009], but also by a corresponding load / tensioning of actuators systems that can compensate the inertial elements that can lead the system to the stability limit.

In addition to these studies for extension degrees of freedom in order to provide a more efficient navigation and balance control have been explored and studies smart damping and actuation based not only on servoactuators, but and to shape memory materials and magnetorheological fluids.

The essential condition for static equilibrium related to the position of the system gravity centre of which projection must be inside the contact surface described by the support elements (feet) of the biomimetic system is, for jumping and biped systems extremely difficult, while for ensuring a dynamic balance the use of specialized control architectures is required.

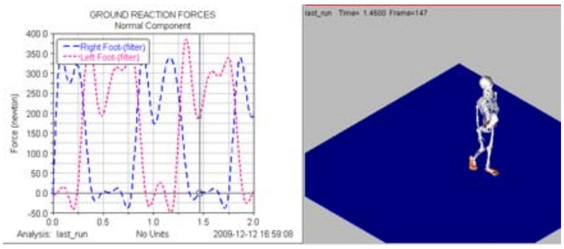


Figure 1: Study of biped locomotion.

Taking into account the human model, it is observed that it has a balanced structure in addition to appropriate state, and a sensorial system that provides the driving of the muscular system to compensate for factors that could lead to system stability limit.

Therefore, to achieve the balance of studied systems, it turned to a series of sensorial elements to ensure the system proximity identification of the stability limit, while an architectural simplification of the system to ensure absolutely necessary degrees of freedom for a dynamic compensation but much more limited than the biological model.

These considerations are related to the complexity of calculations necessary to some systems with more degrees of freedom, respectively considerations of ensuring the bionic system mobility, especially with a lower energy consumption.

Calibration bionic system proved to be a basic element, that essentially influenced subsequent locomotion of the mechatronic system.

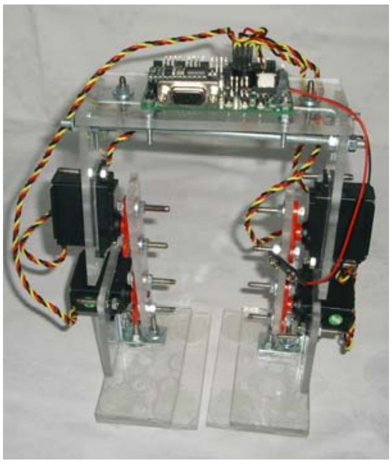


Figure 2: Biped biomimetic structure.

An important component of the study was focused on walking structures issues - quadruped, hexapod and octopod structures.

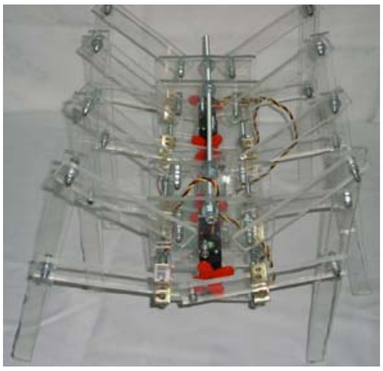


Figure 3: Octopod biomimetic structure.

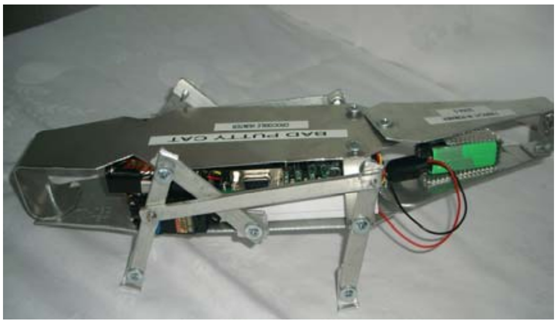


Figure 4: Quatrupede biomimetic structure.

Consulting literature revealed some aspects of the modalities and criteria for classification of walking robots structures, thus achieving a hierarchy in terms of biological models imitated, of physical morphology, of functions they performed and of locomotion strategy that they adopt.

In order to have a more complete image of the themes (walking robots field), we focused on:

- walking stability issues of such robots
- design advantages of such morphologies
- issues of wide range of areas where they prove their aplicability.

Conclusions resulted in this work led to the need for a dual approach: intelligent design and balancing of the mechanical system and implementing of a system/algorithm that allows the achievement of the instability system compensation or its approach to the stability limit, by the dynamic analysis of the specific conditions.

The developed algorithms should be simple, robust, adaptive, sub-optimal for reasons related to the response time.

3 STUDY OF BIOMIMETIC CONTROL SOLUTIONS BASED ON THE LOCOMOTION OF INDIVIDUAL SYSTEMS

Individual systems locomotion behave correctly identification of the control algorithm, but also, the identification of some kinematic and dynamic models, viable and efficient in implementing of these algorithms through command and control architecture.

Studies of systems moving by jumping with a single leg, with 4 feet, respectively hiperredundante systems, systems that have shape memory springs in their structure, have revealed the effectiveness of this type of solution for continuous monitoring of the damping coefficient necessary at the contact with the ground or for damping and driving of the mechatronic architecture (Bizdoaca, 2009).

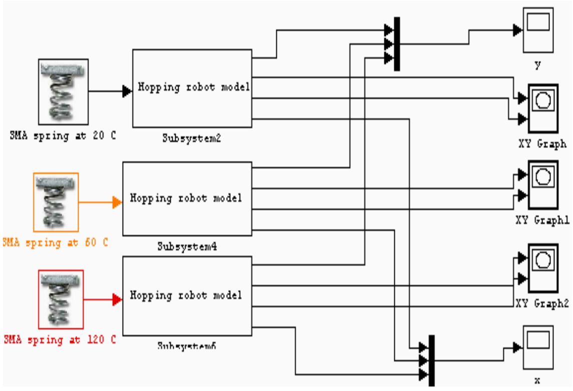


Figure 5: Mathematical model of a one-legged robot that uses as damping element a shape memory spring.

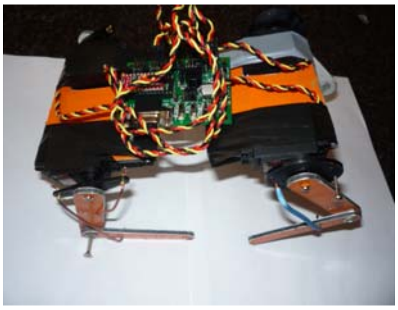


Figure 6: Biomimetic structure type rabbit.

Experimental measurements in conjunction with theoretical and practical work of the research team, have enabled studies to be efficient.

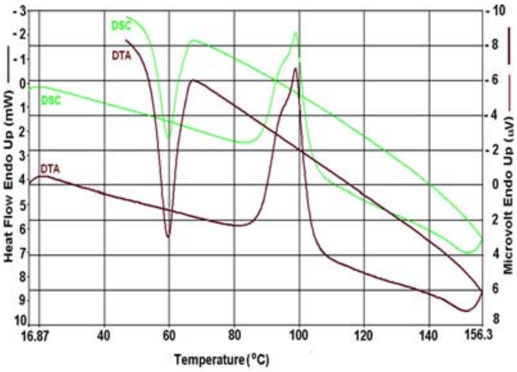


Figure 7: Biomimetic structure type frog. DTA and DSC curves for 18.275mg from SMA spring.

The effect of the energesing smart damping system is extremely efficient, numerical simulations showing this.

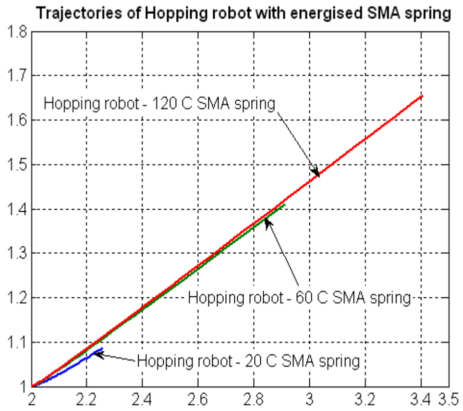


Figure 8: Hopping robot trajectory for different energizing temperatures of the SMA spring.

Particular attention was allocated for the hiperredundante systems both regarding creeping locomotion and handling structures as trunk type.

Kinematic study of walking systems was performed both on the robot body movement in space, given the motion laws of a rigid body in three-dimensional space and on the robot foot considering that it has a structure consisting of three links and three degrees of freedom in relation to the body, achieved by three rotational joints.

The rotation axes for the two joints which form the hip are set to intersect orthogonally.

In the case of existing a defect of the bionic architecture, locomotion was a complement of the activity of the project.

In robotic structures, the faults may be caused by external environmental conditions (operating environment) or by internal conditions (structure, sensors, actuators or control).

Faults monitoring system, according to its complexity, may adopt the following control strategies (from simple to complex): only detection and location of faults and possible suggesting of isolation actions and/or avoiding of fault components; further robot movement until bringing it into a safety state in a neighborhood of the end point for major faults; the robot stop in a safety state with maintaining of stability in the case of catastrophic faults.

General techniques for detection and identification of existing faults and those for recovery after failure, can be applied to walking robots systems, thus determining the operation space under fault. For this purpose were determined the operation areas of the walking robots feet as an annulus sector. Overcoming these allowed operation areas for each leg lead to interference problems.

To avoid this, were a prori eliminated all the areas that can be overcome, so that each leg has its own separate region.

Thus, it was defined for each leg one operation area (called operation cell) having a rectangular shape. Then it was analized, from kinematic point of view, the case of existing a fault due to a blocked joint of a robot leg. This analysis of the fault can be extended to other robot legs because of the robot symmetry.

Based on kinematic restrictions mentioned above, were presented and justified fault-tolerant locomotion algorithms for each considered case of fault both for the straight line robot moving and for a crab stepping type: robot locomotion tolerant at first joint blocking, robot locomotion tolerant at second joint blocking, robot locomotion tolerant at third joint blocking.

4 IMPLEMENTATION OF THE WALKING ROBOT CONTROL ALGORITHM IN SSTA STRATEGY

It is considered the walking robot structure as depicted in Fig.9, having three normal legs L^i, L^j, L^p and a head equivalent to another leg, L^0, containing the robot centre of gravity, G, placed in its foot. The robot body RB is characterized by two position vectors O^0, O^1 and the leg joining points denoted R^i, R^j, R^p. The joining point of the head, L^0, is the central point O^0, R^0= O0, so the robot body RB is univocally characterized by the set,

$$RB = \{O^0, O^1, \lambda^i, \lambda^j, \lambda^p, \lambda^0\}$$

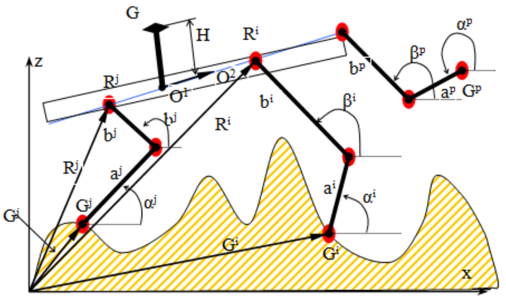


Figure 9: The geometrical structure of the robot.

A robot leg, let us consider of index i, has a body joining point R^i expressed by a complex number and the foot point denoted by the complex number G^i. It contains two joint segments defined by the lengths a^i, b^i with the angles α^i, β^i. Here there are considered three legs: i, j, p.

Due to the complexity of the evolution in an environment with strong uncertaines, an efficient walking algorithm which ensure stability and locomotion of the robotic structure, must be developped.

Stable States Transition Approach (SSTA) control strategy applied for a hexapod structure is proposed. By SSTA strategy is assured the walking robots evolution in uncertain environments subordinated to two goals:

- achievement of the desired trajectory expressed by the functions $O_z^0 = f(x)$ and $\theta = \theta(x)$, where x is the ground abscissa and $O_x^0 = x$; it is considered the evolution from left to right;

- assuarance of the system stability that is, in any moment of the evolution the centre of gravity has to be in the stability area.

Considering the walking robot as a variable causality dynamic system it is possible to realize this desideratum in different variants of assurance the steps succession. The steps succesion suposes a series of elementary actions that are accomplished only if the stability condition exists.

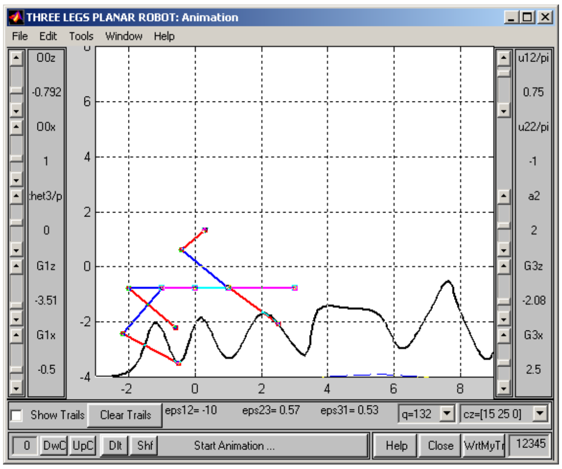
Continuously, by sensorial means or using the passive leg, the robot has informations about its capacity of evolving on the ground. Every time it is considered that the legs i, j are on the ground and the the system is stable ($\varepsilon_{ij} \in [0,1]$). The passive leg G^p is which realises the walking.

By testing the ground is realized its division in lots representing the fields on x axis which constitue the abscissas of some points that can be touched by the G^p leg. A next support point given by the free G^p leg, is chosen so that to existe a next stable state ε_{ip} or ε_{jp}, taking into account the actual state of legs activity.

A variant of mouvements succesion, composed by 12 steps, was proposed in (Petrisor, 2008).

5 EXPERIMENTAL RESULTS

An experimental platform, called RoPa, has been conceived. The RoPa platform is a complex of MATLAB programs for simulation and control of walking robots evolving in uncertain environments according to SSTA control strategy. A number of eight causality orderings of the robot structure have been implemented on RoPa.


THREE LEGS PLANAR ROBOT: Animation

O0z -0.792
O0x 1
:het3/p 0
G1z -3.51
G1x -0.5
u12/pi 0.75
u22/pi -1
a2 2
G3z -2.08
G3x 2.5
eps12= -10
eps23= 0.57
eps31= 0.53
q=132
cz=[15 25 0]
0
12345

Figure 10: RoPa Graphic User Interface.

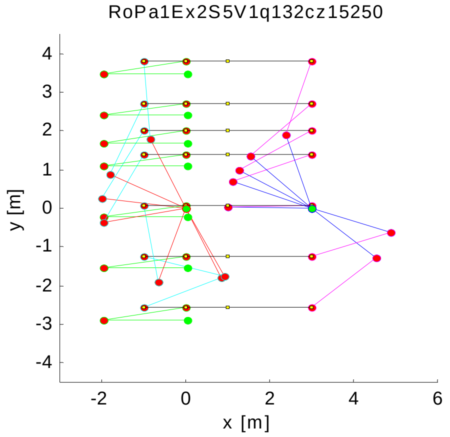


Figure 11: The robot kinematics evolution.

The stability of this evolution is graphical represented by a stability certificate of the evolution (Figure 12).

This certificate attests the stability index of the active pair of legs in any moment.

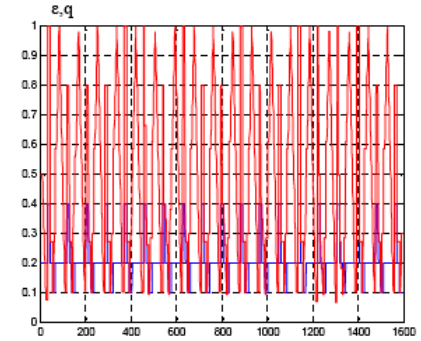


Figure 12: The stability certificate of the evolution.

6 CONCLUSIONS

Experiments on walking structures, made in the mechatronics laboratory, revealed the efficiency of SSTA algorithm, providing a robot system stability especially on hard terrain or with a high degree of uncertainty regarding the nature and topography of the contact surface. Using smart materials in the structure of biomimetic mechatronic architectures lead to an extension of the control capabilities. These latter elements will be explored in subsequent theoretical developments of research activity, practical experimentation, empirical, currently performed being extremely encouraging.

ACKNOWLEDGEMENTS

This report is part of the Romanian National University Research Council (CNCSIS) contribution to the project PNCDI – II - 289/2008.

REFERENCES

Bizdoaca, N. G, Tarnita, D., Petrisor, A., Diaconu, I., Bizdoaca, E.,2009. Control and design for biomimetic application using smart materials. *Annals of DAAAM for 2009 & Proceedings of the 20th International DAAAM Symposium, "Intelligent Manufacturing & Automation: Theory, Practice & Education*", Volume 20, Nr.1, ISBN 978-3-901509-70-4, ISSN 1726-9679, pp. 0949-0950, Published by DAAAM International, Vienna, Austria.

Bizdoaca, N. G, Tarnita, D., Petrisor, A., Bizdoaca, E.,2008. Modular Adaptive Bionics Structure, *WSEAS TRANSACTIONS on BIOLOGY and BIOMEDICINE*, Issue 9, Volume 5, ISSN: 1109-9518, pag 229-238

Centre for Biomimetics, "What is biomimetics, http:// www.rdg.ac.uk/Biomim/brochure.htm.

Cubero S., 2001. A 6-Legged Hybrid Walking and Wheeled Vehicle. 7-th International Conference on Mechatronics and Machine Vision in Practice, USA.

Fulgenzi, C., Spalanzani A., Laugier C., 2007. Dynamic Obstacle Avoidance in uncertain environment combining PVOs and Occupancy Grid, *Robotics and Automation*, 2007, pp.1610-1616.

Galt, S., Luk L., 1999. Intelligent walking gait generation for legged robots. *Proc. 2-th International Conference on Climbing and Walking Robots*.

Hong, L., Ramanathan, R., Bruce, L., 2004. Production of electricity during wastewater treatment using a single chamber microbial fuel cells, Environ. Sci. Technol., vol. 38, no. 7. 10.1021/es034923g, pp. 2281 – 2285.

Jung, S., Hsia, T.C., Bonitz, R., 2004. Force Tracking Impedance Control of Robot Manipulators Under Unknown Environment. *IEEE Transactions on Control Systems Technology,* 12(3), 474-483.

Kiguchi, K., Watanabe, K., Izumi, K., Fukuda, T., 2000. Application of Multiple Fuzzy-Neuro Force Controllers in an Unknown Environment Using Genetic Algorithms, *Proc. of IEEE International Conference on Robotics and Automation*.

Kiguchi, K., Miyaji, H., Watanabe, K., Izumi, K., Fukuda, T.,2000. Design of Neuro Force Controllers for General Environments Using Genetic Programming,' *Proc. of the Fourth Asian Fuzzy Systems Symposium (AFSS2000)*, pp. 668--673.

Nanayakkara, T., Watanabe K., Izumi, K., Kiguchi, K., 2001. Evolutionary Learning of a Fuzzy Behavior Based Controller for a Nonholonomic Mobile Robot in a Class of Dynamic Environments, *Journal of Intelligent and Robotic Systems*, Vol. 32, No. 3.

Petrisor, A., Bizdoaca, N., 2008. Stable States Transition Approach - a new strategy for walking robots control in uncertain environments. *ISI Proceedings of the 4th International Conference on Informatics in Control, Automation and Robotic*s, Madeira, Portugal.

Petrişor, A, Bîzdoacă, N., 2008. Implementation of the Stable States Transition Control Algorithm for a four free joints Walking Robot. *WSEAS Transactions on Systems Journal*, Issue 1, Volume 8, ISSN 1109-2777, pp.107-116.

Schmuacer U., Schneider A. Ihme T., 1996. Six Legged Robot for Service Operations. *Proc. of EOROBOT'96; IEEE Computer Society Press*, pp:135-142.

Thirion, B., Thiry, L., 2001. Concurent Programming for the Control of Hexapod Walking. *7-th International Conference on Mechatronics and Machine Vision in Practice,* USA.

ROBUST 6D POSE DETERMINATION IN COMPLEX ENVIRONMENTS FOR ONE HUNDRED CLASSES

Thilo Grundmann, Robert Eidenberger, Martin Schneider and Michael Fiegert
Siemens AG, Corporate Technology, Autonomous Systems, Munich, Germany
{*thilo.grundmann.ext, robert.eidenberger.ext, martin.schneider, michael.fiegert*}*@siemens.com*

Keywords: Object recognition, 6d Pose estimation, Multi-object scenarios, SIFT, Large database, One-hundred classes.

Abstract: For many robotic applications including service robotics robust object classification and 6d object pose determination are of substantial importance. This paper presents an object recognition methodology which is capable of complex multi-object scenes. It handles partial occlusions and deals with large sets of different and alike objects.

The object recognition process uses local interest points from the SIFT algorithm as features for object classification. From stereo images spatial information is gained and 6d poses are calculated. All reference data is extracted in an off-line model generation process from large training data sets of a total of 100 different household items. In the recognition phase these objects are robustly identified in sensor measurements.

The proposed work is integrated into an autonomous service robot. In various experiments the recognition quality is evaluated and the position accuracy is determined by comparison to ground truth data.

1 INTRODUCTION

Object recognition comprises the tasks of object class identification and pose determination. Although this plays a major role in several scientific domains, many current approaches to object recognition are limited in their applicability due to inaccuracies in the pose determination, the number of detectable objects or the complexity of scenes. Object occlusions and the appearance of similar objects in the same image often constitute problems to robust detection.

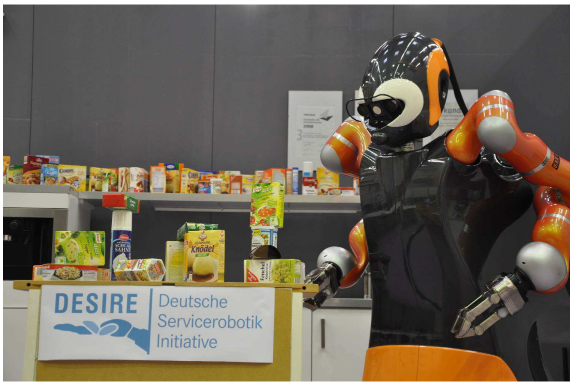


Figure 1: A service robot commonly operates in a complex environment.

In this work we present a detection methodology which allows accurate 6d object pose determination and object class detection. It uses Lowe's SIFT algorithm (Lowe, 1999) for determining local, scale-invariant features in images. The application of the SIFT algorithm on stereo images from a camera pair enables the calculation of precise object poses. This appearance- and model-based approach consists of two separate stages, *Model generation* and *Object recognition and pose determination*.

Model generation is an off-line process, where the object database is established by determining essential information from training data. We consider a set of 100 household items of different or alike appearance. The challenges lie in the reasonable acquisition and efficient processing and storing of large data sets.

Object recognition and pose determination aims on satisfactory object classification results, low misclassification rates and fast processing. The precise detection of the object pose allows the accurate positional representation of objects in a common reference frame which is important to many tasks. The 6d pose is described by 3 translational and 3 rotational components and is formulated in continuous domains.

The proposed method is embedded in a mobile service robotics framework. Figure 1 shows the prototype. The robust and accurate object recognition system is basis to continuing works such as object manipulation (Xue et al., 2007), perception planning (Eidenberger et al., 2009) and physical object dependency analysis (Grundmann et al., 2008), which point out its importance.

Section 2 outlines current state of the art approaches to model generation and model-based object recognition.

In Section 3 the data acquisition procedure and the methodology for model generation is described. Section 4 depicts the principles for object class identification and 6d pose determination.

This paper closes with experiments in Section 5 which demonstrate the proposed theoretical concepts on real data.

2 RELATED WORK

The problem of visual object pose estimation has been studied in the context of different fields of application, robotics or augmented reality to name two.

All approaches can be characterized by a number of parameters like the number of dimensions of the measured pose (mostly one, three or six), the restrictions on the objects to be localized (e.g. in shape, color or texture), the ability to estimate the pose of multiple instances of the same class in one image and the capacity for various different classes in the model database of the proposed system. Also the proposed sensor system (mono/stereo camera, resolution), process runtime and the achieved precision in the pose are characteristic.

One of the earlier papers on object pose estimation (Nayar et al., 1996) dealt already with the high number of one hundred different classes. Nevertheless the measured pose in this case is only one-dimensional, and the scene is assumed to have a black background and no occlusion or objects of unknown classes in it.

Later on (Zhang et al., 1999) and (Walter and Arnrich, 2000) extended the estimated pose to 3DOF assuming the objects to be located upright on a table, using one of five objects in their experiments.

Others (Kragic et al., 2001) restricted their objects to forms that can be modeled by wire frames and achieved good results in full 6d pose estimation examining two different objects. The quality of the pose is indirectly evaluated by successful grasping.

True 6d pose measurement using local SIFT features was demonstrated by (Lowe, 1999), describing a method that is able to localize flat objects within a range of 20 degrees, demonstrated on two scenes, consisting of three different objects each, without evaluation of the pose accuracy.

Azad et. al. presented a stereo camera based method (Azad et al., 2007) for full 6 DOF pose retrieval of textured object using classic SIFT interest points. The method requires the objects to posses flat surfaces for the stereo recognition, and the accuracy of the pose estimation is not measured directly.

Recently Collet et al. presented a mono camera object localization system (Collet et al., 2009), based on SIFT features using ransac and mean shift clustering. They also describe an almost fully automatic process for the model generation and give some experiments with four object classes where measured poses are evaluated against ground truth. The error is described by histograms over the translational and rotational error.

Another model generation system is described in (Pan et al., 2009) which is able to build a model online from a video stream. The model is also built fully automatically, additionally the user is assisted in the image collection by guiding the direction to which the object should be moved and by the visual feedback of the emerging model.

A comprehensive taxonomic and quantitative comparison of multi-view stereo model building methods for dense models can be found in (Seitz et al., 2006), sparse models are out of scope for this paper though.

Our approach is comparable to the method of (Collet et al., 2009), using a similar 3d model which is not restricted in the shape based on SIFT features. Through the use of stereo vision and thus conceptual different methods for the pose determination a higher accuracy in pose measurement is achieved.

Note, that an exact direct comparison is difficult without perfect re-implementation, so we generated translational and rotational histograms similar to the ones in (Collet et al., 2009) which demonstrate a better performance of our approach. But when it comes to comparison, the main drawback to the research done on six dimensional object localization is the absence of benchmark datasets.

3 MODEL GENERATION

Model generation aims on the acquisition of training data and its processing to generate object class models. It is essential to filter significant data and efficiently store it to enable reasonable processing times.

The KIT object modeling center IOMOS (Xue et al., 2009) is shown in Figure 2 with its rotary disk and the rotating bracket, where the sensors are mounted on. A stereo camera system and a laser scanner are used to acquire 360 stereo images for each object and the 3d surface. Rotating the object on the

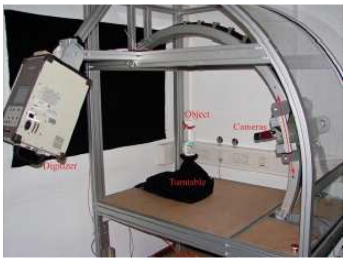


Figure 2: Object modeling center.

turntable and moving the sensors results in sensor data at 360° latitudinal and 90° longitudinal angles in 10° steps.

The build process starts with computing the SIFT interest points (IPs) for each image and calculates 3d points by triangulation of IPs in each stereo image. Then matches over all images are determined and equivalence classes from these matching IPs found. At last each equivalence class is represented by one descriptor and one 3d location. All these equivalence class representatives together build up the model.

Now, each step of the build process is described in detail.

1. SIFT Interest Points Calculation. The base for the process are the SIFT interest points for each image. Each SIFT interest point $s^i = (u,v,s,o,d_l)$ consists of the 2d location (u,v) in the image, its scale s and its orientation o. d_l denotes the l-dimensional descriptor vector. For each training image the set of interest points in the left camera image S_l with $S_l = \{s^1,...s^n\}$ is determined. The interest point set S_r for the right camera image is acquired respectively.

2. Triangulation. 3d points are computed for corresponding IP's in the left and right images. Corresponding IP's are determined by using the epipolar constraint and SIFT descriptor matching. For every IP in a left image V_l we compute the epipolar line L_e in the right image V_r and determine the subset $S_r^e \subset \{s^i \in S_r | dist(s^i, L_e) < \varepsilon_E\}$ with ε_E as the maximum epipolar distance. Then, SIFT descriptor matching is performed. It is important to do the epipolar examination before the SIFT matching step. This way the set of possible matches is constrained to a region in the image. IPs with similar descriptors from other parts are not considered and cannot distort the result. For each matched IP pair (s_l^i, s_r^j) the corresponding 3d location and orientation are computed using triangulation (Hartley and Zisserman, 2004) and transformed into the objects coordinate system to get the spatial feature representation:

$$s_{\#} = (x,y,z,x',y',z',s,o,d_l). \tag{1}$$

The first three elements x,y and z denote the translational coordinates, x',y' and z' represent the direction from where the interest point is visible. Scale s, orientation o and descriptor d_l are equal to the parameters of the 2d interest point.

3. Equivalence Relation. The next step aims at partitioning this set of 3d points into subsets originating from the same physical point. This equivalence relation is seeded from IPs with fitting appearance and location and is completed by the transitive closure.

First candidates are found by appearance. The enormous amount of data with on average 850 3d locations for each training view, 280000 for each object, can be handled efficiently by means of a kd-tree. Using Euclidean distance in the SIFT descriptor space the nearest n_N- neighbors in the kd-tree are searched for each $s_{\#}^i$.

n_N depends on the sampling rate of the training data as SIFT descriptors are only invariant within a limited angular range and each face of the object is only seen from a certain range of camera positions. It was set to 150 in our case.

Then candidates are checked for their spacial fit by projecting them to the view of their matching partner both ways and calculating the Euclidean distance in the image. If this distance, which is the expected reprojection error, exceeds a threshold (5 pixel in our system) the candidate is rejected. In rare cases it can happen that two IP's from the same view are in one equivalence class. These IP's are removed.

A standard connected component algorithm is used to compute the transitive closure.

4. Subdivision and Representatives. We now seek a simple representation for each equivalence class above a minimal size v_{min}. Very small classes (eg. v_{min}=4) are discarded to suppress IPs of low value for the recognition process and noise.

When evaluating classes one finds the locations clustering well, but descriptors to spread considerably. This is not surprising since we do see most points from a wide angle range were the SIFT descriptor cannot be assumed to be invariant. Instead of more complex density models we favor a very simple representative, which is a simple mean for location and normalized mean for descriptor. To make this simple model suitable we sacrifice the simple relation, where one class corresponds to one physical point, and split classes with k-means until they can be represented as spheres. This considerably simplifies and speeds up the recognition process.

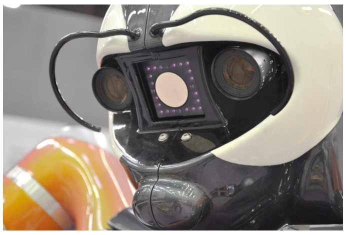


Figure 3: Sensor head: Stereo cameras and 3d-TOF camera mounted on a pan/tilt unit.

The full model for one object needs about 5% of storage of the initial SIFT features. This enables fast recognition since big databases can be held in RAM completely.

4 OBJECT RECOGNITION AND POSE DETERMINATION

Many applications require a high pose determination accuracy. To account for this we use a stereo camera based approach, as preceding experiments using a mono camera and the posit algorithm (Dementhon and Davis, 1995) lead to unsatisfying results. Comparable results were found in (Azad et al., 2009) which showed that stereo approaches outperform mono camera approaches by the factor of ≈ 2.

Our stereo setup on the robot (Figure3) consists of two AVT Pike F-145C firewire cameras with a resolution of 1388 x 1038 pixels each, equipped with 8.5mm objectives and mounted with a disparity of $0.12m$.

Precise intrinsic and stereo calibration of the cameras is essential to our algorithms so they were carried out with the Camera Calibration Toolbox for MATLAB, using about 60 stereo image pairs of a custom made highly planar checkers calibration pattern.

The recognition and localization process consists of the following steps:

1. Calculate SIFT Interest Points. For each of the stereo images, V_l and V_r a corresponding set of interest points is calculated $S_l = \{s^1, ..., s^n\}$ and $S_r = \{s^1, ..., s^m\}$.

2. Find Correspondence to Object Models (Figure4). For all elements $s^i \in S_l$ try to find up to p_{mm} multiple matches $c^k = \{i, j\}$ with a 3d feature from the model database $s^j_\# \in M$. The criterion for a match is that the Euclidean distance in descriptor space is below an absolute threshold $p_{tm} = 0.3$. The multiple match is needed, because with an increasing number of object models in the database, the uniqueness of a lot of features is lost. This happens in a non ignorable way for household items that often follow a corporate design and share large areas of texture.

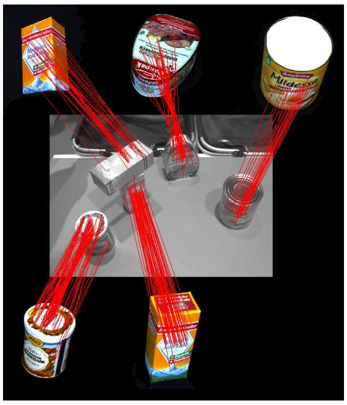


Figure 4: Recognition principle: finding matches between interest points $s_\# \in M$ from given models and interest points detected in one image $s \in S_l$ (step 2).

To speed up the search for matches, the descriptors from the database M are structured in a kd-tree using the ANN library. To increase the performance, the nearest neighbor search is approximated. The quality of the approximation can be parameterized, we used the value p_{app} = 5.

3. Construct Stereo Interest Points. For all matchings $c^k = \{i, j\}$, we try to find multi stereo matches on the right interest point set S_r. The epipolar constraint is used in the same manner as described in Section 3.2, but after the epipolar spatial restriction, a relative multi match is used. This has to be done to account for the classic situation where multiple instances of the same object class are placed side by side on a board, leading to multiple similar features on the epipolar line L_e. After this procedure we obtain a set of l 3d SIFT points $S_\# = \left\{s^1_\#, ..., s^l_\#\right\}$ by triangulation. Each of these 3d SIFT points belongs to an object class, indicated by its corresponding database feature $s^j_\#$. In a system with t known classes this gives a partition of $S_\# = \left\{S^1_\#, ..., S^t_\#\right\}$, whereas some of these class depending partitions might be empty. From here on, all calculation is done separately for each object class.

4. Cluster within Class t. To account for scenes with large numbers of identical objects and to deal with the high number of erroneous 3d SIFT features, we construct initial pose estimates $P = \left\{p^1, ..., p^x\right\}$ from $S^t_\#$. This is done by choosing randomly non collinear triplets of 3d interest points from $S^t_\#$ and check whether their mutual Euclidean distances match those in the model database.

Within the 6d space of the initial pose estimates P, qt-clustering (Heyer et al., 1999) is performed to find consistent 6d pose estimates. The clusters consist of 6d poses p^x which all correspond to a triplet of 3d interest point $S^t_\#$. That way each cluster describes a set of 3d interest points $S^c_\# \subset S^t_\#$.

5. Pose Determination. All 3d interest points from each cluster $S^c_\#$ are used to determine the transformation of the database 3d IPs into the measured cloud by a least squares pose fit (Figure 4). In the resulting list of object classes and poses a similarity search is performed to eliminate duplicate poses, which emerge from imperfect clustering.

One important advantage of this 3d SIFT stereo approach is, that in contrast to other approaches it is able to handle objects of any shape, as long as the object fulfills the requirements on its texture that are inherent to the SIFT algorithm. It is not generally bound to the usage of SIFT features, so in the future improved interest point methods could replace the SIFT interest points. Note, that the parameters in the algorithm strongly influence the performance and thus have to be optimized to fit an application scenario.

5 EXPERIMENTS

The experiments we conducted should evaluate the performance of our system in terms of detection rate and pose accuracy. Since it is very difficult to acquire pose ground truth data in complex scenarios we split the evaluation into two parts. In the first part, we set up scenes where the ground truth pose can be determined with a calibration pattern, leading to single object scenes of low complexity. The detection rate in these simple settings was 100%.

In order to evaluate our method in more realistic scenarios, we set up a second test, based on complex scenes consisting of up to 30 objects placed in arbitrary positions in a real environment supplemented with unknown objects. This way the method's robustness against occlusion, object ambiguities and environmental influences can be evaluated. In such scenarios, ground truth poses are not available, but it is possible to evaluate the correctness of the detection by visual inspection of the bounding volumes that are projected into the images.

The time required for a one shot recognition of a complete scene can be divided up into two parts: The SIFT calculation takes about 0.6 seconds per image on a 2GHz Intel multicore and the processing of the poses takes 0.3 seconds without using the multi cores. Note that there are some parameters like the kd-tree approximation quality which strongly influence the runtime of the processing part.

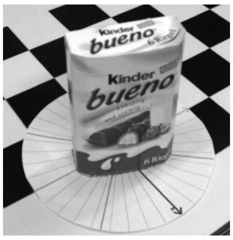


Figure 5: Object with calibration sheet.

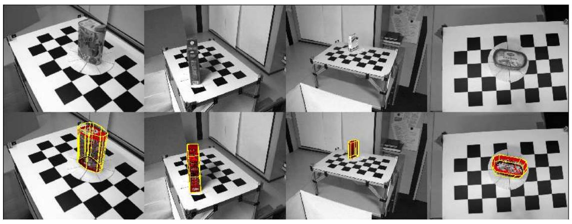


Figure 6: 1st row: objects, placed precisely onto the calibration pattern. 2nd row: Ground truth values projected into image (dot cloud(red), bounding volume(yellow)).

5.1 Evaluation of Pose Accuracy

To evaluate the system's accuracy in the pose estimation, we compared the recognition results of 260 scenes with ground truth. To acquire 6d ground truth data we applied paper sheets with the projected corresponding model onto the bottom of the objects (Figure5). Since the origin of the objects is located in the bottom plane, this method enables us to place the object onto a calibration pattern with 3DOF and with sub-millimeter accuracy.

In the test we placed one object per scene onto a calibration pattern (Figure6) and measured its pose. Using the camera's pose with reference to the calibration pattern which is determined using the Matlab calibration toolbox, we transformed the ground truth pose into the camera coordinate frame where we compared it against the result of the recognition algorithm. We selected seven different objects with varying shapes for this experiment.

The distance of the objects origin and the camera ranged from 0.42m to 1.25m (Figure 7/1). This area was chosen from our experiences with the working area of the mobile service robot.

Due to the relatively simple nature of the scenes all objects were detected correctly, so we collected 260 pose measurement error vectors. As expected

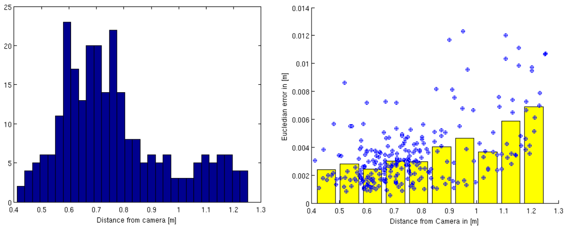


Figure 7: 1st: Histogram of object's distance to camera over the test set. 2nd: Translational error over distance (dots) and the mean translational error over distance (bars).

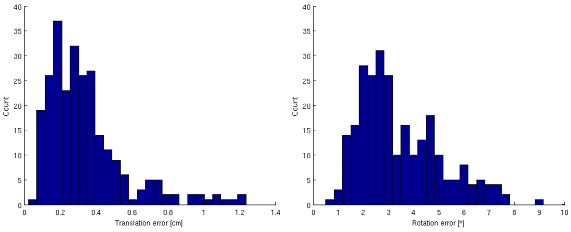


Figure 8: Histograms of translational(in cm) and rotational error(in ○).

from the measurement principle, the results show increasing translational errors with growing distance (Figure7/2). For the evaluation of the rotational errors, we calculated the minimal rotation angle from the resulting 6d pose to the ground truth 6d pose. The overall distribution of translational and rotational errors are shown in Figure 8, the corresponding standard deviation of the translational error is shown in Table 1.

To get a more expressive model of this measurement process, we transformed the error into the camera frame, and calculated the covariance for the translational components of the measurement errors (Figure9). As expected from the measurement principle, the standard deviation of the z_c-component in the camera frame is considerably higher as the standard deviations in the other directions x_c and y_c (Tab.1). Using this model it is possible to model detection inaccuracies fairly precisely. Note, that the inaccuracies also contain potential errors in the 3d models as well as the bias from placing the object imprecisely onto calibration pattern.

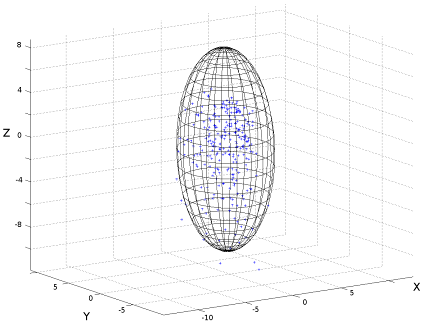


Figure 9: The xyz-deviations of all test scenes, depicted in the camera frame, with its covariance ellipsoid (95% quantile).

Table 1: Translational error in the camera frame.

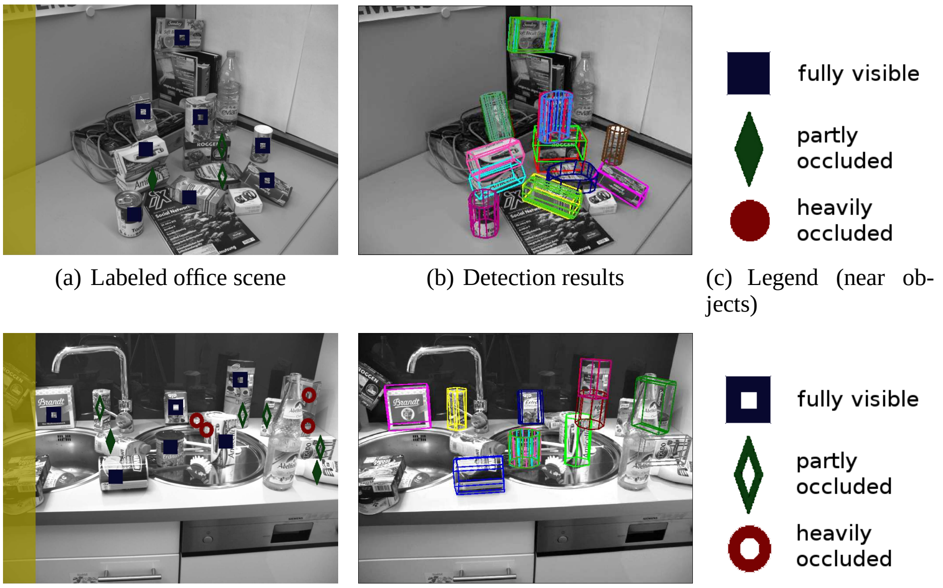
stddev	[mm]
$\|(x_c y_c z_c)\|_2$	2.2436
x_c	1.3763
y_c	1.3617
z_c	3.4235

5.2 Object Recognition in Complex Environments

In this experiment complex and cluttered scenes are considered. A series of 60 images of different scenarios at different locations are investigated. All contain chaotic object arrangements including trained and unknown, random objects. For the evaluation, all objects are categorized according to their distance from the camera and to their occlusions. Thus, we differentiate between close and far items and cluster the objects into fully visible, partly occluded and heavily occluded ones.

Figures 10(a) and 10(d) show two scenes with the initially labeled objects. Squares mark fully visible objects, diamonds indicate partly occluded items and circles denote heavily occluded ones. Close objects are shown solid, far away objects as halves. The yellow area on the left of these images is not considered for the stereo matching, so objects in this part of the image are ignored for the recognition. The respective detection and classification results for the two scenes are illustrated in Figures 10(b) and 10(e). The bounding volumes with respect to the 6d poses of the recognized objects are projected into the image. In the first sample, all objects are detected. However, a small number of false positives can be recognized for the salt boxes, the rye bread and the oil can. False positives are objects that are detected although they are not physically there. They result from ambiguous objects with similar textures on several sides or similar objects in the database, such as the three almost identical salts. In Figure 10(e) not all objects are recognized because of heavier occlusions. Though, fully visible ones are clearly identified. The successful detection also depends on the lighting as the trained images were acquired under one specific lighting condition. Variations influence the recognition process.

By visual inspection the detection results are com-



(a) Labeled office scene (b) Detection results (c) Legend (near objects)

(d) Labeled kitchen scene (e) Detection results (f) Legend (far objects)

Figure 10: One shot detection results for two complex scenes in a kitchen and an office environment. The results are evaluated with respect to the a priori categorized objects.

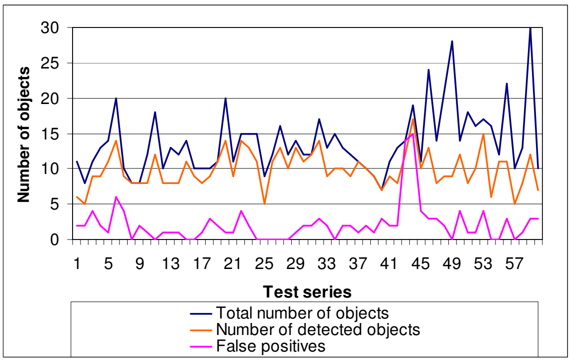


Figure 11: Object detection results and false positives in comparison to the total number of objects in scene for 60 different scenarios with a total of 826 objects.

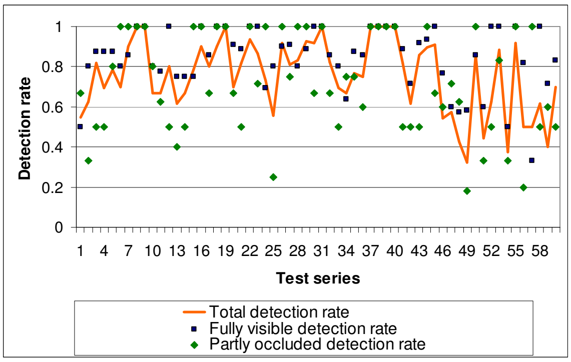


Figure 12: Overall detection rate and detection rates of fully visible and partially occluded objects.

pared to the pre-selection. The true positive and false positive classifications rates are manually determined by checking the class and pose recognition results. Figure 11 shows the detection results of each of the 60 scenarios. The total number of objects in the scenes ranges from 7 to 30. The number of recognized objects and the false positives for each scene are also plotted. While the number of false positives is generally low, a few scenes with very dense object arrangements showed more than 10 false object hypotheses.

In Figure 12 the detection rates are plotted. The overall rates show the number of detected objects over the total objects in the scene. The other curves depict the rate of detecting fully visible objects and partly occluded ones. The total recognition results over all scenes are listed in Table 2. Rates for each category a separately and jointly determined. As expected, closer objects perform better in the recognition. In matters of occlusions, a decay can be recognized from fully visible to heavily occluded items. The overall detection rate of 72 per cent is lower than the peak

Table 2: Object class detection results broken down to the detection categories.

	detection rates			
distance	fully visible	partly occluded	heavily occluded	total
near	0.863	0.763	0.179	0.796
far	0.813	0.593	0.167	0.617
total	0.846	0.684	0.170	0.722

detection rate of 86 per cent for close and fully visible objects. The random object alignments with unfavorable object poses, lighting influences and object occlusions are reasons for recognition failures. However, considering the large database and the complexity of the scenes the one shot recognition results are promising.

6 CONCLUSIONS

We presented a system that is able to detect and localize objects from up to 100 different classes. The 6*d* detection accuracy of the object pose and the detection rate are evaluated in extensive experiments, which demonstrated a true positive detection rate of 72% in highly complex cluttered multi object scenes with partly occlusions. The resulting pose errors had a standard deviation of 3.4*mm* in the direction of the camera (z_c) and 1.4*mm* in x_c and y_c.

A satisfactory trade-off is found between fast processing and good recognition rates and detection errors and failure recognitions. The system is suitable to applications in cluttered environments with random object alignments and unknown objects.

In future works, we plan to include sparse bundle adjustment into the model generation process to increase the precision in the 3*d* models which is expected to increase the pose precision on the one hand, but also to loosen the precision requirements on the camera pose.

ACKNOWLEDGEMENTS

This work has partly been supported by the German Federal Ministry of Education and Research (BMBF) under grant no. 01IME01D, DESIRE.

REFERENCES

Azad, P., Asfour, T., and Dillmann, R. (2007). Stereo-based 6d object localization for grasping with humanoid robot systems. In *IEEE IROS 2007*.

Azad, P., Asfour, T., and Dillmann, R. (2009). Stereo-based vs. monocular 6-dof pose estimation using point features: A quantitative comparison. In *Autonome Mobile Systeme 2009*, Informatik aktuell. Springer.

Collet, A., Berenson, D., Srinivasa, S., and Ferguson, D. (2009). Object recognition and full pose registration from a single image for robotic manipulation. In *IEEE ICRA 09*.

Dementhon, D. F. and Davis, L. S. (1995). Model-based object pose in 25 lines of code. *International Journal of Computer Vision, Springer Netherlands*, Volume 15.

Eidenberger, R., Grundmann, T., and Zoellner, R. (2009). Probabilistic action planning for active scene modeling in continuous high-dimensional domains. *IEEE ICRA 2009*.

Grundmann, T., Eidenberger, R., and Zoellner, R. (2008). Local dependency analysis in probabilistic scene estimation. In *ISMA 2008. 5th International Symposium on Mechatronics and Its Applications*.

Hartley, R. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press.

Heyer, L. J., Kruglyak, S., and Yooseph, S. (1999). Exploring expression data: Identification and analysis of coexpressed genes. *Genome Res.*, 9.

Kragic, D., Miller, A. T., and Allen, P. K. (2001). Real-time tracking meets online grasp planning. In *IEEE ICRA 2001*, Seoul, Republic of Korea.

Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *International Conference on Computer Vision*, pages 1150–1157, Corfu, Greece.

Nayar, S., Nene, S., and Murase, H. (1996). Real-time 100 object recognition system. In *IEEE ICRA 1996*.

Pan, Q., Reitmayr, G., and Drummond, T. (2009). ProFORMA: Probabilistic Feature-based On-line Rapid Model Acquisition. In *Proc. 20th British Machine Vision Conference (BMVC)*, London.

Seitz, S. M., Curless, B., Diebel, J., Scharstein, D., and Szeliski, R. (2006). A comparison and evaluation of multi-view stereo reconstruction algorithms. In *IEEE CVPR 2006*.

Walter, J. A. and Arnrich, B. (2000). Gabor filters for object localization and robot grasping. In *ICPR 2000*.

Xue, Z., Kasper, A., Zoellner, J., and Dillmann, R. (2009). An automatic grasp planning system for service robots. In *14th International Conference on Advanced Robotics (ICAR)*.

Xue, Z., Marius Zoellner, J., and Dillmann, R. (2007). Grasp planning: Find the contact points. In *IEEE Robio 2007*.

Zhang, J., Schmidt, R., and Knoll, A. (1999). Appearance-based visual learning in a neuro-fuzzy model for fine-positioning of manipulators. In *IEEE ICRA 1999*.

MINDLAB, A WEB-ACCESSIBLE LABORATORY FOR ADAPTIVE E-EDUCATIONAL ROBOT TELEOPERATION

P. Di Giamberardino, M. Spanò Cuomo and M. Temperini
Department of Computer and System Sciences, University of Rome ”La Sapienza”, Rome, Italy
{*digiamberardino, marte*}*@dis.uniroma1.it, m.spano.cuomo@gmail.com*

Keywords: WebLab, e-learning, Robot control, Teleoperation, Lego Mindstorms.

Abstract: The paper deals with a web based remote access laboratory, called MindLab, for educational support, via e-teaching and e-learning approaches, in practical robotic experiment to students of robotics and control theory courses. The hardware components are based on commercial easy to use, cheap and modular Lego Mindstorms devices. The software part, in addiction to the framework required for web access to the hardware resources, is designed and implemented in order to allow the teacher to define particular didactic paths depending on each student's skills and on the desired final abilities to be acquired. Exercises are proposed following a skill/competence level set, enabling higher levels access according to the results obtained. After a short description of the hardware setup, the software functionalities, mainly focusing on the exercise definition and presentation aspects, are illustrated, giving also an example of a standard exercise. Concluding remarks end the paper.

1 INTRODUCTION

A crucial aspect in both face to face and distance learning is in the need for direct hands-on learning experiences, conducted by the students in laboratory (Traylor R.L., Heer D. and Fiez T.S., 2003; Gomes L. and Bogosyan S., 2009).

Unfortunately, there are severe limitations in the availability of sufficiently large and equipped laboratories to cover an access reasonably open to all the needing students.

A remote environment would allow the students to operate their experiments in distance, by managing their queue and supporting the learning experience. The effect is a more efficient sharing of the available resources and a potentially full time availability. This advantage is often stressed in several applications (Gomes L. and Bogosyan S., 2009; López D., Cedazo R., Sánchez F.M. and Sebastian J.M., 2009).

Implementing the networked solution by a web-based application does have its merits (Gomes L. and Bogosyan S., 2009; López D., Cedazo R., Sánchez F.M. and Sebastian J.M., 2009; Balestrino A., Caiti A. and Crisostomi E., 2009), in that it allows easily widespread availability of the exercise environment: the ”communication environment” is basically a web-browser; students can operate from home, directly; no problem of installation of proprietary software on students PC should occur.

Such support could be in the visualization of experiment results, and in the provision of data for error analysis; and it can be also less conventional, and extend to recording and personalizing the student activity (e.g. by adapting to the learner's experience and goals the list of accessible exercises).

We propose a web based system interfacing robots for laboratory web-based activities. It supports learning activities in planning and control of autonomous robotic devices; different robots, in different configurations, can be available (either at the same time, in differently dressed areas of the laboratory, or in series); the exercises are defined by the teacher and specified to fit for certain learning goals and to certain robot configurations: learning goals of an exercise are specified in terms of difficulty to meet and competence to gain; *paths of experiences* are defined by the teacher as a sequence of exercises that the student is supposed to be able to perform at the end of the learning activity in the course; each learner is associated to a profile that is maintained during her/his activity: exercises are listed as available to the given learner only if the her/his profile according to the exercise's learning goals; after an exercise has been completed by the learner, her/his profile is updated by the skills gained from the learning goals and new exercises in the path are possibly made available.

The laboratory is located at the Faculty of Information Engineering of Sapienza University, in the premises of Latina, and it can be reached at the web address http://infocli31.dislt.uniroma1.it/webRobot/, where at present, due to its current use, only a Italian language version is available.

The main contribution is related to the particular software design which gives the possibility of specific didactic paths construction, tailored on each student initial skills and desired learning goals.

In the most recent literature, several examples of web based laboratories are illustrated. As evidenced in (García–Zubia J., Orduña P., López–de–Ipiña D. and Alves G.R., 2009), the most of them put their main attention in the hardware aspects, banishing the software to a little more than a mere communication interface.

A wide analysis of the state of the art in remote laboratories is performed in (Gomes L. and Bogosyan S., 2009), where a first simplified classification of local/remote and simulated/hands-on experiments is given, and some benefits are evidenced.

Several applications of virtual and augmented reality have been known to have some didactic usage, such as in (Kosuge K., Kikuchi J. and Takeo K., 2002; Marín R., Sanz P.J., Nebot P. and Wirz R., 2005). An application of remote accessibility of an experimental lab for student with disabilities is in (Colwell C., Scanlon E. and Cooper M., 2002). iLab (Lerman S. and del Alamo J., 2000-2005) makes accessible laboratories developed at MIT.

Regarding the interaction between remotely available laboratories and distance learning services, (Chellali R., Dumas C., Mollet N. and Subileau G., 2009) laments that the problems, in making complex systems available through e-learning, are rather conceptual than technical.

(Borgolte U., 2009) presents the architecture of a remotely operable laboratory with a mobile robot, used for education. Programming is done in COLBERT, offering quasi-parallel execution of activities.

In (Casini M., Prattichizzo D. and Vicino A., 2004) Automatic Control Telelab is presented, as a web-based (through java applet technology) system, allowing to put on-line a series of experiences in various fields of automation and control. Students have access, through the “experiment interface”, to the graphically and video rendered outputs of the experiment. There are several experimental settings, and an additional field related to LEGO Mindstorm is announced.

A survey on Web technologies used in control systems courses can be found in (Poindexter S. E. and Heck B. S., 1999), where the authors describe the use of virtual (Merrick C. M. and Ponton J. W., 1996; Schmid C., 1998) and remote labs. A distinguishing feature of remote labs as compared to virtual labs is that users can interact with real physical processes through the Internet, making them more attractive than controlling software simulations. At the same time, the design and implementation of a remote lab is more challenging due to safety and fault tolerant aspects.

In (Jara C. A., Candelas F. A. and Torres F., 2008) an environment for remote operation and simulation for a robot arm is described. It uses easy java simulation for the applet-based graphical interface, making it necessary on the local PC only a web browser and java+java3D runtime system. A dedicated and expensive robot is made available.

(López D., Cedazo R., Sánchez F.M. and Sebastian J.M., 2009) describes a web accessible experimental setup for embedded real time systems programming for a robot arm. Free software is used both for client and for server applications. The hardware setup is fixed and the different sessions propose different tasks to be executed by the robot arm; the tested skill is the ability of produce a correct and functional program for executing the given task.

The here presented remote laboratory, MindLab, has been designed and built up focusing on the possibility of a quite flexible and configurable exercise settings, where three main components - configurations of robots, experiment areas, exercises definitions - of the remote learning activities can be accommodated in varied manners: different robot(s) configurations can be defined, different playground(s) settings can be arranged and (clusters of) exercises can be defined according to specific robot and layout configurations.

Our system provides a quite simple-to-use environment supporting programming and experimenting on Lego Mindstorms robots, a family of commercial robots which is widely available, reasonably sophisticated in terms of functions, pretty easy to use and sufficiently affordable in terms of cost. Moreover, being intrinsically modular in terms of sensors and actuators number and distribution, is is easily and quickly reconfigurable and several different configurations can be obtained starting from the same set of devices.

In MindLab, though, the main issue we worked on so far is in giving rise to a truly e-learning framework, capable to support methodologically and pedagogically sound distance activities, and in particular personalized and adaptive management of the learner’s experience (Fernandez, G., Sterbini, A. and Temperini, M., 2007; Limongelli, C., Sciarrone, F., Temperini, M., and Vaste, G., 2009; Sterbini, A. and Temperini, M., 2009; Limongelli, C., Sciarrone, F., Temperini, M., and Vaste, G., 2010).

2 THE MINDLAB FRAMEWORK

MindLab is basically a web-application running on a server resident into a small laboratory: the server can access, via bluetooth connection, the robots, as instructed by the application. The communication scheme is depicted in Figure 1

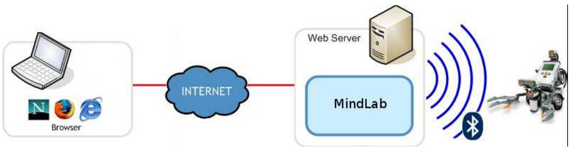


Figure 1: Communication scheme in MindLab.

The system allow to make exercises with robots based on the LEGO Mindstorms NXT technology. This choice has been driven by the idea that such a system could give sufficient flexibility to the laboratory experimental layouts, besides being economically feasible and allowing to use and maintain several robots.

To program the robot, the support can be based on graphical interfaces, such as in NXT-G, Robolab, and NI LabVIEW Toolkit, or on the use of a plain programming language, such as C, C++, URBI, leJOS, NXC, NBC. The original firmware included in the “brick” NXT does support only the last two languages; all the other systems need a dedicated firmware to replace the original.

Then, a first issue was in the possibility to replace the original firmware of the brick by a new one, more extended and complex. The reward in that replacement was in the possibility to get total control of the hardware, import algorithms (e.g. from artificial intelligence disciplines) more easily and enjoy better performances. A second issue for decisions was in the programming language to be supported by the system.

As of the former issue, we thought that full control could also make it necessary to display higher level of programming skills and to deal with less predictable and tractable programming errors.

Then, the decision to limit programming skills and freedom and maintain the original firmware was adopted.

Regarding the latter issue, another constraint in our project was the will to make an additional interface available to input pseudo-code, rather than plain programming code. The pseudo-code (that is a more straightforward way to specify the robot behavior, expressed through system supported menus of commands) would then be translated in the programming language supported by the system before processing its execution. This would allow for a lower/simpler interaction level between student and system: it can be useful at initial stages. At a higher level of student skills and responsibility, direct input of plain programming code is also possible. So one of the factors in our choice of the programming language supported by the system was in having it to be the target for pseudo-code translation. These considerations bring us to selected *Not Exactly C* (NXC). NXC is an high level programming language deriving from the NBC (*Next Bytes Code*, used in previous LEGO technology *RXC*)

Communication between server and Bricks can be based both on a cabled USB connection and on a wireless Bluetooth one.

To complete the hardware architecture, a webcam based system for the remote visualization of the running experiment is included.

3 THE EXERCISES ENVIRONMENT

The system can manage several *courses*; each course is an environment in which paths of exercises are defined, to reach determined skills.

An exercise is defined by specifying the robot layout for its execution and two cognitive aspects: difficulty degree and competence level (both are numbers in [1,5]; the latter represents the skills that are demonstrated/gained by solving the exercise).

Each course have several students enrolled, and one or more teachers responsible.

An administrator manage the system, e.g. grant to certain users teacher privileges, create new courses and assign their teachers, configure exercise assessment and execution ways, browse the log file containing records of the system activities, and configure in the system the robots that are available on the ground (basically asserting the physical robot configuration in terms of sensors and engines attached.

The students enrolled in a course have access to the list of exercises they can perform in that course, and to the records of performed exercises. A student can perform and exercise if the related difficulty and competence level are *sufficiently close* to her/his skills. A learner’s skills are represented through a model, constituted directly by the records of the exercises previously solved by the learner.

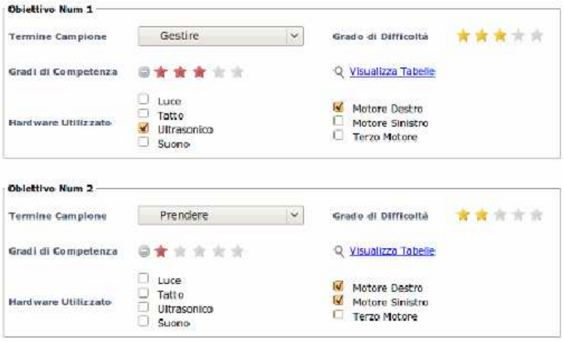
To solve an exercise, the student devises a computer program (either directly, in NXC, or by listing its pseudo-code through and assistive interface), and, after being entitled to access an available robot, inputs the program and sees the results. As a matter of facts, exercise assessment is just semi-automatic: the final judgment is stated by the teacher, that can declare the exercise solved to grant the learner with the related difficulty and competence levels.

Then, the teachers monitor and assess the exercises performed by their courses' students.

Moreover they specify new exercises. Figure 2 shows part of the definition interface. Here the less straightforward among the data managed by the system to define an exercise: ROBOT CONFIGURATION for the exercise, selected among those defined by the administrator, corresponding to the machine actually available in the lab; HARDWARE SPECIFICATION, i.e. the set of devices (sensors and actuators) that are supposed to be needed for the exercise (and so present in the robot configuration); DIFFICULTY degree, meant to express how challenging, in terms of coordination and complexity of the control, the endeavor can be; it's an integer number $\in [1, \ldots, 5]$; COMPETENCE level that is supposed to be shown in solving the exercise; it's a number as above. Note that this is not a prerequisite: it is more similar to the certification of a learner's *cognitive level* (cf. Bloom, 1964) granted by the learning activity associated to the exercise.

There is one more part in an exercise definition, and it is very important: the activity is definitely specified by listing a set of *objectives*, sort of sub-tasks for the whole activity. Although such segmentation could be pedagogically useful in providing an order of action for the students, it is not particularly stressed in the syste: As a matter of facts, it is over the objectives that the teacher, with the system's assistance, actually states the difficulty and competence levels, and the hardware specifications. Then, the exercise is the *summa* of the sub-tasks (its levels are the highest among the objectives' ones, and its hardware is the union of the objectives' requests).

The definition interface for objectives is in Figure 2.



Obiettivo Num 1

Termine Campione: Gestire

Grado di Difficoltà: ★★★☆☆

Gradi di Competenza: ★★★☆☆

Visualizza Tabelle

Hardware Utilizzato: ☐ Luce ☐ Tatto ☑ Ultrasonico ☐ Suono ☑ Motore Destro ☐ Motore Sinistro ☐ Terzo Motore

Obiettivo Num 2

Termine Campione: Prendere

Grado di Difficoltà: ★★☆☆☆

Gradi di Competenza: ★☆☆☆☆

Visualizza Tabelle

Hardware Utilizzato: ☐ Luce ☐ Tatto ☐ Ultrasonico ☐ Suono ☑ Motore Destro ☑ Motore Sinistro ☐ Terzo Motore

Figure 2: Two objectives in the definition of an exercise.

To help teachers in assigning difficulty levels to objectives a table of correspondence between keywords and levels is available. For instance, keywords such as "Turn+DIRECTION" or "Stop" are given automatically difficulty 1; ”Measure" is level 3; "Looking_For" is level 4. Presently we have 30 keyword (distributed in the 5 levels) in the table. The teacher is also provided with a set of accompanying recommendations related to the pedagogycal interpretation of the keywords/levels; e.g. if the objective requires to "manage different types of movements" it will have at least a classification 2 of difficulty.

Competence levels basically express programming capabilities and the knowledge and solution of typical problems. In order to mark such level in an objective the teacher is required to peruse an help table, specifying at the moment, some remark about each level and 20 indications on the levels, overall. For instance, basic knowledge of sensors and engines, basic terminology, basic movements, are the indications matchable in cognitive level 1; at level 4, instead, capabilities such as "knowledge of the hardware elements and ability to let it interact", "capability to use all the programming commands supported in MindLab"; while a level 5 capability is "knowledge of MindLab commands limits".

Keywords and recommendations are not fixed once and for all; we think that different courses (meaning exercises cycles) and teachers might require different descriptions for the levels of difficulty and competence. So, keywords and recommendations in the system are actually completely reconfigurable and expandable, also at the level of the single course (and related teacher). We think this can help avoiding to frame everybody in a predefined pedagogical format, and increase teacher comfort in the system.

3.1 An Example of Exercise

TEXT OF THE EXERCISE

The robot should follow a black line on the floor. At line end, it has to check around for obstacles closer than 15cm. If no obstacle is seen, emit a sound, otherwise measure the intensity of light reflected by the floor and output it.

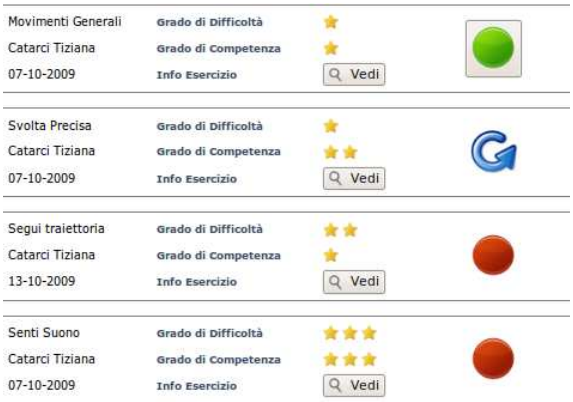
OBJECTIVES

An analysis of the text (we underlined significant segments) helps defining the objectives of the exercise. They are enumerated in the following, giving *keyword*, *difficulty* (*D*) and *Competence level* (*C*): *Follow a Line*, *D* = 3; *C* = 3; *Check* (for obstacle), *D* = 4, *C* = 4; *Emit Sound*, *D* = 1, *C* = 2; *Measure*, *D* = 3, *C* = 3.

So the exercise has four objectives and a overall qualification of: Difficulty = 4; Competence = 4.

3.2 Solving an Exercise

Figure 3 shows the student interface for exercises. Exercises whose solution the student can attempt are pointed out by a green dot. Those exercises for which the learner submitted a solution are also pointed out, to mean whether they have been accepted or rejected by the teacher, or their assessment is pending. Exercises that are defined with hardware requirements not matching an available robot configuration would be pointed out (not in figure).



Movimenti Generali — Grado di Difficoltà ★
Catarci Tiziana — Grado di Competenza ★
07-10-2009 — Info Esercizio

Svolta Precisa — Grado di Difficoltà ★
Catarci Tiziana — Grado di Competenza ★★
07-10-2009 — Info Esercizio

Segui traiettoria — Grado di Difficoltà ★★
Catarci Tiziana — Grado di Competenza ★
13-10-2009 — Info Esercizio

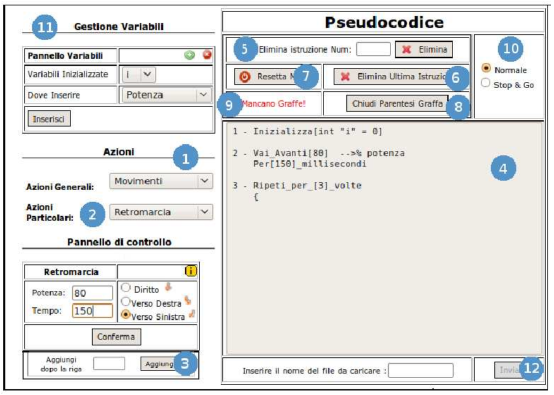
Senti Suono — Grado di Difficoltà ★★★
Catarci Tiziana — Grado di Competenza ★★★
07-10-2009 — Info Esercizio

Figure 3: The learner's interface to exercises. The first in list has $D = 1$ and $C = 1$. The second was attempted and is pending assessment. The others are not yet available since their Ds and Cs are not yet suffciently close to the leraner's profile. When the submitted exercise will be accepted, the profile will increase to $D = 1$ and $C = 2$. This will make exercises with $D \leq 2$ and $C \leq 3$ available to the learner.

From here the student can access the solution interface for the selected exercise, and submit a program proposal in two ways.

The first way is to use a very plain interface, allowing to just paste the program and sending it along. At this stage only compiler-related feedbacks are available (if the program compiles well, it is send to the selected robot for execution). There is no simulation-baed check at this stage, to prevent patently wrong programs to execute, nor recovery programs are set into the system to let the robot return to a safe position afetr an unfortunate experiment. This is an operative lack in the system so far.

The second way to submit a solution is through a pseudo-code subsystem, that allow the student to develop the program in a guided manner, by specifying the sequence of movements and actions through menus and parameter sub-panels (Figure4).



Gestione Variabili

Pannello Variabili
Variabili Inizializzate: i
Dove Inserire: Potenza

Azioni

Azioni Generali: Movimenti
Azioni Particolari: Retromarcia

Pannello di controllo

Retromarcia
Potenza: 80
Tempo: 150
○ Diritto
○ Verso Destra
● Verso Sinistra

Aggiungi dopo la riga

Pseudocodice

Elimina istruzione Num:

● Normale
○ Stop & Go

Mancano Graffe!

```
Inizializza[int "i" = 0]

Vai_Avanti[80]  -->% potenza
Per[150]_millisecondi

Ripeti_per_[3]_volte
  {
```

Inserire il nome del file da caricare :

Figure 4: The pseudo-code interface. Menus allows to select the operations to list in the program. For certain operation a sub-panel opens up, to allow the setting of due parameters.

4 CONCLUSIONS

The paper has shown design and implementation aspects of the system MindLab, a web application supporting remote operation of LEGO Mindstorms robots for e-teaching and e-learning activities. The choice of such commercial hardware makes, without prejudice for the flexibility and the learning effectiveness, pretty affordable the task of making a laboratory available to distance learning, also in cases when the economic resources of a school are limited.

Our approach imports, in such a didactic teleoperation setting, our experience in the field of personalized and adaptive e-learning. In each course the teacher can define an exercising-learning path of mandatory exercises; the learner, then, can approach only those exercises whose pedagogical specifications are sufficiently close to the present cognitive state of the learner.

We have several robots available to remote exercising at the same time, both in separated environments (where different students can can operate with different robots in different portions of the laboratory, and for collaborative exercises with coordinated machines.

We have remarked the flexibility of the (small) exercise environment, with the possibility to define various layouts for different tasks, and the system capability to manage different robots, each one with its configuration, for different exercises. And we have pointed out some lacks in the present implementation of the system, namely the absence of any simulation devices (that could avoid execution of wrong plans) and the scarce recovery mechanisms to enact in case of execution of such wrong programs, or other accidents.

In terms of future work we plan to remedy to the abovementioned lack, by equipping the system with dedicated modules. There are some other lines development: one is the definition of a more accurate, yet not more cumbersome, method to define the pedagogical characterization of an exercise; a second line of research is the definition of a more permanent student model, such as it is not entirely initialized when a new course is endeavored by the learner.

From an adaptive-e-learning point of view, then, we will try to formalize and implement a framework in which the sequence of exercises for a student, is configured according to the learner's individual cognitive state and learning goals, and possibly changed (i.e. *adapted*) during the course activity, responding to changes in the student model.

REFERENCES

Traylor R. L., Heer D. and Fiez T. S. (2003). Using an Integrated Platform for Learning to Reinvent Engineering Education. *IEEE Transactions on Educations*. Vol. 46, No. 4, pp. 409–419.

Gomes L. and Bogosyan S. (2009). Current Trends in Remote Laboratories. *IEEE Transactions on Industrial Electronics*. Vol. 56, No. 12, pp. 4744–4756.

López D., Cedazo R., Sánchez F.M. and Sebastian J.M. (2009). Ciclope Robot: Web–Based System to Remote Program an Embedded Real–Time System. *IEEE Transactions on Industrial Electronics*. Vol. 56, No. 12, pp. 4791–4797.

Balestrino A., Caiti A. and Crisostomi E. (2009). From Remote Experiments to Web–Based .Learning Objects: An Advanced Telelaboratory for Robotics and Control Systems. *IEEE Transactions on Industrial Electronics*. Vol. 56, No. 12, pp. 4817–4825.

García–Zubia J., Orduña P., López–de–Ipiña D. and Alves G.R. (2009). Addressing Software Impact in the Design of Remote Laboratories. *IEEE Transactions on Industrial Electronics*. Vol. 56, No. 12, pp. 4757–4767.

Kosuge K., Kikuchi J. and Takeo K. (2002). VISIT: A teleoperation system via the computer network. Beyond Webcams:an introduction to online robots. MIT Press, pp. 215–220.

Marín R., Sanz P. J., Nebot P. and Wirz R. (2005). A Multimodal Interface to Control a Robot Arm via the Web: A Case Study on Remote Programming. *IEEE Transactions on Industrial Electronics*. Vol. 52, pp. 1506–1520.

Colwell C., Scanlon E. and Cooper M. (2002). Using remote laboratories to extend access to science and engineering. *Computers & Education*. Vol. 38, pp. 65–76.

Lerman S. and del Alamo J. (2000-2005) iLab: Remote Online Laboratories. [Online]. available: http://icampus.mit.edu/projects/iLabs.shtml.

Chellali R., Dumas C., Mollet N. and Subileau G. (2009). SyTroN: A Virtual Classroomfor Collaborative and Distant E-Learning Systemby Teleoperating Real Devices. *International Journal of Computer Games Technology*. Vol. 2009, doi:10.1155/2009/627109. Hindawi.

Borgolte U. (2009). Interface design of a virtual laboratory for mobile robot programming. In *Mndez-Vilas, A., Solano Martn, A. , Mesa Gonzlez, J.A. and Mesa Gonzlez, J. (Eds.) Proceedings of the m-ICTE2009 Int. Conf. on Research, Reflections and Innovations in Integrating ICT in Education*. 22-24 Apr. 2009, Lisbon, Portugal. Vol. 1. ISBN 978-84-692-1789-4. FORMATEX, Badajoz, Spain.

Casini M., Prattichizzo D. and Vicino A. (2004). The Automatic Control Telelab. A Web-based technology for Distance Learning. *IEEE Control Systems Magazine*. Vol. 24, No. 3, pp. 36–44.

Poindexter S. E. and Heck B. S. (1999). Using the Web in your courses: What can you do? What should you do?. *IEEE Control Systems Magazine*. Vol. 19, No. 1, pp. 83–92.

Merrick C. M. and Ponton J. W. (1996). The ECOSSE control hypercourse. *Comput. Chem. Eng.*. Vol. 20, sup., No. 972, pp. S1353-S1358.

Schmid C. (1998). The virtual lab VCLAB for education on the Web. in *Proc. American Control Conference*. Philadelphia, PA, pp. 1314-1318.

Jara C. A., Candelas F. A. and Torres F. (2008). Virtual and Remote Laboratory for Robotics ELearning. in *Proceedings 18th European Symposium on Computer Aided Process Engineering ESCAPE 18, Braunschweig B. and Joulia X. (Eds)*. Elsevier.

Fernandez, G., Sterbini, A. and Temperini, M. (2007). Learning Objects: A Metadata Approach. In *Proc. IEEE Region 8 Eurocon 2007 Conference*. Warsaw, Poland.

Limongelli, C., Sciarrone, F., Temperini, M., and Vaste, G. (2009). Adaptive Learning with the LS-Plan System: A Field Evaluation. *IEEE Trans. Learning Technologies*. Vol.2, No. 3, pp.203–215.

Sterbini, A. and Temperini, M. (2009). Adaptive Construction and Delivery of Web-Based Learning Paths. In *Proc. 39th ASEE/IEEE Frontiers in Education Conference*. San Antonio, TX, USA.

Limongelli, C., Sciarrone, F., Temperini, M., and Vaste, G. (2010). *The Lecomps5 Framework for Personalized Web-Based Learning: a Teacher's Satisfaction Perspective*. Computers in Human Behaviour, Elsevier, to appear.

Bloom, B. S. (1964). *Taxonomy of Educational Objectives*. David McKay Company Inc.

INTEGRATING CONTEXT INTO INTENT RECOGNITION SYSTEMS

Richard Kelley, Christopher King, Amol Ambardekar, Monica Nicolescu, Mircea Nicolescu
Department of Computer Science and Engineering, University of Nevada, Reno, 1664 N. Virginia St., Reno, U.S.A.
{*rkelley, cjking, ambardek, monica, mircea*}*@cse.unr.edu*

Alireza Tavakkoli
Department of Computer Science, University of Houston-Victoria, 3007 N. Ben Wilson, Victoria, U.S.A.
tavakkolia@uhv.edu

Keywords: Intent recognition, Human-robot interaction, Natural language processing.

Abstract: A precursor to social interaction is social understanding. Every day, humans observe each other and on the basis of their observations "read people's minds," correctly inferring the goals and intentions of others. Moreover, this ability is regarded not as remarkable, but as entirely ordinary and effortless. If we hope to build robots that are similarly capable of successfully interacting with people in a social setting, we must endow our robots with an ability to understand humans' intentions. In this paper, we propose a system aimed at developing those abilities in a way that exploits both an understanding of actions and the context within which those actions occur.

1 INTRODUCTION

A precursor to social interaction is social understanding. Every day, humans observe each other and on the basis of their observations "read people's minds," correctly inferring the goals and intentions of others. Moreover, this ability is regarded not as remarkable, but as entirely ordinary and effortless. If we hope to build robots that are similarly capable of successfully interacting with people in a social setting, we must endow our robots with an ability to understand humans' intentions. In this paper, we propose a system aimed at developing those abilities in a way that exploits both an understanding of actions and the context within which those actions occur.

2 RELATED WORK

Whenever one wants to perform statistical classification in a system that is evolving over time, hidden Markov models may be appropriate (Duda et. al., 2000). Such models have been very successfully used in problems involving speech recognition (Rabiner, 1989). Recently, there has been some indication that hidden Markov models may be just as useful in modeling activities and intentions. For example, HMMs have been used by robots to classify a number of manipulation tasks (Pook and Ballard, 1993)(Hovland et. al., 1996)(Ogawara et. al., 2002). These approaches all have the crucial problem that they only allow the robot to detect that a goal has been achieved *after* the activity has been performed; to the extent that intent recognition is about prediction, these systems do not use HMMs in a way that facilitates the recognition of intentions. Moreover, there are reasons to believe (see below) that without considering the disambiguation component of intent recognition, there will be unavoidable limitations on a system, regardless of whether it uses HMMs or any other classification approach.

Extending upon this use of HMMs to recognize activities, previous work (of which (Tavakkoli et. al., 2007) is representative) has examined the use of HMMs to predict intentions. The work to date has focused on training a robot to use HMMs (via theory-of-mind inspired approaches), and on the best information to represent using those models. That work has mostly ignored questions such as scalability and the role that contextual information plays in the recognition process. The present work begins to address these issues, and introduces the idea of using a digraph-based language model to provide contextual knowledge.

To build that language model, we use the typed dependency extraction facilities of the Stanford Parser (Marneffe et. al., 2006). In contrast with constituency grammar, which views a sentence as being made up of phrases, dependency grammar represents grammatical links between pairs of words. *Typed* dependencies explicitly label the links between words with grammatical relations (Marneffe et. al., 2006). To the best of our knowledge, such a representation has not yet been used as the basis for any human-robot interaction work; our graph-based approach is also new.

3 LEXICAL DIGRAPHS

As mentioned above, our system relies on contextual information to perform intent recognition. While there are many sources of contextual information that may be useful to infer intentions, we chose to focus primarily on the information provided by object affordances, which indicate the actions that one can perform with an object. The problem, once this choice is made, is one of training and representation: given that we wish the system to infer intentions from contextual information provided by knowledge of object affordances, how do we learn and represent those affordances? We would like, for each object our system may encounter, to build a representation that contains the likelihood of all actions that can be performed on that object.

Although there are many possible approaches to constructing such a representation, we chose to use a representation that is based heavily on a graph-theoretic approach to natural language – in particular, English. Specifically, we construct a graph in which the vertices are words and a labeled, weighted edge exists between two vertices if and only if the words corresponding to the vertices exist in some kind of grammatical relationship. The label indicates the nature of the relationship, and the edge weight is proportional to the frequency with which the pair of words exists in that particular relationship. For example, we may have vertices $drink$ and $water$, along with the edge $((drink, water), \ direct\_object, \ 4)$, indicating that the word "water" appears as a direct object of the verb "drink" four times in the experience of the system. From this graph, we compute probabilities that provide the necessary context to interpret an activity.

3.1 Dependency Parsing and Graph Representation

To obtain our pairwise relations between words, we use the Stanford labeled dependency parser. The parser takes as input a sentence and produces the set of all pairs of words that are grammatically related in the sentence, along with a label for each pair, as in the "water" example above.

Using the parser, we construct a graph $G = (V, E)$, where E is the set of all labeled pairs of words returned by the parser for all sentences, and each edge is given an integer weight equal to the number of times the edge appears in the text parsed by the system. V then consists of the words that appear in the corpus processed by the system.

3.2 Graph Construction and Complexity

3.2.1 Graph Construction

Given a labeled dependency parser and a set of documents, graph construction is straightforward. Briefly, the steps are

1. Tokenize each document into sentences.
2. For each sentence, build the dependency parse of the sentence.
3. Add each edge of the resulting parse to the graph.

Each of these steps may be performed automatically with reasonably good results, using well-known language processing algorithms. The end result is a graph as described above, which the system stores for later use.

One of the greatest strengths of the dependency-grammar approach is its space efficiency: the output of the parser is either a *tree* on the words of the input sentence, or a graph made of a tree plus a (small) constant number of additional edges. This means that the number of edges in our graph is a linear function of the number of nodes in the graph, which (assuming a bounded number of words per sentence in our corpus) is linear in the number of sentences the system processes. In our experience, the digraphs our system has produced have had statistics confirming this analysis, as can be seen by considering the graph used in our recognition experiments. For our corpus, we used two sources: first, the simplified-English Wikipedia, which contains many of the same articles as the standard Wikipedia, except with a smaller vocabulary and simpler grammatical structure, and second, a collection of childrens' stories about the objects in which

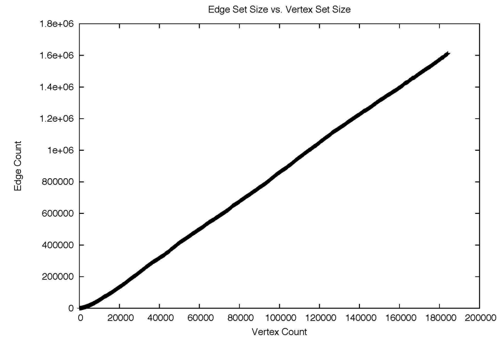


Figure 1: The number of edges in the Wikipedia graph as a function of the number of vertices during the process of graph growth.

we were interested. In Figure 1, we show the number of edges in the Wikipedia graph as a function of the number of vertices at various points during the growth of the graph. The scales on both axes are identical, and the graph shows that the number of edges for this graph does depend linearly on the number of vertices.

The final Wikipedia graph we used in our experiments consists of 244,267 vertices and 2,074,578 edges. The childrens' story graph is much smaller, being built from just a few hundred sentences: it consists of 1754 vertices and 3873 edges. This graph was built to fill in gaps in the information contained in the Wikipedia graph. The graphs were merged to create the final graph we used by taking the union of the vertex and edge sets of the graphs, adding the edge weights of any edges that appeared in both graphs.

4 VISION-BASED CAPABILITIES

In support of our intent recognition system, we require a number of visual capabilities for our robot. Among these, our system must be able to segment and track the motion of both humans and inanimate objects. Because we are interested in objects and their affordances, our system must also be able to visually identify objects and, for objects whose state can change over time, object states. Moreover, tracking should be done in three-dimensional space. To support this last requirement, we use a stereo-vision camera.

To perform segmentation and object recognition, we use a variant of maximally stable extremal regions for color images (Forssen, 2007). In our variant, we identify "strong" and "weak" edges in the image (based on our thresholding), and constrain the region-merging of color-based MSER so that region growth is inhibited across weak edges and prevented entirely across strong edges. This approach allows for increased stability, for multiple regions of different homogeneity to coexist near one another, and for more coherent segmentation of textured regions.

Having segmented a frame into regions, we perform object recognition using a mixture of Gaussians, computing probabilities at the *region* level rather than the pixel level. Because objects tend to consist of a smaller number of regions than pixels, this can lead to a substantial speedup.

Once we have segmented a frame and identified the regions of interest in that frame, we perform tracking via incremental support vector data descriptions and connected component analysis. We refer the interested reader to other, vision-specific work (Tavakkoli et. al., 2007).

5 INTENT RECOGNITION SYSTEM

5.1 Low-level Recognition via Hidden Markov Models

As mentioned above, our system uses HMMs to model activities that consist of a number of parts that have intentional significance. Recall that a hidden Markov model consists of a set of hidden states, a set of visible states, a probability distribution that describes the probability of transitioning from one hidden state to another, and a probability distribution that describes the probability of observing a particular visible state given that the model is in a particular hidden state. To apply HMMs, one must give an interpretation to both the hidden states and the visible states of the model, as well as an interpretation for the model as a whole. In our case, each model represents a single well-defined activity. The hidden states of represent the intentions underlying the parts of the activity, and the visible symbols represent changes in measurable parameters that are relevant to the activity. Notice in particular that our visible states correspond to dynamic properties of the activity, so that our system can perform recognition as the observed agents are interacting.

We train our HMMs by having our robot perform the activity that it later will recognize. As it performs the activity, it records the changes in the parameters of interest for the activity, and uses those to generate sequences of observable states representing the activity. These are then used with the Baum-Welch algorithm (Rabiner, 1989) to train the models, whose topologies have been determined by a human operator in advance.

During recognition, the stationary robot observes a number of individuals interacting with one another and with stationary objects. It tracks those individuals using the visual capabilities described above, and takes the perspective of the agents it is observing. Based on its perspective-taking and its prior understanding of the activities it has been trained to understand, the robot infers the intention of each agent in the scene. It does this using maximum likelihood estimation, calculating the most probable intention given the observation sequence that it has recorded up to the current time for each pair of interacting agents.

5.2 Adding Context

Our system uses contextual information to infer intentions. This information is linguistic in nature, and in section 3 we show how lexical information representing objects and affordances can be learned and stored automatically. In this subsection, we outline how that lexical information can be converted to probabilities for use in intent recognition.

Context and Intentions. In general, the context for an activity may be any piece of information. For our work, we focused on two kinds of information: the location of the event being observed, and the identities of any objects being interacted with by an agent. Context of the first kind was useful for basic experiments testing the performance of our system against a system that uses no contextual information, but did not use lexical digraphs at all; contexts and intentions were defined by entirely by hand. Our other source of context, object identities, relied entirely on lexical digraphs. In experiments using this source of information, objects become the context and their affordances – represented by verbs in the digraph – become the intentions. As explained below, if s is an intention and c is a piece of contextual information, our system requires the probability $p(s \mid c)$, or in other words the probability of an affordance given an object identity. This is exactly what is provided by our digraphs. If "water" appears as a direct object of "drink" four times in the robot's linguistic experience, then we can obtain a proper probability of "drink" given "water" by dividing four by the sum of the weights of all edges which have "water" as the direct object of some word. In general, we may use this process to obtain a table of probabilities of affordances or intentions for every object in which our system might be interested, as long as the relevant words appear in the corpus. Note that this may be done without human intervention.

Inference Algorithm. Suppose that we have an activity model (*i.e.* an HMM) denoted by w. Let s denote an intention, let c denote a context, and let v denote a sequence of visible states from the activity model w. If we are given a context and a sequence of observation, we would like to find the intention that is maximally likely. Mathematically, we would like to find

$$\arg\max_s p(s \mid v, c),$$

where the probability structure is determined by the activity model w.

To find the correct s, we start by observing that by Bayes' rule we have

$$\max_s p(s \mid v, c) = \max_s \frac{p(v \mid s, c) p(s \mid c)}{p(v \mid c)}. \qquad (1)$$

We can further simplify matters by noting that the denominator is independent of our choice of s. Moreover, we assume without loss of generality that the possible observable symbols are independent of the current context. Based on these observations, we can write

$$\max_s p(s \mid v, c) \approx \max_s p(v \mid s) p(s \mid c). \qquad (2)$$

This approximation suggests an algorithm for determining the most likely intention given a series of observations and a context: for each possible intention s for which $p(s \mid c) > 0$, we compute the probability $p(v \mid s)p(s \mid c)$ and choose as our intention that s whose probability is greatest. The probability $p(s \mid c)$ is available, either by assumption or from our linguistic model, and if the HMM w represents the activity model associated with intention s, then we assume that $p(v \mid s) = p(v \mid w)$. This assumption may be made in the case of location-based context for simplicity, or in the case of object affordances because we focus on simple activities such as reaching, where the same HMM w is used for multiple intentions s. Of course a perfectly general system would have to choose an appropriate HMM dynamically given the context; we leave the task of designing such a system as future work for now, and focus on dynamically deciding on the context to use, based on the digraph information.

5.3 Intention-based Control

In robotics applications, simply determining an observed agent's intentions may not be enough. Once a robot knows what another's intentions are, the robot should be able to act on its knowledge to achieve a goal. With this in mind, we developed a simple method to allow a robot to dispatch a behavior based on its intent recognition capabilities. The robot first infers the global intentions of all the agents it is tracking, and for the activity corresponding to the inferred global intention determines the most likely local intention. If the robot determines over multiple time steps that a certain local intention has the largest probability, it can dispatch a behavior in response to the situation it believes is taking place.

6 EXPERIMENTAL VALIDATION

We performed a series of experiments to test different aspects of our system. In particular, we wished to test the following claims: first, that contextual information could improve performance over a context-agnostic system; second, that intention-based control as described above can be used to solve basic, but still realistic, problems; and last, we wished to test lexical digraphs as a source of contextual information for inferring intentions related to objects and their affordances.

6.1 Setup

To validate our contextual approach, we performed a set of experiments using a Pioneer 3DX mobile robot, with an on-board computer, a laser rangefinder, and a stereo camera. We trained our robot to understand three basic activities: *following*, in which one agent trails behind another; *meeting*, in which two agents approach one another directly; and *passing*, in which two agents move past each other without otherwise directly interacting. We also built a model of *reaching* for use in the object-based tests.

We placed our trained robot in several indoor environments and had it observe the interactions of multiple human agents with each other, and with multiple static objects. In our experiments, we considered both the case where the robot acts as a passive observer and the case where the robot executes an action on the basis of the intentions it infers in the agents under its watch.

The first set of experiments was performed in a lobby, and had agents meeting each other and passing each other both with and without contextual information about which of these two activities is more likely in the context of the lobby. To the extent that meeting and passing appear to be similar, we would expect that the use of context would help to disambiguate the activities. Although contrived, these scenarios do facilitate direct comparison between a context-aware and a context-agnostic system.

To test our intention-based control, we set up two scenarios. In the first scenario (the “theft” scenario), a human enters his office carrying a bag. As he enters, he sets his bag down by the entrance. Another human enters the room, takes the bag and leaves. Our robot was set up to observe these actions and send a signal to a “patrol robot” in the hall that a theft had occurred. The patrol robot is then supposed to follow the thief as long as possible.

Table 1: Quantitative evaluation - meet versus pass.

Scenario (with Context)	Correct Duration [%]
Meet (No context) - Agent 1	65.8
Meet (No context) - Agent 2	72.4
Meet (Context) - Agent 1	97.8
Meet (Context) - Agent 2	100.0

In the second scenario, our robot is waiting in the hall, and observes a human leaving the bag in the hallway. The robot is supposed to recognize this as a suspicious activity and follow the human who dropped the bag for as long as possible.

Lastly, to test the lexical-digraph-based system, we had the robot observe an individual as he performed a number of activities involving various objects. These included books, glasses of soda, computers, bags of candy, and a fire extinguisher.

6.2 Results

To provide a quantitative evaluation of intent recognition performance, we use two measures:

- *Accuracy rate* = the ratio of the number of observation sequences, of which the winning intentional state matches the ground truth, to the total number of test sequences.
- *Correct Duration* = C/T, where C is the total time during which the intentional state with the highest probability matches the ground truth and T is the number of observations.

6.2.1 Similar-looking Activities

As we can see from Table 1, the system performs substantially better when using context than it does without contextual information. Because *meeting* and *passing* can, depending on the position of the observer, appear very similar, without context it may be hard to decide what two agents are trying to do. With the proper contextual information, though, it becomes much easier to determine the intentions of the agents in the scene.

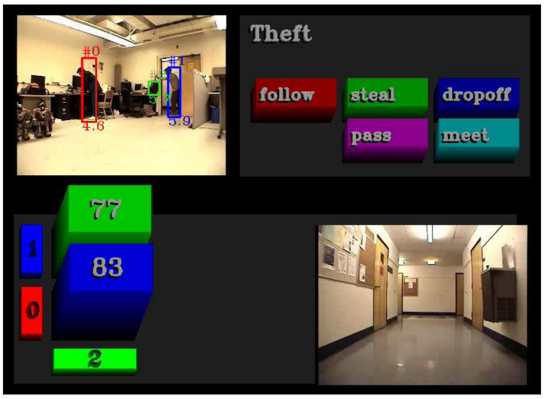


Figure 2: An observer robot catches an agent stealing a bag. The top left video is the observer's viewpoint, the top left bars represent possible intentions, the bottom right bars are the robot's inferred intentions for each agent (with corresponding probabilities), and the bottom right video is the patrol robot's viewpoint.

6.2.2 Intention-based Control

In both the scenarios we developed to test our intention-based control, our robot correctly inferred the ground-truth intention, and correctly responded the inferred intention. In the theft scenario, the robot correctly recognized the theft and reported it to the patrol robot in the hallway, which was able to track the thief. In the bag drop scenario, the robot correctly recognized that dropping a bag off in a hallway is a suspicious activity, and was able to follow the suspicious agent through the hall. Both examples indicate that intention-based control using context and hidden Markov models is a feasible approach.

6.2.3 Lexical-digraph-based System

To test the lexically-informed system, we considered three different scenarios. In the first, the robot observed a human during a meal, eating and drinking. In the second, the human was doing homework, reading a book and taking notes on a computer. In the last scenario, the robot observed a person sitting on a couch, eating candy. A trashcan in the scene then catches on fire, and the robot observes the human using a fire extinguisher to put the fire out.

Defining a ground truth for these scenarios is slightly more difficult than in the previous scenarios, since in these scenarios the observed agent performs multiple activities and the boundaries between activities in sequence are not clearly defined. However, we can still make the interesting observation that, except on the boundary between two activities, the correct duration of the system is 100%. Performance on the boundary is more variable, but it isn't clear that this is an avoidable phenomenon. We are currently working on carefully ground-truthed videos to allow us to better compute the accuracy rate and the correct duration for these sorts of scenarios.

7 CONCLUSIONS

In this paper, we proposed an approach to intent recognition that combines visual tracking and recognition with contextual awareness in a mobile robot. Understanding intentions in context is an essential human activity, and with high likelihood will be just as essential in any robot that must function in social domains. Our approach is based on the view that to be effective, an intent recognition system should process information from the system's sensors, as well as relevant social information. To encode that information, we introduced the lexical digraph data structure, and showed how such a structure can be built and used. We discussed the visual capabilities necessary to implement our framework, and validated our approach in simulation and on a physical robot.

REFERENCES

R. Duda, P. Hart, and D. Stork, *Pattern Classification*, Wiley-Interscience (2000)

L. R. Rabiner, A tutorial on hidden-Markov models and selected applications in speech recognition, in Proc. IEEE 77(2) (1989)

P. Pook and D. Ballard, Recognizing teleoperating manipulations, in Int. Conf. Robotics and Automation (1993), pp. 578585.

G. Hovland, P. Sikka and B. McCarragher, Skill acquisition from human demon- stration using a hidden Markov model, Int. Conf. Robotics and Automation (1996), pp. 27062711.

K. Ogawara, J. Takamatsu, H. Kimura and K. Ikeuchi, Modeling manipulation inter- actions by hidden Markov models, Int. Conf. Intelligent Robots and Systems (2002), pp. 10961101.

A. Tavakkoli, R. Kelley, C. King, M. Nicolescu, M. Nicolescu, and G. Bebis, "A Vision-Based Architecture for Intent Recognition," *Proc. of the International Symposium on Visual Computing*, pp. 173-182 (2007)

P. Forssen, "Maximally Stable Colour Regions for Recognition and Matching," *CVPR 2007.*

M. Marneffe, B. MacCartney, and C. Manning, "Generating Typed Dependency Parses from Phrase Structure Parses," *LREC 2006.*

HAND PROSTHESIS CONTROL

Software Tool for EMG Signal Analysis

Tomasz Suchodolski and Andrzej Wolczowski
Institute of Informatics, Automation and Robotics, Wroclaw University of Technology
Wybrzeze Wyspianskiego 27, 50-370 Wroclaw, Poland
{tomasz.suchodolski, andrzej.wolczowski}@pwr.wroc.pl

Keywords: Hand Prosthesis, EMG Signals, Decision Tree, Pattern Recognition.

Abstract: The paper discusses the problem of the decision process of controlling the bio-prosthesis of the hand that is treated as the human intention recognition by means of the analysis of the electromyography (EMG) signals from the hand muscles. The number of movements, which is indispensable for the dexterity of the prosthesis, makes the recognition not entirely reliable. The approach presented herein includes three methods: the decision tree, neuron networks, and genetic algorithms in order to enhance the reliability of the EMG signal recognition. Simultaneously, the paper presents the software designed for the needs of the research and adapted to processing the EMG signals in compliance with these methods.

1 INTRODUCTION

The dexterity of the human hand is the result of both its kinematic structure and the control of it by the central nervous system. The construction of multi-joint anthropomorphic mechanic structure that can copy even very complicated movements of the human hand poses at present no problem. However, the question of the reliable bio-control of such a dexterous prosthesis remains unsolved.

The usual practice is to make use of myopotentials – which are electrical signals that accompany the activity of the muscles (Wolczowski, 2001). This is feasible since, after the amputation of the hand, there remain in the stamp a significant number of the muscles that earlier controlled the movement of the fingers (Su Y, et al., 2007; Mohammadreza, 2007). Through the tensing of these muscles, the person with a disability may express their intentions to control their prosthesis as they do with their healthy hand. The intention recognition calls for efficient analysis of EMG signals. Taking into account a large number of possible movements that characterize a dexterous hand, the chances of error occurrence are high. The minimization of this error occurrence constitutes the current challenge in devising the person-prosthesis interface. The experiments on the recognition of the movement intended by a person with an amputated hand can be replaced with the analysis of the relationship that holds between the movements of the healthy hand and the myopotentials that occur then, taking into consideration the limitations that arise due to the lack of some of the muscles in the amputated hand.

The solution presented here consists in the decomposition of the recognition, which leads to the multi-stage process that can be presented as the decision tree (DT).

In order to reduce the occurrence of recognition errors irrespective of the number of the processed data we have opted for replacing the usual decision nodes with Neural Networks of the multilayer perceptron (MLP) type (enodes). In order to create an MLP population, we have made use of the Genetic Algorithm (GA), which enhanced the synthesis of optimal MLPs.

For the needs of the research "EMG Analysis" application has been developed, which allows for the program transformation of the registered signals. Its operation is described in detail in subsequent chapters. Initially the EMG signal in the form of numerical data is transmitted to the SQL database. Signal acquisition and the manner in which the accumulated data are being refined are described in Chapter two. Chapter three describes methods of EMG signal analysis; extraction of characteristic features and concept for signal classification. Chapter four describes the experiment that has been carried out and sums up the results.

2 EMG SIGNALS ACQUISITION

In order to classify the EMG signal (or to determine the class of the intended movement) one has to make a number of transformations of the obtained signal, which boils down to its multistage analysis. Each of the said stages affects the quality of the whole process that is the reliability of movement recognition. That is why it is so important for each of the stages to be optimised as to their reliability and calculability. Main stages of the process that are being carried out by "EMG Analysis" application are presented below.

2.1 Signal Acquisition

In order to obtain the data the EMG signal was measured by means of 8-channel measuring system (Krysztoforski and Wolczowski, 2005). The system is made up of: (a) – computer that register the results of the measurements; it is fitted with (b) - A/D converters card (Ni 4477); and (c) - 8 active differential electrodes. The electrodes are fitted with two metal contacts (De Luca, 2006) that are directly integrated with the circuit amplifying the measured signals (potentials difference).

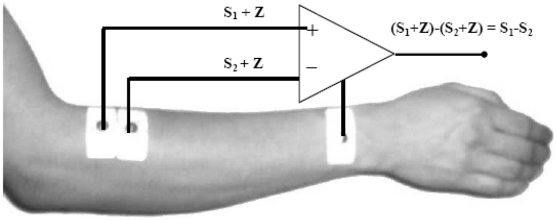


Figure 1: The differential measurement method. The EMG signal is represented by 's' and the noise signals by 'z'.

That kind of construction makes it possible to eliminate most of the disturbances that occur during the measurement. The value of the reduction is determined by coefficient CMRR of the amplifier. In this system the coefficient amounts to 84dB.

The signals from the electrodes are transformed into a digital form that is convenient for further analysis on the A/D converters card and they are stored in the SQL database of the measuring computer. These data are labelled for their further identification (measurement number, class of problem – i.e. posture). Data acquisition carried out in this way enables further program transformations of the obtained EMG signal by means of consecutive application modules of "EMG Analysis".

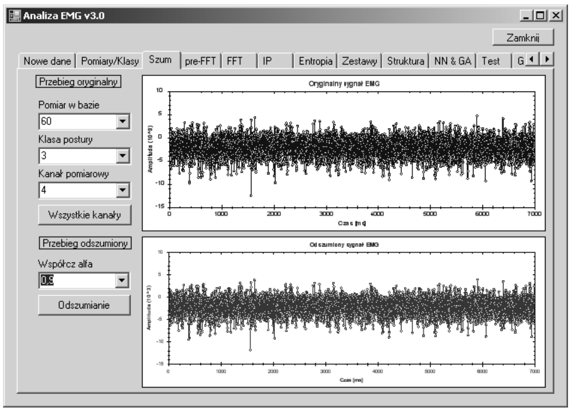
2.2 Preparing the Data for the Analysis

In order to reduce the level of noise in the obtained EMG signal, the signal undergoes an initial processing that consists in averaging of the signal course in the domain of time, independently of each channel (1), according to the rule:

$$y[n] = \alpha * x[n] + (1-\alpha)y[n-1] \tag{1}$$

y[n] - current output sample from averaging system;
x[n] - current EMG sample.

The selection window placed in the form shown in Figure 2 marked as "alpha coefficient" enables the operator to select a proper degree of the signal averaging depending on its quality ($\alpha \in \langle 0.5, 1.0 \rangle$).



Pomiar w bazie: 60
Klasa postury: 3
Kanał pomiarowy: 4
Współcz alfa: 0.9

Figure 2: Actual signal (upper part of the figure), the filtered signal. The alpha parameter equals 0,9.

The signal that has been processed as described above may be used for identifying characteristic features i.e. such features that differentiate the analysed signals one from another (in consequence the type of the hand movement).

3 EMG SIGNAL ANALYSIS

The process of EMG signal recognition is made up of two main stages: that of determining the features of the measured signal and that of signal classification on the basis of the determined features (Wolczowski and Suchodolski, 2007).

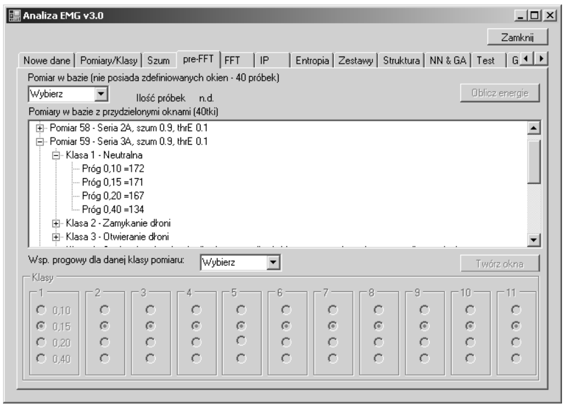
3.1 Features Extraction

In order to determine characteristic features of the signal two methods have been applied: Fast Fourier Transform and Wavelet Transform.

To carry the task out, the signal amplitude had to be analysed (Wojtczak, et al., 2008). A low-amplitude signal is difficult to classify due to the participation of the constant constituent (in the form of noise) and also due to the fact that the low value of the signal amplitude may imply both that the muscle (the hand) is at rest or its insignificant participation in a given class of movement. The analysis has been carried out according to the following scheme:

- Division of the input data vector that consists of 7.000 samples/channel (7s measurement) into 40 element windows (40 ms), and calculation of the sum of the contents each window (E_{sum}). In this way the information of the energy of the recorded movement is obtained; the minimum E_{min} and the maximum E_{max} energy values in the set;
- Finding out the energy threshold E_{edg} (activateing energy) in accordance with rule (2). In this way a group of windows is singled out that may contain important information in relation to the signal for the calculations that are being carried out (level of the signal amplitude);
- The Windows that fulfil the energy condition $E_{egd} > E_{sum}$ are then used to single out 256 element-windows (necessary for the spectrum analysis of the signal). This is done through the enlargement of the space of the selected 40-element windows by a further 216 elements (216 ms).

$$E_{egd} = E_{\min} + k * \left(E_{\max} - E_{\min}\right) \tag{2}$$



Pomiar w bazie (nie posiada zdefiniowanych okien - 40 próbek)
Wybierz
Ilość próbek n.d.

Pomiary w bazie z przydzielonymi oknami (40tki)

- Pomiar 58 - Seria 2A, szum 0.9, thrE 0.1
- Pomiar 59 - Seria 3A, szum 0.9, thrE 0.1
  - Klasa 1 - Neutralna
    - Próg 0,10 =172
    - Próg 0,15 =171
    - Próg 0,20 =167
    - Próg 0,40 =134
  - Klasa 2 - Zamykanie dłoni
  - Klasa 3 - Otwieranie dłoni

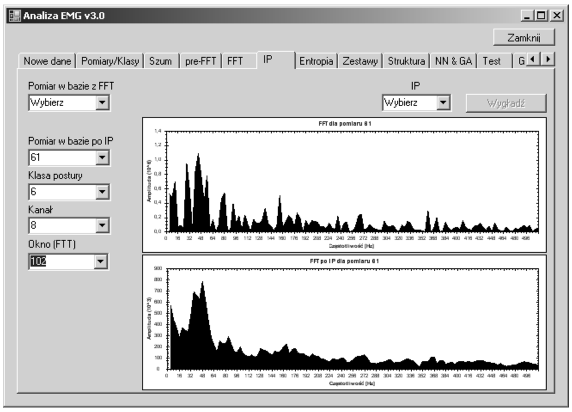
Wsp. progowy dla danej klasy pomiaru: Wybierz

	1	2	3	4	5	6	7	8	9	10	11
0,10	○	○	○	○	○	○	○	○	○	○	○
0,15	●	●	●	●	●	●	●	●	●	●	●
0,20	○	○	○	○	○	○	○	○	○	○	○
0,40	○	○	○	○	○	○	○	○	○	○	○

Figure 3: Signal usefulness analysis.

This stage is represented in Figure 3. In order to optimize this process, the application carries out a simulation of calculations simultaneously for a few values of the coefficient k ($k \in (0.1, \ldots 0.4)$). The operator can for each movement class select by means of the checkbox different values of his coefficient, which affects the number of the newly created 256-element windows.

The Windows determined in such a manner become the input vector for the feature extractor. As a result of the transformations for Fourier Transform we obtain the representation of the signal in the frequency domain (Figure 4).



Pomiar w bazie z FFT: Wybierz
IP: Wybierz
Pomiar w bazie po IP: 61
Klasa postury: 6
Kanał: 8
Okno (FTT): 102

Figure 4: Exemplary FFT course (before and after averaging).

The spectrum pictures of the signal thus obtained are further subjected to the process of averaging of harmonics for the purpose of enhancing the degree of resemblance (data generalization) according to (3). The averaged values thus become the vector of signal features.

$$S[k] = \sum_{i=IPS}^{IPE} \frac{Y[i]}{IPE - IPS + 1}, \quad k = 0,1,\ldots,N-1 \tag{3}$$

In order to determine the features by means of Wavelet Transform, Mallat algorithm has been used (Mallat, 1989), which is used to obtain, among others, the vector of signal details which, after transformation is treated as the feature vector. The signal is filtered in accordance with:

$$c_m(n) = \sum_k l^*_{2n-k} c_{m-1}(\kappa) \tag{4}$$

$$d_m(n) = \sum_k h^*_{2n-k} c_{m-1}(\kappa) \tag{5}$$

According to the size of the windows (256 elements) that constitute the input vector that undergoes transformations, 6 levels of signal filtration had to be applied; each of the Mallat levels reduces by half the size of the input vector. Each of the Mallat levels is represented by number values that are the sum of absolute magnitudes of the detail vectors that arose earlier. The number of the features that thus arose (6 features) is smaller than the number of the assumed movement classes to be recognised. Thus a further 6 features had to be created in order for them to represent the dynamics of the changes in the recorded movement. This was done by summing up the real values respectively for each of the Mallat vectors that arose. In this way it was possible to obtain 12 values that represent the information contained in the signal as well as their approximations at each of the obtained levels of filtration.

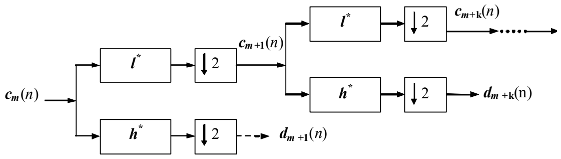
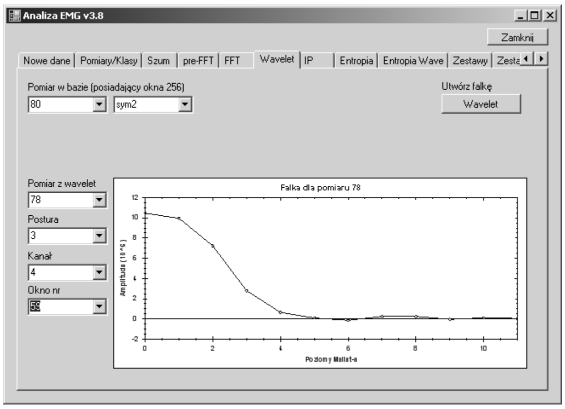


Figure 5: Signal filtration in compliance with Mallat algorithm.



Pomiar w bazie (posiadający okna 256): 80, sym2

Pomiar z wavelet: 78

Postura: 3

Kanał: 4

Okno nr: 59

Figure 6: Exemplary Wavelet form.

3.2 Data Classification

The classification of the feature vector is of great importance in the presented approach. Due to a large number of movement classes and high recognition reliability (which is indispensable for the dexterity of the prosthesis) the classification process is a difficult problem. That is why, for convenience's sake, the classification process should be performed in an extensive manner – through many stages. In the presented research the method of multi-stage problem solution has been proposed, which is performed through the decision tree (DT). The structure of DT calls for: (a) the determination of its decision logic (structures and the assignment of classes to the terminal nodes); (b) determination of the subsets of features for indirect nodes; (c) determination of the decision making rules (algorithms of recognition in each indirect node). To build the DT based on Neural Network Tree (NNTree) solution (Zhao, 2001) the C4.5 algorithm was used, the one worked out by Quinlan (1993); it makes use of the entropy of information in the learning subset. Entropy in information theory is defined as the average amount of information that corresponds to a sign that symbolises an occurrence out of a certain set. The selection of the attribute is determined on the basis of the increase of the information.

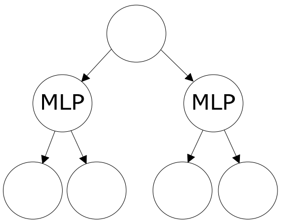
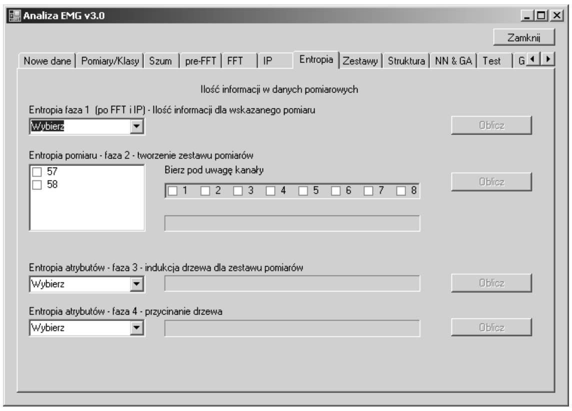


Figure 7: Exemplary NNTree.

The input vector that enables tree induction is the constituents: Fourier Transform (128 size vectors) or Wavelet Transform (12 size vectors) previously determined by information entropy of the set.

At this stage (Figure 8) it is possible to determine the data size that is to be transformed (single measurement, measurement grouping; specification of relevant signals).



Ilość informacji w danych pomiarowych

Entropia faza 1 (po FFT i IP) - Ilość informacji dla wskazanego pomiaru: Wybierz

Entropia pomiaru - faza 2 - tworzenie zestawu pomiarów: □ 57 □ 58

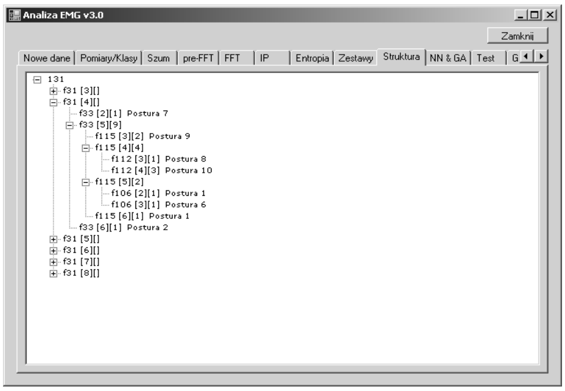
Bierz pod uwagę kanały: □ 1 □ 2 □ 3 □ 4 □ 5 □ 6 □ 7 □ 8

Entropia atrybutów - faza 3 - indukcja drzewa dla zestawu pomiarów: Wybierz

Entropia atrybutów - faza 4 - przycinanie drzewa: Wybierz

Figure 8: Entropy tab page.

Quinlan's method divides a given node into as many descending nodes, as many values there are in the most informative feature (constituent of the selected transform). The tree structure is made by recurring computation of the information increase coefficient for all available attributes at each recurring selection of the algorithm and the selection of the attribute for which this value is maximal (constituent of the selected transform). The recurring selection of the algorithm that makes up the tree structure ends the searching process of the input vector (FFT or DWT feature vector) provided that in a given sub-tree created by means of the recurring procedure a set of attributes represents one decision class. In this case a terminal mode is created that is marked with class label. The structure thus created has an input node, a root, and descendant nodes some of which (terminals) determine the decision class.



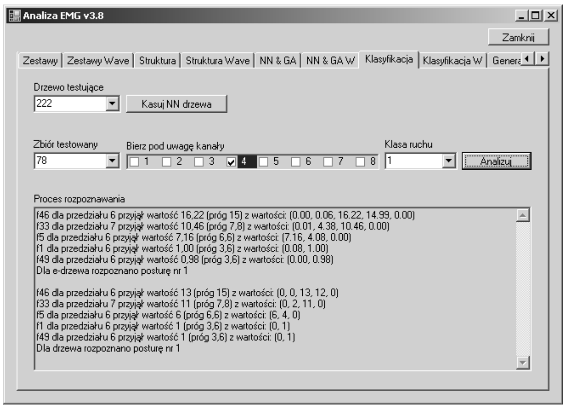
```
⊟ 131
    ⊞ f31 [3][]
    ⊟ f31 [4][]
        f33 [2][1]  Postura 7
        ⊟ f33 [5][9]
            f115 [3][2]  Postura 9
            ⊟ f115 [4][4]
                f112 [3][1]  Postura 8
                f112 [4][3]  Postura 10
            ⊟ f115 [5][2]
                f106 [2][1]  Postura 1
                f106 [3][1]  Postura 6
            f115 [6][1]  Postura 1
        f33 [6][1]  Postura 2
    ⊞ f31 [5][]
    ⊞ f31 [6][]
    ⊞ f31 [7][]
    ⊞ f31 [8][]
```

Figure 9: Exemplary decision tree.

To avoid too large a tree (and excessive matching), the pruning was used on the basis of the estimation of the significance of the difference of the classify-cation error for a given node and its sub-nodes. The algorithm estimates the probability of the decrease of the error occurrence and cuts off the divisions for which this probability does not exceed a given threshold value or provides identical results (similar results for the sub-nodes). The boundaries of the ranges that divide the space of the input vector (FFT or DWT) into the scopes of values that can be obtained were selected as a result of experiments so that their distribution was even for the whole span of the input. However, this method has a drawback; namely, new feature vectors that are to be classified can have values that cross the boundaries of two neighbouring scopes and represent different classes of hand movement. Therefore, the simple decision rules that consisted in comparing the new value with existing scopes were replaced with the decision making module i.e. MLP. On the basis of actual values from the database, *n* networks were trained (*n* – number of scopes in the MLP database {m_1...m_n}). Each of them (MLP_1..MLP_n) is responsible for the classification of the value from a given range (m_1...m_n). This solution introduces the genetic algorithm to create the MLP population that will allow for solving the problem of range classification.

The process is running until the intended classification error is obtained (the value falls within the range) or until the maximum iteration number is obtained (then the best configuration is returned – the least obtained error).

After the training process, the configuration and the weights of best MLP modules are directed to SQL database where they are assigned to given value ranges to be further used for classification process. The process consists in computing the attributes taken earlier from the input vector (EMG spectrum) in compliance with the decision scheme i.e. the tree structure. The decision rules are taken from the truth table. Additionally, the taken value is computed by means of relevant MLP_n module that simulates a given tree node (Figure 10). Thus we obtain the classic and neuron tree response. The activation of one of the descendant nodes brings about the recurring selection of another rule. The process is running until the final node (terminal) is reached. This node determines the decision class i.e. the type of identified posture.



Drzewo testujące: 222

Zbiór testowany: 78

Bierz pod uwagę kanały: □ 1 □ 2 □ 3 ☑ 4 □ 5 □ 6 □ 7 □ 8

Klasa ruchu: 1

Proces rozpoznawania

```
f46 dla przedziału 6 przyjął wartość 16,22 (próg 15) z wartości: (0.00, 0.06, 16.22, 14.99, 0.00)
f33 dla przedziału 7 przyjął wartość 10,46 (próg 7,8) z wartości: (0.01, 4.38, 10.46, 0.00)
f5 dla przedziału 6 przyjął wartość 7,16 (próg 6,6) z wartości: (7.16, 4.08, 0.00)
f1 dla przedziału 6 przyjął wartość 1,00 (próg 3,6) z wartości: (0.08, 1.00)
f49 dla przedziału 6 przyjął wartość 0,98 (próg 3,6) z wartości: (0.00, 0.98)
Dla e-drzewa rozpoznano posturę nr 1

f46 dla przedziału 6 przyjął wartość 13 (próg 15) z wartości: (0, 0, 13, 12, 0)
f33 dla przedziału 7 przyjął wartość 11 (próg 7,8) z wartości: (0, 2, 11, 0)
f5 dla przedziału 6 przyjął wartość 6 (próg 6,6) z wartości: (6, 4, 0)
f1 dla przedziału 6 przyjął wartość 1 (próg 3,6) z wartości: (0, 1)
f49 dla przedziału 6 przyjął wartość 1 (próg 3,6) z wartości: (0, 1)
Dla drzewa rozpoznano posturę nr 1
```

Figure 10: Classification process of the recorded signal.

4 EXPERIMENT AND OBTAINED RESULTS

The experiments were carried out on healthy persons by placing each time 6 electrodes on the skin of the forearm over relevant muscles.

Eleven movement classes were tested: 1. neutral; 2(3). flexion /extension of the hand digits; 4(5) pronation and supination of the forearm; 6(7) flexion / extension of the wrist; 8(9). flexion/ extension of the thumb; 10(11). flexion / extension of the fingers II-V.

For each class of movement the measurement lasted 7s and was preceded with a 5s break. In that way the discreet signals were obtained of the size 7.000 samples * 6 channels) per class (sampling

Table 1: Averaged classification errors of the tested movements for FFT input vector (Err$_1$-DT, Err$_2$-NNTree).

Class	1	2	3	4	5	6	7	8	9	10	11
Err$_1$ [%]	13	16	14	19	17	12	18	23	19	17	19
Err$_2$ [%]	9	14	14	17	9	10	13	21	12	11	14

Table 2: Averaged classification errors of the tested movements for DWT input vector (Err$_1$-DT, Err$_2$-NNTree).

Class	1	2	3	4	5	6	7	8	9	10	11
Err1 [%]	12	14	19	17	12	13	15	18	17	18	15
Err2 [%]	6	11	11	13	13	7	14	16	10	8	8

frequency 1 kHz). The data were located in the SQL database. The measurement data (1/3 of the signals selected at random) were used to generate the decision tree and MLPs. The remaining part of the set (2/3 of the signals selected at random) was assigned as a testing set.

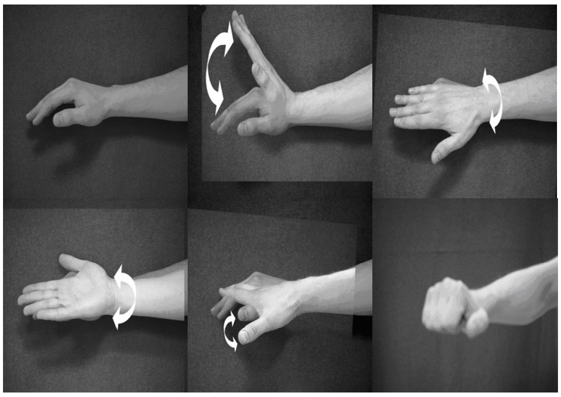


Figure 11: Tested movement classes.

The NNTree generated in this way gave an average recognition value equal to 87% (FFT) when the DT with classical nodes in turn gave a value equal to 83% (FFT). The detail results for tested movements are presented in Table1; were Err1 and Err2 represent the errors obtained respectively by DT and by NNTree methods for the FFT feature vectors. As we can see the best results were for classes 1, 5, 6 and 10.

In the case of DWT feature vectors an average recognition values were equal to 89% for NNTree and 85% for DT method. I this case the best results were for classes 1, 6, 10 and 11 (see: Table 2).

5 CONCLUSIONS

The tests with the presented “EMG Analysis” software are in the incipient stage. The conducted experiments aimed on principle at comparing the features extraction methods for EMG signals and their further classification. They prove the usefulness of the presented multistage analysis methods for the recognition of the decisions that control the manipulation and grasping movements of the artificial hand.

REFERENCES

Wolczowski A., 2001. Smart hand: the concept of sensor based control. In *MMAR’01, 7th IEEE International Conference on Methods and Models in Automation and Robotics*. Szczecin Univ. of Technol. Publishing.

Su Y., Fisher M. H., Wolczowski A., Bell D. G., Burn D. J., Gao R. X., 2007. Towards and EMG-controlled prosthetic hand using a 3-D electromagnetic positioning system. *IEEE Transactions on Instrumentation and Measurement*, (56/1).

Mohammadreza A. O., 2007. Myoelectric control system – A survey. *Biomedical Signal Processing and Control,* vol. 2, issue 4, pp 275-294.

Krysztoforski, K., Wolczowski, A., 2005. *Progress of Robotics - Industrial and medical robotic systems*, A laboratory stand for EMG signal measurements. WKL. Warszawa.

De Luca, C. J., 2006. *Electromyography. Encyclopedia of Medical Devices and Instrumentation*. (John G. Webster, Ed.) John Wiley Publisher.

Wolczowski, A., Suchodolski T., 2007. Bioprosthesis Control: Human-Machine Interaction Problem. In *Challenges for Assistive Technology - AAATE 07*. IOS Press.

Wojtczak P., Amaral T. G., Dias O. P., Wolczowski A., Kurzynski M., 2008. Hand movement recognition based on biosignal analysis. *Engineering Applications of Artificial Intelligence*, (22/(4/5)).

Mallat, S., 1989. A Theory for Multiresolution Signal Decomposition*, IEEE Trans. Pattern Analysis and Machine Intelligence*, (11).

Zhao, Q. F., 2001. Evolutionary design of neural network tree – integration of decision tree, neural network and GA. In *IEEE Congress on Evolutionary Computation*.

Quinlan, J. R., 1993. C4.5: *Programs for Machine Learning*, Morgan Kaufmann. San Mateo, California.

REGISTRATION OF INDOOR 3D RANGE IMAGES USING VIRTUAL 2D SCANS

Marco Langerwisch and Bernardo Wagner
Institute for Systems Engineering, Real Time Systems Group
Leibniz Universität Hannover, Appelstraße 9A, D-30167 Hannover, Germany
{langerwisch, wagner}@rts.uni-hannover.de

Keywords: 3D laser range finders, Scan matching, Point cloud registration, Iterative closest point algorithm, Virtual 2D scans, Real time, Mobile robots.

Abstract: For mapping purposes, autonomous robots have to be capable to register 3D range images taken by 3D laser sensors into a common coordinate system. One approach is the Iterative Closest Point (ICP) algorithm. Due to its high computational costs, the ICP algorithm is not suitable for registering 3D range images online. This paper presents a novel approach for registering indoor 3D range images using orthogonal virtual 2D scans, utilizing the typical structure of indoor environments. The 3D registration process is split into three 2D registration steps, and hence the computational effort is reduced. First experiments show that the approach is capable of registering 3D range images much more efficient than ICP algorithm and in real time.

1 INTRODUCTION

Autonomous robots dealing with localization and navigation often do not have a map of their environment available in advance. In this case, the robot has to build a map on its own, based on some perceptual sensing. While conventional approaches use 2 dimensional (2D) mapping (Thrun, 2003), where three degrees of freedom (3DoF) have to be considered, recent approaches use 3 dimensions (3D) with its six degrees of freedom (6DoF), i.e. the x, y and z coordinates, and the roll, pitch, and yaw angles.

One approach used here for sensing the environment in three dimensions are 3D laser range finders. To build a map out of these captured single 3D range images, the rotation and translation have to be calculated to assemble these images into a common global coordinate system.

Lots of research work has been done in mapping two 3D range images into a common coordinate system. One approach is to use the Iterative Closest Point (ICP) algorithm. It calculates the point correspondences between the two images minimizing a mean square error function. Disadvantages of this approach are the computational costs due to the high number of 3D points, and therefore the execution time of the registration. For example, research in accelerating the ICP algorithm for 3D point mapping is done by (Nüchter et al., 2007). They use the algorithm to register 3D range images taken by a robot equipped with a tiltable 2D SICK laser range finder (Nüchter et al., 2005), trying to accelerate the 3D registration by using different representations of the 3D scans, and point reduction during the ICP algorithm.

Based on the typical orthogonality found in indoor environments (Nguyen et al., 2007) presents a lightwight localization and mapping algorithm. The main idea is to reduce complexity by mapping only planes that are parallel or perpendicular to each other. Because objects with different shapes are ignored, the resulting map does not contain any of these features.

A very similar approach to the contribution of our paper is presented in (Jez, 2008). The author first extracts leveled maps of the 3D range images. This is done by aligning the range images with the horizontal plane and then extracting points of vertical structures. These leveled maps are passed to a 2D ICP algorithm to calculate rotation and translation in 3DoF which are applied to the 3D range images. To obtain the final 6DoF transformation, the transformed range images are passed to a full 3D ICP algorithm. Drawbacks are the inability to deal with dynamic obstacles in this approach, and the full ICP at the end, that has high computational costs again.

In contrast, our aim is to avoid a full 3D ICP computation. Moreover, our approach is able to cope with dynamic objects and non-rectangular shaped indoor structures like round columns. It accelerates the range image registration in indoor enviroments compared to 6DoF ICP to let the robot register range images online. The structure of indoor environments is used to extract virtual 2D scans with a much lower number of points. Having three orthogonal virtual 2D scans, each of these can be registered very fast via 2D ICP to calculate rotation and translation. Finally, we combine the 2D transformations to get full 3D rotation and translation. First experiments show that with our approach, a mobile robot becomes capable to register 3D range images in real time with similar quality to 3D ICP registration.

The following section outlines the ICP algorithm and some needed notations, while section 3 describes the basic concepts of virtual 2D scans. Our new approach itself is presented in section 4, and section 5 shows experimental results that we obtained on one of our service robots. This paper closes with a conclusion and an outlook on future work.

2 ITERATIVE CLOSEST POINT ALGORITHM

To merge two images that have at least partial coverage into one common coordinate system, transformation (R,t) consisting of translation vector t and rotation matrix R have to be calculated. This process is called registration. One approach to register two images is the Iterative Closest Point (ICP) algorithm invented by (Besl and McKay, 1992), and (Chen and Medioni, 1991). It calculates the point correspondences between the two images while trying to minimize a mean square error function. An initial pose estimate is needed, so the ICP is a local registration algorithm (Magnusson et al., 2009).

The ICP algorithm registers a range image, given by a data point set $D=(d_1,...,d_{N_D})$, into the coordinate system of a range image, given by model point set $M=(m_1,...,m_{N_M})$, to minimize the error function

$$e(R,t)=\sum_{i=1}^{N_M}\sum_{j=1}^{N_D} w_{i,j} \parallel m_i-(Rd_j+t) \parallel^2 \qquad (1)$$

with $w_{i,j}$ assigned 1 if m_i is the closest point for d_j, otherwise 0.

$$N=\sum_{i=1}^{N_M}\sum_{j=1}^{N_D} w_{i,j} \qquad (2)$$

is the number of points taken into account for its calculation.

The main challenges here are how to compute the closest points and the transformation (R,t) efficiently. In our implementation we use a k-d-tree, a generalization of binary search trees, as point representation for finding the closest point correspondences (Bentley, 1975), and an algorithm based on singular value decomposition (SVD) (Arun et al., 1987) to compute the transformation (R,t).

3 VIRTUAL 2D SCANS

Calculating with full 3D raw point data sets is very computational demanding. The aim of virtual 2D scans is to reduce these 3D point sets to 2D point sets (Wulf et al., 2004). Virtual 2D scans are calculated regarding a special purpose, e.g. localization or navigation. For this special purpose, the raw 3D range images include many redundant and unnecessary information, and points respectively. Extracting points without losing relevant information and projecting them at a 2D plane reduces subsequent computational costs, because 2D algorithms can be used with a relative small number of points instead of 3D algorithms.

According to (Wulf, 2008), virtual 2D scans are calculated in two steps. First, reduce the amount of points of the 3D point set S:

$$f: S \to S_r, \quad \text{having } S_r \subset S. \qquad (3)$$

The size of the new point set S_r is $l \ll |S|$. Function f depends on the special purpose of the virtual 2D scan and will be defined in the next section.

In the second step the virtual 2D scan V_S is calculated from the reduced point set S_r by

$$g:(x,y,z) \to (x,y), \quad x,y,z \in \mathbb{R}, \qquad (4)$$

where x, y and z are the x, y and z coordinates of a 3D point. Equation 4 projects the 3D points of the reduced point set S_r on the 2D plane.

Provided that (3) fits the purpose of this reduction, all relevant information are kept in the virtual 2D scan V_S, and subsequent algorithms will be computationally less demanding.

4 3D RANGE IMAGE REGISTRATION

The following subsection reveals the basic concept underlying our approach, while in section 4.2 the actual algorithm we used for registering 3D range images via virtual 2D scans is described. The special virtual 2D scans used in our algorithm are presented in section 4.3.

4.1 Splitting of 6DoF Transformations

The basic idea behind our approach is that a 6DoF transformation can be split up into several transformations, each having less than six degrees of freedom. Each single transformation can then be computed by an ICP algorithm with much higher efficiency than full 6DoF ICP, resulting in a smaller total computation time. The single transformations are combined subsequently to gain the full 6DoF transformation.

Our approach is to split the 6DoF transformation into three 2D transformations that can each be calculated by a 2D ICP algorithm. For calculating the transformations, we extract three orthogonal virtual 2D scans. In our case, these virtual 2D scans are aligned along the x-y-plane, the x-z-plane, and the y-z-plane respectively. Using virtual 2D scans utilizes the advantages described in the previous section. Virtual 2D scans can cope with every shape in indoor environments, round columns for example. They can be defined insensitive to dynamic objects.

4.2 Registration Algorithm

It is assumed that the x-axis points forwards from the robot, the y-axis to the right and the z-axis downwards. The functions according to (3) of the virtual 2D scans will be described in the next subsection. Moreover, we assume that the scans to be registered are coarsely aligned by an initial pose estimate, e.g. by odometry or a gyrometer on a moving platform. Hence, the approach presented is a local registration algorithm just as the ICP. The 3D data point set D is to be registered into the coordinate system of the 3D model point set M.

First, a virtual 2D scan of the x-y-plane is extracted from both 3D point sets. Then, we compute a 2D ICP algorithm as described in section 2. We get the 2D rotation matrix R^{xy} and the 2D translation vector t^{xy}. Apply the rotation matrix

$$R' = \begin{pmatrix} R^{xy} & 0 \\ 0 & 1 \end{pmatrix} \tag{5}$$

and the translation vector

$$t' = \begin{pmatrix} t_1^{xy} & t_2^{xy} & 0 \end{pmatrix}^T \tag{6}$$

to the data point set D.
Next, extract virtual 2D scans of the x-z-plane from M and D'. Compute a 2D ICP algorithm as described in section 2. We get the 2D rotation matrix R^{xz} and the 2D translation vector t^{xz}. Calculate rotation matrix

$$R'' = \begin{pmatrix} R_{11}^{xz} & 0 & R_{12}^{xz} \\ 0 & 1 & 0 \\ R_{21}^{xz} & 0 & R_{22}^{xz} \end{pmatrix}. \tag{7}$$

Because we translated in x- and y-direction in the previous step, we just use the translation in z-direction:

$$t'' = \begin{pmatrix} 0 & 0 & t_2^{xz} \end{pmatrix}^T \tag{8}$$

Apply R'' and t'' to the data point set D' transformed in the previous step.
Finally, extract virtual 2D scans of the y-z-plane from M and D'' and compute a 2D ICP algorithm as described in section 2. We get the 2D rotation matrix R^{yz} and the 2D translation vector t^{yz}. Calculate rotation matrix

$$R''' = \begin{pmatrix} 1 & 0 \\ 0 & R^{yz} \end{pmatrix}. \tag{9}$$

Because translation in all three directions has been calculated before, no more translation is applied in this step.
The global rotation matrix calculates to

$$R = R''' \cdot R'' \cdot R', \tag{10}$$

and the global translation vector calculates to

$$t = R''' \cdot R'' \cdot t' + R''' \cdot t''. \tag{11}$$

Finally, the registered data point set is

$$D_{reg} = R \cdot D + t. \tag{12}$$

A limitation of this algorithm is the assumption of small errors in the initial pose estimate. Having larger errors, the convergence of each single step to an optimal solution does not guarantee global optimality. Other variations of this algorithm are subject of current research, for example the introduction of iteration steps. Nevertheless, the previous algorithm has shown promising results even in case of errors in initial alignment (see section 5).

4.3 Virtual 2D Scans for Registration

Three virtual 2D scans are needed, one for the x-y-plane, one for the x-z-plane and one for the y-z-plane. The calculation is based on the assumption that the 3D range image is given by a point cloud having all scan points aligned in vertical columns and horizontal rows with reference to the center of the scanning device. Here, the assumption is met because we use a yawing 3D scan (Wulf et al., 2004). In case of other scanning devices, the 3D points have to be reordered first.

For the x-y-plane, the indoor landmark scan from (Wulf, 2008) is taken. It extracts landmarks of cluttered indoor scenes like offices or corridors. Because of its robustness against dynamic objects and occluded walls and its ability to cope with non-rectangular shaped environments, it fits very well for our purposes. The idea is to extract one point per vertical column that has the largest radial distance to the scan root. Function f of (3) is defined accordingly. In indoor environments, we get a kind of ground plan because usually the point with the largest radial distance lies on a wall. If a wall is discontinuous due to doors or gangways, the following wall is extracted to the virtual 2D scan.

When computing the virtual 2D scan for the x-z- and the y-z-planes, the 3D scans have already been rotated around the z-axis and translated in x- and y-direction (see section 4.2). Hence we need to calculate the rotation around the x- and y-axis, and the translation in z-direction using virtual 2D scans. Our approach puts planes orthogonal to x- and y-axis respectively into the coordinate system. Points located near those planes are extracted to virtual 2D scans.

For the virtual 2D scan for the x-z-plane, we compute S_r (see (3)) as follows:
p is a 3D point composed of the coordinate-tuple (p_x, p_y, p_z), and d is a distance threshold.
For each horizontal row in S look for the closest points p' and p'' to the x-z-plane, having $p'_x >= 0$, and $p''_x < 0$ respectively. If p'_y is lower than d, $S_r = S_r \cup p'$. If p''_y is lower than d, $S_r = S_r \cup p''$.
Equation 4 has to be modified to

$$g : (x, y, z) \to (x, z), \quad x, y, z \in \mathbb{R}. \qquad (13)$$

Finally, only rotation around x-axis is left that needs to be calculated by the virtual 2D scan for the y-z-plane (see section 4.2). The subset S_r of the last virtual 2D scan is calculated similar to above with respect to the y-z- instead of the x-z-plane.

Unfortunately, the virtual 2D scans for the x-z- and the y-z-plane are sensitive to dynamic objects located in a distance closer than d to the x-z-plane, and y-z-plane respectively. A possible solution could be virtual 2D scans similar to that of the x-y-plane. This is subject of current research, as well as other implementations of the virtual 2D scans capable of dealing with outdoor environments.

5 EXPERIMENTAL RESULTS

This section shows the experimental results of our approach under real time constraints. At first, we will introduce the experimental setup, while the qualitative and computational results of two experiments will be presented in subsections 5.2, 5.3, and 5.4. This section closes with an evaluation of two series of experiments registering range images with simulated error.

5.1 Experimental Setup

The 3D range images were taken by a rotating 3D laser scanner based on SICK's LMS291. An introduction to the scanning device is given in (Wulf et al., 2007). Several 3D scans were taken from different positions on a typical office floor with offices and doors on both sides.

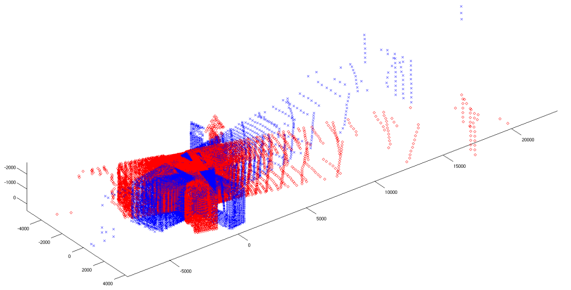


Figure 1: Model 3D scan (blue crosses), data 3D scan (red diamonds), not registered.

Table 1: Results of first experiment.

	Unreg	ICP	RegVirt
$e(R,t)$	29752	3369	3325
N	20483	25877	25870
Time [ms]	-	74050	67

For registering two 3D point clouds we used an embedded PC running our robotic framework RACK on a real time linux operating system. For comparison of the algorithms we used an accelerated ICP implementation, proposed e. g. in (Nüchter et al., 2007), as described in section 2. We conducted our tests by executing both algorithms with identical input data.

5.2 Qualitative Results

The scans we picked for the first evaluation consist of 25858 and 25987 raw 3D points respectively. Both scans in their local coordinate systems are shown in Figure 1. When executing our registration algorithm using virtual 2D scans, three 2D registration steps have to be performed. After 25 iterations of the 2D ICP, the scans were registered successfully in the first step (x-y-plane). The second and third registration steps took 14 and 4 iterations respectively.

For qualitative comparison of our new approach and the 3D ICP we calculated the error function $e(R,t)$ according to (1) before and after full 3D registration. Table 1 shows the results of the error function and the number of points N according to (2) taken into account for its calculation. The translation error between the results of both algorithms is 22 mm.

As can be seen, there are no qualitative differences worth mentioning. The error function reaches a similar value, as well as the number of points taken into account for calculating the error function. This leads to the result that our new approach reaches a similar

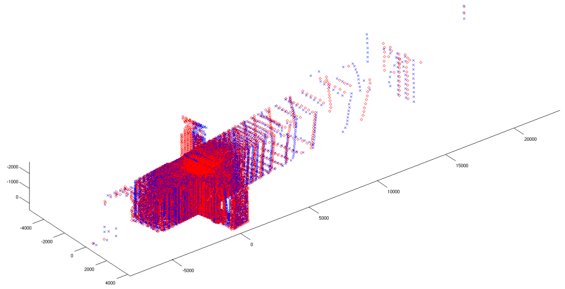


Figure 2: Model 3D scan (blue crosses), data 3D scan (red diamonds), registered.

Table 2: Results of second experiment.

	Unreg	ICP	RegVirt
$e(R,t)$	33908	5048	6376
N	23564	26131	26157
Time [ms]	-	36950	94

quality to a conventional ICP algorithm when registering two indoor 3D range images with small translation in z-direction and rotation around x- and y-axes.

Figure 2 shows both 3D point clouds registered into the model coordinate system by our new registration approach.

5.3 Non-zero Pitch and Roll Angles

To evaluate how well our approach copes with significantly non-zero pitch and roll angles, we tilted our experimental platform for appr. 4 cm to the top while taking the 3D data point set. Together with the yaw motion, this results in non-zero pitch and roll angles with reference to the coordinate system of the 3D model point set. Figure 3a shows the unregistered point clouds. The scans consist of 26088 and 26311 raw 3D points respectively. The remaining setup is the same as presented in section 5.1.

Table 2 shows the results of the error function and the number of points N according to (2) taken into account for its calculation. Computing the registration took 24, 18, and 9 iterations respectively with our new approach. The translation error between both results is 62 mm. Using a gyrometer on a moving platform could increase accuracy, i.e. decreasing the slight difference in the error function resulting from the coarse alignment before the registration. However, the minor difference in the quality can be neglected when achieving real-time computing capabilities in return, as presented in the next subsection.

Figure 3b shows both 3D point clouds registered into the model coordinate system by our new registration approach.

5.4 Computational Requirements

Because of the reduction from 3D to 2D ICP algorithm, our new approach needs much less computational effort for registering the 3D range images. As can be seen in Table 1, the calculation time reduces from appr. 74 seconds in the case of full 3D ICP to 67 milliseconds in the case of our new approach. In the second experiment, the reduction is from appr. 37 seconds to 94 milliseconds (see Table 2). When a full 3D scan takes 1.2 seconds as in our case (Wulf et al., 2007), our algorithm is capable of registering subsequent taken 3D range images online, i.e. in real time.

5.5 Simulated Error

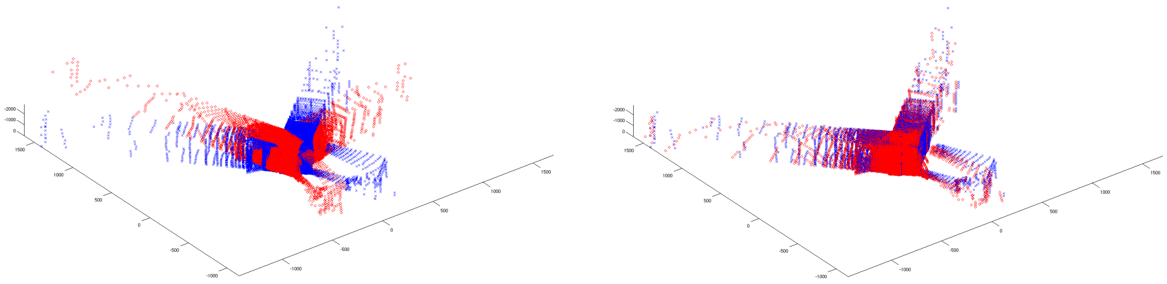
Finally, we evaluated two series of registration with simulated errors. Therefore, we transformed the model 3D scan of Figure 3a and registered it to the untransformed model scan. In a first series, we translated the scan in each direction, ranging from -30 cm to +30 cm, and taking incremental steps of 10 cm. This resulted in a series of 343 registrations for both algorithms. Surprisingly, our new approach had a mean translation error of 0.38 mm, compared to 0.49 mm using the ICP algorithm. The computation time was 25 ms (min. 17 ms, max. 31 ms) in average, compared to 19.7 s (min. 253 ms, max. 42.5 s).

In a second series, we rotated the scan around all axes, ranging from -10° to +10° each. The applied increment was 5°, resulting in a total of 125 registrations. The mean rotation error of our new approach was 0.86° compared to 0.01° in case of ICP. The average computation times were 55 ms (min. 16 ms, max. 113 ms), and 30.3 s (min. 254 ms, max. 56.6 s).

Examining these two experimental series with simulated errors, the computation times compared to the ICP algorithm were convincing again. The translation error with our new approach was even lower than with ICP. In case of rotation we evaluated a higher rotation error using our approach, but we are confident to gain an improvement with further research regarding the algorithm (see section 4.2).

6 CONCLUSIONS AND FUTURE WORK

This paper presents an approach for registering indoor 3D range images in real time. In our experiments, it registers two 3D point clouds up to 1100 times faster than the well-known ICP algorithm. This is done by splitting the full 6DoF transformation into three transformations. Each single transformation is calculated



(a) Unregistered 3D scans. (b) Registered 3D scans.

Figure 3: Registration of 3D scans with non-zero roll, pitch, and yaw angles.

by a 2D ICP algorithm with orthogonal virtual 2D scans, utilizing the typical structure of indoor environments. Due to the use of virtual 2D scans, the algorithm can cope with dynamic objects (with small restictions) as well as with non-rectangular environments. The resulting registration leads to a similar quality to the full 3D ICP algorithm.

Limitations and therefore future research work have been identified in sections 4.2 and 4.3. Nevertheless, our experiments showed promising results. A comparison to other approaches like techniques based on the extraction of planes out of the 3D range images and matching them will be necessary to stress the advantages of our algorithm. Further future work will deal with the implementation and evaluation of a full SLAM cycle based on the registration of 3D range images using virtual 2D scans.

REFERENCES

Arun, K. S., Huang, T. S., and Blostein, S. D. (1987). Least-squares fitting of two 3-D point sets. *IEEE Trans. Pattern Anal. Mach. Intell.*, 9(5):698–700.

Bentley, J. L. (1975). Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18(9):509–517.

Besl, P. and McKay, H. (1992). A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256.

Chen, Y. and Medioni, G. (1991). Object modeling by registration of multiple range images. In *Proc. IEEE International Conference on Robotics and Automation ICRA 1991*, pages 2724–2729 vol.3.

Jez, O. (2008). 3D mapping and localization using leveled map accelerated ICP. In Bruyninckx, H., Preucil, L., and Kulich, M., editors, *EUROS*, volume 44 of *Springer Tracts in Advanced Robotics*, pages 343–353. Springer.

Magnusson, M., Nüchter, A., Lorken, C., Lilienthal, A., and Hertzberg, J. (2009). Evaluation of 3d registration reliability and speed - a comparison of icp and ndt. In *Proc. IEEE International Conference on Robotics and Automation ICRA 2009*, pages 3907–3912.

Nguyen, V., Harati, A., and Siegwart, R. (2007). A lightweight SLAM algorithm using orthogonal planes for indoor mobile robotics. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems IROS 2007*, pages 658–663.

Nüchter, A., Lingemann, K., and Hertzberg, J. (2005). Mapping of rescue environments with Kurt3D. In *Proc. IEEE International Safety, Security and Rescue Robotics, Workshop*, pages 158–163.

Nüchter, A., Lingemann, K., Hertzberg, J., and Surmann, H. (2007). 6D SLAM - 3D mapping outdoor environments. *J. Field Robot.*, 24(8-9):699–722.

Thrun, S. (2003). *Robotic mapping: a survey*, pages 1–35. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Wulf, O. (2008). *Virtuelle 2D-Scans - Ein Verfahren zur echtzeitfähigen Roboternavigation mit dreidimensionalen Umgebungsdaten*. PhD thesis, Leibniz Universität Hannover, Hannover, Germany.

Wulf, O., Arras, K., Christensen, H., and Wagner, B. (2004). 2D mapping of cluttered indoor environments by means of 3D perception. In *Proc. IEEE International Conference on Robotics and Automation ICRA 2004*, volume 4, pages 4204–4209 Vol.4.

Wulf, O., Nüchter, A., Hertzberg, J., and Wagner, B. (2007). Ground truth evaluation of large urban 6D SLAM. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems IROS 2007*, pages 650–657.

POSTERS

OBSTACLES AVOIDANCE IN THE FRAME WORK OF PYTHAGOREAN HODOGRAPH BASED PATH PLANNING

M. A. Shah*, A. Tsourdos, P. M. G. Silson, D. James and N. Aouf
Dept. of Informatics and Sensors, Cranfield University, Cranfield, U.K.
**m.shah@cranfield.ac.uk*

Keywords: Pythagorean Hodograph, Bending energy, Path Planning, Feasible paths.

Abstract: This paper deals with the problem of obstacle avoidance in the path planning based on Pythagorean Hodograph. The proposed obstacle avoidance approach is based on changing the curvature in case of Pythagorean Hodograph curves. However this may result in an increase in the length of the path. Furthermore in some cases obstacle avoidance by changing the curvature of the path may result not only in an increase in path length but also to a tremendous increase in bending energy. An increase in bending energy of a path as a result of curvature change above certain limit makes the path very difficult or impossible to fly.

1 INTRODUCTION

In path planning for UAVs the obstacle avoidance is a common problem. In literature different researcher adopted different ways to solve the problem of obstacle avoidance depending on the context of the problem. (Madhavan at el 2006) and (H. Bruyninckx at el 1997) proposed a solution to avoid the obstacle in case of the Pythagorean Hodograph based path planning for UAVs by manipulating the curvature of the PH path. By changing the curvature of the path an obstacle can be avoided but the path length is increased tremendously as a result. For example figure 1and 2 illustrate this fact:

But sometimes situations arise where obstacles avoidance by changing the curvature of the path results in path of very high bending energy. A path with a very high bending energy is very difficult for UAVs to follow and hence not a feasible path. Figure 3 shows this fact.

Therefore curvature manipulation method can not be solely used for obstacle avoidance in PH based path planning because it is bound to fail for some cases. We need an alternative method for obstacle avoidance which can guarantee feasible (safe and flyable) paths. To elaborate such a method is the subject of this paper.

The rest of the paper is organized as follows: Section 2 describes the problem formulation. Section 3 introduces the Pythagorean Hodograph to generate the initial paths. Section 4 introduces the proposed solution to the problem. Section 5 discusses the simulation results and finally section 6 draws the conclusion.

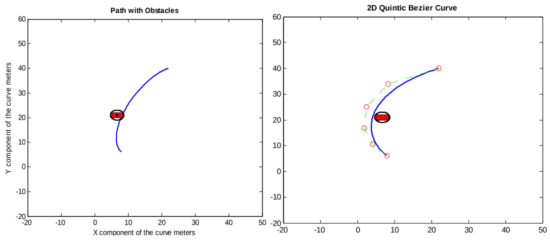


Figure 1: Obstacle avoidance by curvature change.

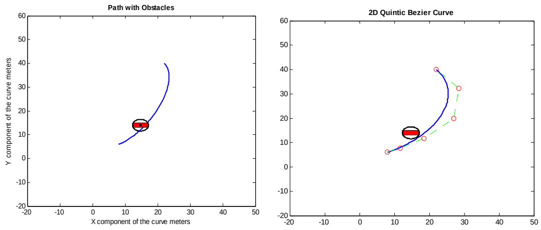


Figure 2: Obstacle avoidance by curvature change.

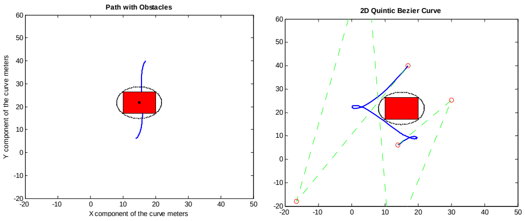


Figure 3: Obstacle avoidance by curvature change.

2 PROBLEM FORMULATION

A mission is planned to fly a group of unmanned aerial vehicles safely from base B to target T as shown in Figure 4. All vehicles start from the base at the same time. The environment contains stationary and moving obstacles. During the flight, the vehicles will avoid inter-collision and collision with the stationary and moving obstacles. Initially, UAV paths are planned offline by the path planning module (Path Planner) on the basis of the available knowledge about the environment. The UAVs start following these paths, if during the flight any of the UAVs comes across an obstacle, which was not known before then changes are made to the initial path of corresponding UAV to avoid the obstacles while maintaining UAVs cooperation.

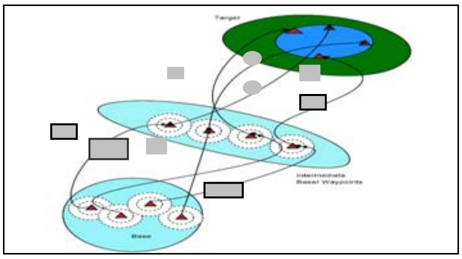


Figure 4: Swarm of UAVs Scenario.

The base and the goal points for each UAV are specified by initial and final poses. Let the starting pose of the ith UAV in the terms of its position and orientation is $P_{si}(x_{si}, y_{si}, z_{si}, \theta_{si})$ and the final pose of the same UAV is $P_{fi}(x_{fi}, y_{fi}, z_{fi}, \theta_{fi})$ then the path of the UAV is defined as a parametric curve *r(t)* = [*x(t), y(t), z(t)*] such that the kinematics and dynamic constraints of the vehicle are satisfied. Mathematically this can be written as:

$$P_{si}(x_{si}, y_{si}, z_{si}, \theta_{si}, \phi_{si}) = r(t_s)$$

$$P_{fi}(x_{fi}, y_{fi}, z_{fi}, \theta_{fi}, \phi_{fi}) = r(t_f)$$

$$P_{si}(x_{si}, y_{si}, z_{si}, \theta_{si}, \phi_{si}) \xrightarrow{r_i(t)} P_{fi}(x_{fi}, y_{fi}, z_{fi}, \theta_{fi}, \phi_{fi})$$

Subjected to:

$$|\kappa_i(t)| \leq \kappa_{\max}$$

$$|\tau_i(t)| \leq \tau_{\max}$$

With curvature k_i and torsion τ_i for i=1,..,*n* , *n* is the number of cooperating UAVs. In case of obstacle interruption these initial paths are modified such that the modified path is feasible (collision free and flyable). We assume in the developments of our paper that the UAVs are flying at a constant altitude. How we generate these paths is explained in the next section.

3 PROPOSED SOLUTION TO THE PROBLEM

The initial trajectories of the UAVs are calculated from known initial and final poses. Therefore the poses play a pivotal rule in generating and modifying these trajectories. For each path there are two poses i.e the initial pose and final pose. The orientation of these poses could be from any of the four quadrants of Cartesian plane. Since there are total of four quadrants, and any two quadrant can be selected at a time for two orientation (one for initial pose and one for final pose), therefore we can have a total of:

$$\binom{4}{2} = \frac{4!}{(4-2)! \times 2!} = 6$$

possible cases. These cases are shown in the figure 5 as a, b, c, d, e and f. There are four more possible cases if the initial and final poses are taken from the same quadrant. These cases are shown in figure 5 as g, h, i and j. The orientations corresponding to initial poses are represented by green arrow and the orientations corresponding to the final poses are represented by red arrow. Referring to the figure 5 below, we have the following cases:

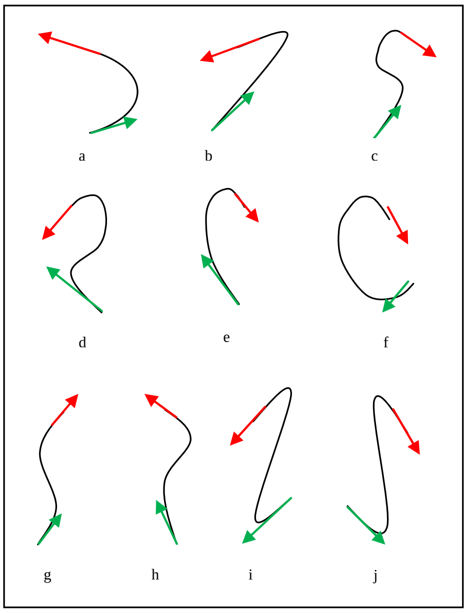


Figure 5: All possible combinations of the initial and final pose taken from four quadrant.

Case a: The initial pose belongs to quadrant 1 and the final pose belongs to quadrant 2. i.e.

$$0 \le \theta_{si} \le \frac{\pi}{2} \text{ and } \frac{\pi}{2} \le \theta_{fi} \le \pi$$

Case b: The initial pose belongs to quadrant 1 and the final pose belongs to quadrant 3:

$$0 \le \theta_{si} \le \frac{\pi}{2} \text{ and } \pi \le \theta_{fi} \le 3\frac{\pi}{2}$$

Case c: The initial pose belongs to quadrant 1 and the final pose belongs to quadrant 4:

$$0 \le \theta_{si} \le \frac{\pi}{2} \text{ and } 3\frac{\pi}{2} \le \theta_{fi} \le 2\pi$$

Case d: The initial pose belongs to quadrant 2 and the final pose belongs to quadrant 3:

$$\frac{\pi}{2} \le \theta_{si} \le \pi \text{ and } \pi \le \theta_{fi} \le 3\frac{\pi}{2}$$

Case e: The initial pose belongs to quadrant 2 and the final pose belongs to quadrant 4:

$$\frac{\pi}{2} \le \theta_{si} \le \pi \quad \text{ and } \quad 3\frac{\pi}{2} \le \theta_{fi} \le 2\pi$$

Case f: The initial pose belongs to quadrant 3 and the final pose belongs to quadrant 4:

$$\pi \le \theta_{si} \le 3\frac{\pi}{2} \text{ and } 3\frac{\pi}{2} \le \theta_{fi} \le \pi$$

Case g: both the initial pose and the final pose belong to quadrant 1: $0 \le \theta_{si}, \theta_{fi} \le \frac{\pi}{2}$

Case h: both the initial pose and the final pose belong to quadrant 2: $\frac{\pi}{2} \le \theta_{si}, \theta_{fi} \le \pi$

Case i: both the initial pose and the final pose belong to quadrant 3: $\pi \le \theta_{si}, \theta_{fi} \le 3\frac{\pi}{2}$

Case j: both the initial pose and the final pose belong to quadrant 4: $3\frac{\pi}{2} \le \theta_{si}, \theta_{fi} \le 2\pi$

Now cases a, b, e and f are safe as shown by figure 1 and figure 2 in the introduction section.

Cases c, d, g, h, i and j have the possibility to give paths of unacceptably higher bending energy when manipulated to avoid obstacles. Therefore they needed some method such that safe and flyable paths are produced.

The method to avoid the obstacle in the case of c, d, g, h, i and j (unsafe cases) comprise of introducing an intermediate waypoint (pose) somewhere between the initial and final pose such that the first and second pose, and, second and third pose can be connected by two PH quintic curves. Each of these two individual PH component becomes like one of the case a, b, e or f.

Since pose is a combination of position and direction, therefore the position and direction of the inserted intermediate pose must be determined. The following paragraphs describe the determination of the position and direction.

4.1 Position of the Intermediate Pose

Referring to the figure 6, if (x_{cen}, y_{cen}) are the coordinates of the centre of the obstacle, r_d is the radius of the circle enclosing the obstacle, then the coordinates of the points of the circle enclosing the obstacle are (x, y) given by the following equations:

$$x = x_{cen} + r_d \cos\theta$$

$$y = y_{cen} + r_d \sin\theta$$

Where $\theta \in [0 \;\; 2\pi]$

The position (x_{iwp}, y_{iwp}) corresponding to the new waypoint is given by:

$$x_{iwp} = d_{saf} + x_{cn} + r_d \cos\xi$$

$$y_{iwp} = d_{saf} + y_{cn} + r_d \sin\xi$$

$\xi = \frac{\pi}{2}$, if the UAV is approaching from the bottom of the obstacle.

$\xi = \pi$, if the UAV is approaching from the right.

$\xi = 3\frac{\pi}{2}$,if the UAV is approaching from the top.

$\xi = 0$,if the UAV is approaching from the left. d_{saf} is the safety distance.

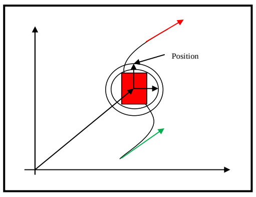


Figure 6: Position specification of the intermediate waypoint.

4.2 Direction of the Intermediate Pose

The intermediate waypoint is inserted between the initial and final poses. The direction of the intermediate waypoint θ_{iwp} is such that when the two consecutive poses are connected via PH quintic in case of c, d, g, h, i and j each individual PH segment becomes like one of the cases a, b, d or e. We consider each individual case separately.

Case g, i: If both the directions of initial and final poses belong to quadrant 1 or quadrant 3 then the

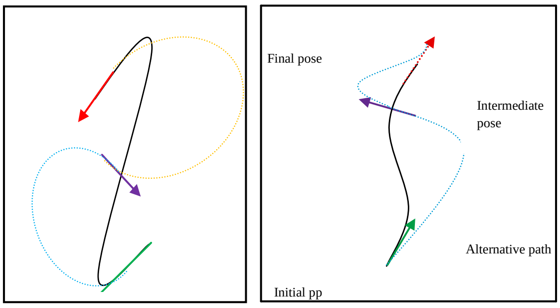


Figure 7: Direction of intermediate waypoint for cases g and i.

direction of intermediate pose is $\theta_{iwp} = \theta_{si} + \frac{\pi}{2}$

Case h, j: If both the directions of initial and final poses belong to quadrant 2 or quadrant 4 then the direction of intermediate pose is $\theta_{iwp} = \theta_{si} - \frac{\pi}{2}$

Case c: If the directions of initial pose belong to quadrant 1 and that of final poses belong to quadrant 4 then the direction of intermediate pose is

$$\theta_{iwp} = \theta_{si} + \frac{\pi}{2}$$

Case c inverted: If the directions of initial pose belong to quadrant 4 and that of final poses belong to quadrant 1 then the direction of intermediate pose is $\theta_{iwp} = \theta_{si} - \frac{\pi}{2}$

This is shown in figure 9.

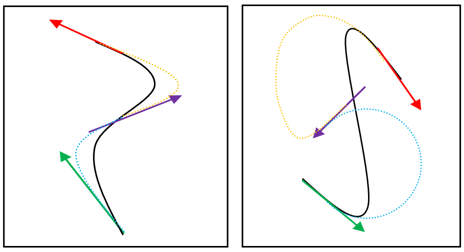


Figure 8: Direction of intermediate waypoint for cases h and j.

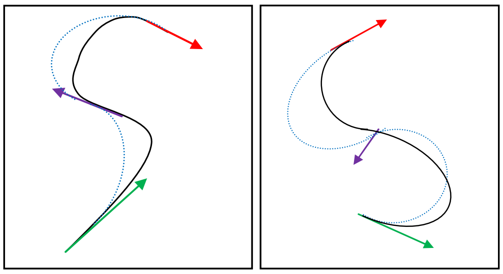


Figure 9: Direction of intermediate waypoint for cases c and c inverted.

Case d: If the directions of initial pose belong to quadrant 2 and that of final poses belong to quadrant 3 then the direction of intermediate pose is: $\theta_{iwp} = \theta_{si} - \frac{\pi}{2}$

Case d inverted: If the directions of initial pose belong to quadrant 3 and that of final poses belong to quadrant 2 then the direction of intermediate pose is $\theta_{iwp} = \theta_{si} + \frac{\pi}{2}$

4 SIMULATION RESULTS

The simulation results are obtained y taking only one case in which the direction of initial and final poses belong to quadrant 1. The method can be tested for the rest of cases. The two methods (the curvature manipulation method and the proposed method) were simulated by taking the initial pose (14, 6, 60^o) and final pose (17, 40, 60^o). The following results were obtained.

Figure 11 shows the path and the obstacle (red rectangle).

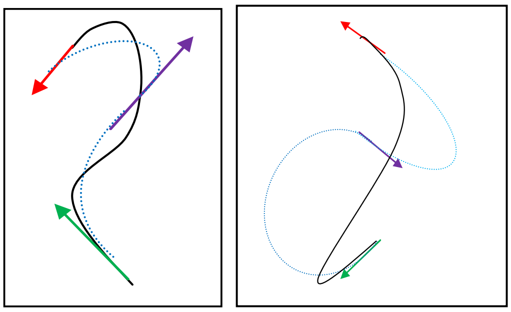


Figure 10: Direction of intermediate waypoint for cases d and d inverted.

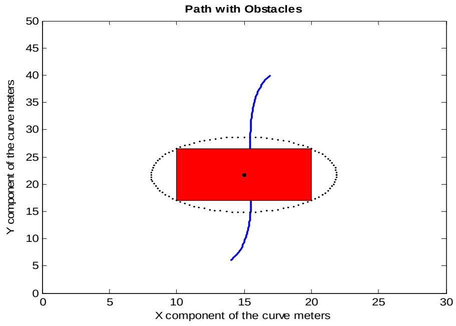


Figure 11: The path with the obstacle.

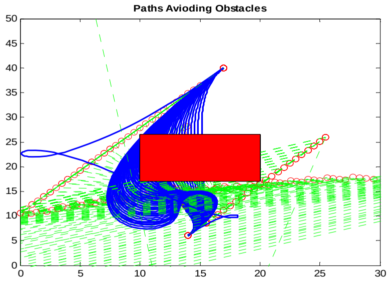


Figure 12: The path with different stages of curvature manipulation to avoid the obstacle.

When the curvature manipulation method was applied to avoid the obstacle (figure 12), the result was a path with an unacceptable bending energy as shown in the figure 13. The path shown in the figure 13 is not a flyable path because its high bending energy makes it difficult to obey the dynamic constraints. If we try to impose the dynamic constraints then its length will be increased tremendously. The proposed method can offer a remedy to this problem. The proposed method is applied to insert an intermediate pose between the initial and final pose. Then the three poses were connected by the application of Pythagorean Hodograph curves such that the individual curves comply with the afore mentioned safe cases.

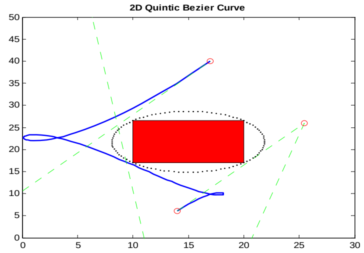


Figure 13: The final obstace free path with very high bending energy.

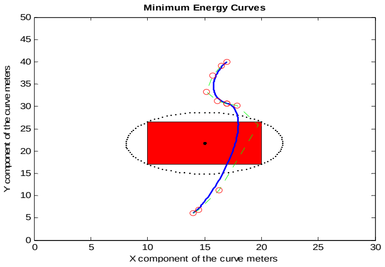


Figure 14: The path with same initial and final poses resulted from the proposed method.

The result is shown in the figure 14. Figure 15 show the same path after imposing the dynamic constraints. Figure 16 shows the iterative process to avoid the obstacle. Figure 17 shows the final obstacle free path. The path shown in figure 17 obeys the dynamic constraints. The binding energy is much lower compared to the path achieved with the curvature manipulation method. The length of the path is optimal as well.

5 CONCLUSIONS

The curvature manipulation method fails to give feasible paths if the angle of the initial pose and final pose in the first quadrant is increased above 60^o and 55^o on the upper side and decreased below 40^o and 30 on the lower side. Therefore the operational angle band of the method is very narrow which makes it unsuitable for practical purposes. The proposed method can accommodate poses with all the angles. More over the path length of the resultant path is optimal.

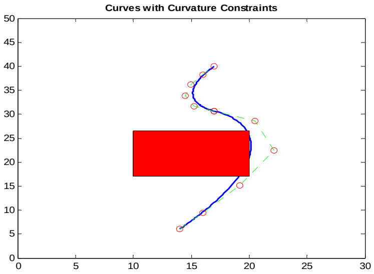


Figure 15: The path with curvature constraint.

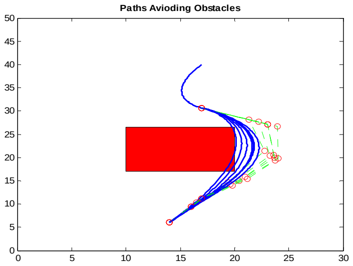


Figure 16: The curvature manipulation of the path to avoid the obstacle.

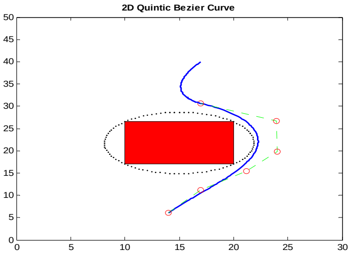


Figure 17: The obstacle free path with reasonable bending energy and reasonable path length.

ACKNOWLEDGEMENTS

This research was sponsored by Engineering Physical Science Research Council (EPSRC) and British Aerospace (BAE Systems). Support is gratefully acknowledged and appreciated by the authors.

REFERENCES

S. Madhavan, A. Tsourdos, B. White. “A Solution to Simultaneous Arrival of Multiple UAVs using Pythagorean Hodograph” American Control Conference, Minnesota USA, June 2006.

H. Bruyninckx, D. Reynaerts, “Path planning for mobile and hyper redundant robots using Pythagorean hodograph curves”, In 8th International Conference on Advanced Robotics, ICAR, 97 pages 595-600, 1997.

MOBILE ROBOT OBSTACLE DETECTION USING AN OVERLAPPED ULTRASONIC SENSOR RING

Sungbok Kim, Jaehee Jang and Hyun Bin Kim
Department of Digital Information Engineering, Hankuk University of Foreign Students, Korea
sbkim@hufs.ac.kr, {genesisofgaia, rlagusqls}@gmail.com

Keywords: Ultrasonic Sensor, Overlapped Beam Pattern, Positional Uncertainty, Obstacle Position, Sensor Model.

Abstract: This paper presents the obstacle detection of a mobile robot using an ultrasonic sensor ring with overlapped beam pattern. Basically, it is assumed that a set of ultrasonic sensors are installed at regular intervals along the side of a circular mobile robot of nonzero radius. First, by exploiting the overlapped beam pattern, it is shown that the positional uncertainty inherent to an ultrasonic sensor can be significantly reduced for both single and double obstacle detection. Second, given measured distances from adjacent ultrasonic sensors, the geometric method of computing the position of the detected obstacle with respect to the center of a mobile robot is described. Third, through experiments using our ultrasonic sensor ring prototype, the validity and the performance of the proposed overlapped ultrasonic sensor ring are demonstrated.

1 INTRODUCTION

Since the mid 1980s, ultrasonic sensors have been widely used for map building and obstacle avoidance. Typically, an ultrasonic sensor equipped at a mobile robot operates in reflective mode, rather than in direct wave mode. Once the acoustic wave is emitted by the transmitter, an ultrasonic sensor can measure the obstacle distance using the elapsed time until the reflected wave is sensed by the receiver, which is called the time of flight. The detected obstacle is present at a certain point along the arc of radius given by the obstacle distance; however, the exact position of an obstacle on the arc remains unknown. This is called the positional uncertainty inherent to an ultrasonic sensor.

To alleviate the problem of positional uncertainty, we propose to overlap the cone shaped beams of two ultrasonic sensors in part. With the overlapped beam pattern, the entire sensing zone of each ultrasonic sensor can be divided into two smaller sensing subzones: the overlapped and the unoverlapped subzones. If both ultrasonic sensors detect an obstacle, the detected obstacle should belong to the overlapped subzone. On the other hand, if either ultrasonic sensor detects an obstacle, the detected obstacle should belong to the unoverlapped subzone. This indicates that the positional uncertainty of an ultrasonic sensor can be reduced by exploiting the overlapped beam pattern of two ultrasonic sensors. In this paper, we present the obstacle detection of a mobile robot using an ultrasonic sensor ring with overlapped beam pattern.

2 POSITIONAL UNCERTAINTY REDUCTION

Assume that N ultrasonic sensors of the same type are arranged in a circle at regular intervals with their beams overlapped. After numbering the ultrasonic sensors from 1 to N in clockwise order, let $\mathrm{S}_k,\ k = 1,\cdots,N$, denote the k^{th} ultrasonic sensor of an overlapped ultrasonic sensor ring. Fig. 1 shows three adjacent ultrasonic sensors, S_N, S_1, and S_2, which are placed at the left, the center, and the right, respectively. Due to beam overlapping, the entire sensing zone of the ultrasonic sensor S_1 can be divided into three sensing subzones, denoted by ‘I’, ‘II’, and ‘III’. In Fig.1, an obstacle P is depicted by a bold.

First, let us consider the detection of a single obstacle using three overlapped ultrasonic sensors. Let ρ_N, ρ_1, and ρ_2 be the distances of the obstacle P measured by three ultrasonic sensors, S_N, S_1, and S_2, respectively. Table 1 shows the distance measurements of S_N, S_1, and S_2, depending on to the position of P among one of three sensing subzones, either I, II, or III. In Table 1, '∞' indicates the situation that P cannot be detected by the corresponding ultrasonic sensor. Seen from Table 1, the combination of three ultrasonic sensors which return finite measured distances varies according to where the obstacle belongs among three sensing subzones. This implies that the sensing subzone to which the obstacle belongs can be determined based on the combination of three ultrasonic sensors returning finite measured distances. It should be noted that the beam overlapping between adjacent ultrasonic sensors can lead to the significant reduction in positional uncertainty.

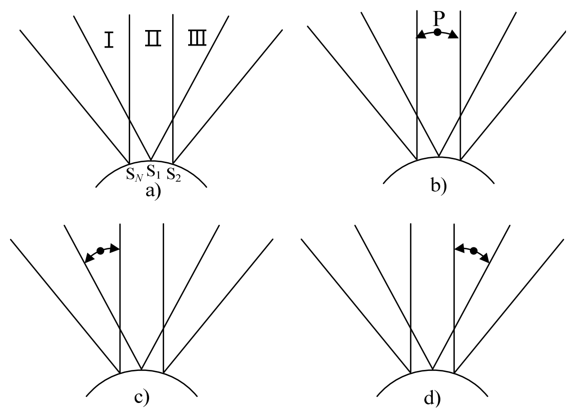


Figure 1: Single obstacle detection using overlapped ultrasonic sensors.

Table 1: Distance measurements of ultrasonic sensors in the case of single obstacle detection.

	2b)	2c)	2d)
P	II	I	III
S_N	∞	ρ_N	∞
S_1	ρ_1	ρ_1	ρ_1
S_2	∞	∞	ρ_2

Next, let us consider the detection of double obstacles using three overlapped ultrasonic sensors. Note that the number of obstacles that can be detected by two adjacent ultrasonic sensors is at most two, regardless of how many obstacles are present in front of them. Assume that one obstacle is present at far distance, called the far obstacle and denoted by P_f, and the other obstacle is present at the near distance, called the near obstacle and denoted by P_n. Fig. 2 shows three different cases of how the far and near obstacles are located. Let $\rho_{f,i}$ and $\rho_{n,i}$ ($< \rho_{f,i}$), $i = N, 1, 2$, be the distances of the obstacles, P_f and P_n, measured by three ultrasonic sensors, S_N, S_1, and S_2, respectively.

Table 2 shows the distance measurements of three ultrasonic sensors, S_N, S_1, and S_2, depending on the relative positions of the far and the near obstacles, P_f and P_n. Referring to Table 2, the following decisions can be made on where P_f and P_n are positioned among three sensing subzones, I, II, and III. For instance, if the left ultrasonic sensor S_N returns larger value than the other two ultrasonic sensors, S_1 and S_2, as shown in Fig. 2a), P_f is positioned within I and P_n is positioned within III.

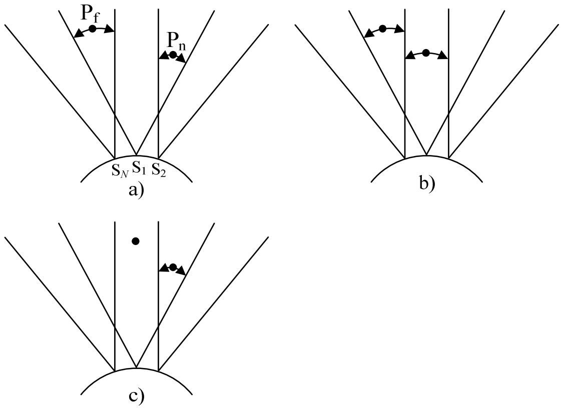


Figure 2: Double obstacle detection using overlapped ultrasonic sensors.

Table 2: Distance measurements of ultrasonic sensors in the case of double obstacle detection.

	3a)	3b)	3c)
P_f	I	I	II
P_n	III	II	III
S_N	$\rho_{f,N}$	$\rho_{f,N}$	∞
S_1	$\rho_{n,1}$	$\rho_{n,1}$	$\rho_{n,1}$
S_2	$\rho_{n,2}$	∞	$\rho_{n,2}$

3 MOBILE ROBOT REFERENCED OBSTACLE DETECTION

Each ultrasonic sensor of an ultrasonic sensor ring returns the distance of an obstacle that is measured with reference to the vertex of its own, called the sensor referenced obstacle distance. On the other hand, it is the obstacle position with reference to the center of a mobile robot, called the mobile robot referenced obstacle position that should be known for the obstacle detection/avoidance of a navigating mobile robot.

First, suppose that an obstacle P is present at the distance ρ_1 within the unoverlapped sensing subzone II of the ultrasonic sensor S_1, as shown in Fig. 3. Let G_1 and F_1 denote the intersecting points of the circle of radius ρ_1 centered at the point S_1, and the right beam boundary of the ultrasonic sensor S_N and the right beam boundary of the ultrasonic sensor S_2, respectively.

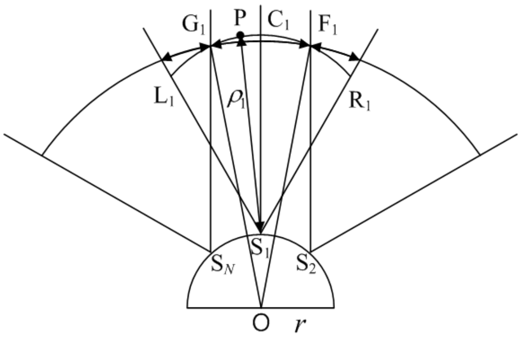


Figure 3: The mobile robot referenced position of an obstacle within the unoverlapped sensing zone.

Due to the positional uncertainty, an obstacle P can exist at a certain point along the arc $\widehat{G_1C_1F_1}$ of radius ρ_1 centered at the point S_1, instead of the arc $\widehat{L_1C_1R_1}$. Considering that the ultimate goal of obstacle detection is to prevent possible collision with obstacles, it is reasonable to take conservative stance in guessing the obstacle position. To prepare for the worst case scenario, the collision free region can be specified as the intersection of the circle of radius $\overline{OF_1}$ centered at the point O and the cone of angle $\angle G_1OF_1$ centered along the line $\overline{OC_1}$. As a result of the beam overlapping of an ultrasonic sensor ring, 1) the radius of the collision free region increases, that is, $\overline{OF_1} > \overline{OR_1}$, and 2) the angle of the collision free region decreases, that is, $\angle G_1OF_1 < \angle L_1OR_1$. The increased radius can allow a mobile robot more room for obstacle avoidance, and the decreased angle can improve the spatial resolution in obstacle detection.

Next, suppose that an obstacle P is present at the distance ρ_1 within the overlapped sensing subzone III of the ultrasonic sensor S_1, as shown in Fig. 7. Let ρ_1 and ρ_2 be the distance of P from the vertices of ultrasonic sensors, S_1 and S_2, respectively. Owing to the positional uncertainty of two ultrasonic sensors, P exists along the arc $\widehat{L_1R_1}$ of radius ρ_1 centered at the point S_1, and it also exists along the arc $\widehat{L_2R_2}$ of radius ρ_2 centered at the point S_2. As shown in Fig. 4, the position of P can now be obtained from the intersection of these two circles, which results in two intersecting points in general. Finally, the obstacle position can be uniquely determined out of two intersecting points, based on the relative locations of ultrasonic sensors, S_1 and S_2. It should be noted that the problem of uncertainty does not exist anymore for an obstacle within the overlapped sensing subzone, although the measurement errors may still affect the accuracy in computing the obstacle position.

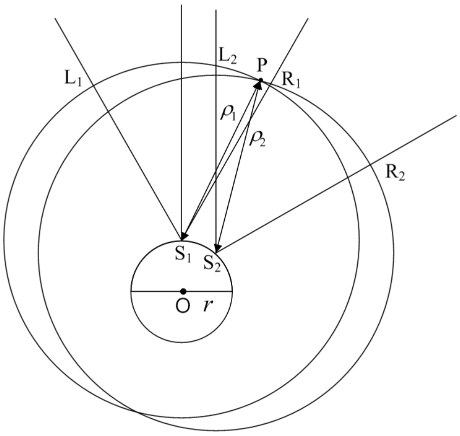


Figure 4: The mobile robot referenced position of an obstacle within the overlapped sensing subzone.

4 EXPERIMENTAL RESULTS

An omnidirectional overlapped ultrasonic sensor ring prototype was built using the ultrasonic sensor modules which contain MA40B8 of beam width $\alpha = 50°$ from Murata Inc. As shown in Fig. 5, twelve ultrasonic sensor modules are first installed at a regular spacing of $\beta = 30°$ between two circular acrylic plates of radius $r = 19$ cm, which is then fixed on top of a circular mobile robot concentrically.

First, a mobile robot is commanded to move along a straight line, which is 1 m apart from a circular obstacle of diameter 12.5 cm. Fig. 6 shows the division of the straight line depending on the combination of ultrasonic sensors involved in obstacle detection. It can be observed that the distance from P_1 to P_3 is about 170 cm and the distance from P_2 to P_5 is about 140 cm. Using these data, the effective beam widths can be calculated: 70° for the center ultrasonic sensor and 60° for the left and right ultrasonic sensors.

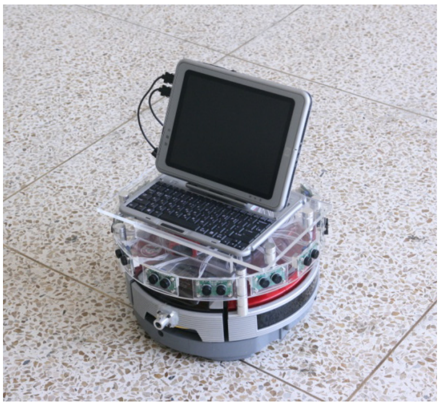


Figure 5: Our overlapped ultrasonic sensor ring prototype.

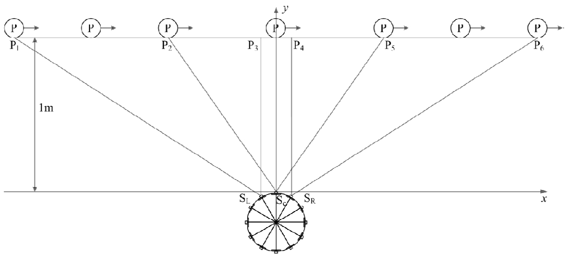


Figure 6: The experimental results for single obstacle detection.

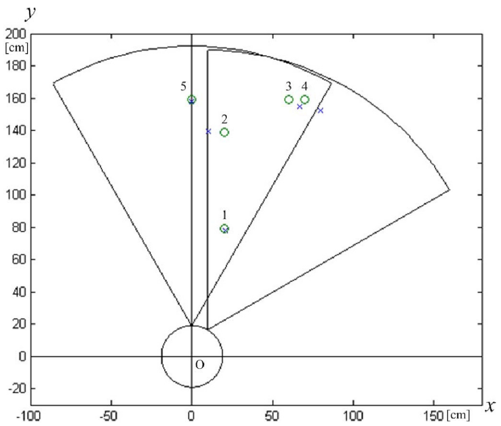


Figure 7: The experimental setting and results for computing mobile robot referenced obstacle positions.

Next, for five different locations of an obstacle, we compute the mobile robot referenced position of an obstacle from the measured distances of three ultrasonic sensors. Fig. 7 shows the actual and the computed obstacle positions, which are marked by 'o' and 'x', respectively. The discrepancy between the actual and the computed obstacle positions tends to increase as an obstacle is located away from the center of the overlapped ultrasonic sensor ring and close to the boundary of the overlapped sensing subzone. However, it is confirmed through repetitive experiments that the maximum discrepancy is bounded within 10 cm.

5 CONCLUSIONS

In this paper, we proposed an overlapped ultrasonic sensor ring which consists of relatively small number of low cost ultrasonic sensors with low directivity, to reduce the positional uncertainty in obstacle detection. The proposed ultrasonic sensor ring made of low directivity ultrasonic sensors is advantageous over its high directivity counterpart in both sensor device and data processing requirements. It is expected that the results of this paper can facilitate early deployment of low cost mobile platforms for personal service robots.

REFERENCES

Borenstein, J. and Koren, J., 1989. Real-Time Obstacle Avoidance for Fast Mobile Robots. In *IEEE Trans. Sysems, Man, and Cybernetics*, vol. 19, no. 5, pp. 1179-1187.

Choset, H., Nagatani, K., and Lazar, N. A., 2003. The Arc-Traversal Median Algorithm: A Geometric Approach to Increase Ultrasonic Sensor Azimuth Accuracy. In *IEEE Trans. Robotics and Automation*, vol. 19, no. 3, pp. 513-522.

Crowley, J. L., 1989. World Modeling and Position Estimation for a Mobile Robot Using Ultrasonic Ranging. In *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 674-680.

Elfes, A., 1987. Sonar-Based Real-World Mapping and Navigation. In *IEEE J. Robotics and Automation*, vol. RA-3, no. 3, pp. 249-265.

McKerrow, P. J., 1993. Echolocation-from Range to Ontline Segments. In *Robotics and Autonomous Systems*, vol. 11, no. 4, pp. 205-211.

Moravec, H., and Elfes, A., 1985. High Resolution Maps for Wide Angles Sonar. In *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 116-121.

Murphy, R., 2000. *Introduction to AI Robotics,* The MIT Press.

Thrun, S., Burgard, W., and Fox, D., 2005. *Probabilistic Robotics,* The MIT Press.

Wijk, O. and Christensen, H. I., 2000. Triangulation-Based Fusion of Sonar Data with Application in Robot Pose Tracking. In *IEEE Trans. Robotics and Automation*, vol. 16, no. 6, pp. 740-752.

SALT AND PEPPER NOISE DETECTION BASED ON NON-LOCAL MEANS

Carlos Junez-Ferreira, Fernando Velasco-Avalos
FIE, Universidad Michoacana de San Nicolás de Hidalgo, Ciudad Universitaria, Morelia, Mexico
cjunez@www.fic.umich.mx, fvelasco@umich.mx

Nelio Pastor-Gomez
FIC, Universidad Michoacana de San Nicolás de Hidalgo, Morelia, Mexico
npastor@umich.mx

Keywords: Image denoising, Salt and pepper noise, Non-local means, Adaptive median filter.

Abstract: Denoising is an important task inside the image processing area. In this paper, an algorithm for detecting and suppressing salt and pepper noise is presented. Firstly, the algorithm computes an estimation of the denoised image by using a variant of the Non-Local Means proposal. This estimation is segmented in order to detect corrupted pixels avoiding misclassifying pixels with extreme values that belong to objects on the uncorrupted image. Once pixels are classified, the algorithm performs a suppression step by using an adaptive median filter. Obtained results show that the implementation of this proposal gives good noise detection and suppression.

1 INTRODUCTION

Frequently, digital images are corrupted by undesirable random variations in intensity values, called noise. The presence of noise is due to several factors, among which are found the process of acquisition, compression and transmission of data. A common type of noise is impulsive noise, known as salt and pepper. This kind of noise randomly changes intensities of some pixels to the maximum ($v_{\max}$) or minimum ($v_{\min}$) values of the intensity range on the image (Chan et al., 2005).

The median filter has been widely used for impulsive noise suppression due to its ability to preserve edges in a better way than weighted averaging methods. It must be noticed that, median filter does not insure the edge preservation, mainly in images with high noise density. Numerous algorithms based on the median filter have been presented, as the proposed by Wang & Zhang (1999), Yuan & Tan (2006), and Chan et al. (2005).

2 IMPULSE NOISE DETECTION

Evidently, good noise detection is essential to carry out its suppression and to get a better restoration quality. One way to detect noise is through an adaptive median filter, as used by Chan et al. (2005), which is described below.

Let $\tilde{\boldsymbol{y}}$ the obtained image by applying an adaptive median filter to an image with salt and pepper noise $\boldsymbol{y}$. Moreover, according to the salt and pepper noise model, noisy pixels take its values from the set $\{v_{\min}, v_{\max}\}$. Then, it could be defined the set of corrupted pixels candidates as

$$N = \{y_{i,j} : \tilde{y}_{i,j} \neq y_{i,j}, y_{i,j} \in \{v_{\min}, v_{\max}\}\} \tag{1}$$

for all $(i, j) \in A$.

Another proposal has been presented by Wang & Zhang (1999), where the following expression is used

$$N = \{y_{i,j} : |\tilde{y}_{i,j} - y_{i,j}| > T\} \tag{2}$$

where $(i, j) \in A$ and T is a predefined threshold. Expression (2) is used iteratively in order to detect impulse noise.

3 NON-LOCAL MEANS

Buades et al. (2005) proposed the Non-Local Means algorithm, based on the idea that images contain repeated structures and that averaging these structures, the noise of an image can be reduced.

Given a discrete image with noise $\boldsymbol{y}$ the restored value $\hat{y}_{m,n}$, for the pixel at location $(m,n) \in A$, is computed as the weighted average of all pixels of the image,

$$\hat{y}_{m,n} = \sum_{(i,j)\in A} w_{i,j}^{m,n} y_{i,j} \tag{3}$$

where the family of weights $\{w_{i,j}^{m,n}\}$ depends on the similarity between pixels at positions $(m,n) \in A$ and $(i,j) \in A$, and it satisfies the conditions $0 \le w_{i,j}^{m,n} \le 1$ and $\sum_{(i,j)\in A} w_{i,j}^{m,n} = 1$.

Structural similarity between $y_{i,j}$ and $y_{m,n}$ depends on similarity between vectors $V(\Omega_{i,j})$ and $V(\Omega_{m,n})$, where $\Omega_{k,l}$ denotes a fixed size neighbourhood and centered at pixel $y_{k,l}$.

Similarity between above mentioned vectors is measured by a decreasing function of Euclidean distance, $d_{i,j}^{m,n} = \left\| V(\Omega_{m,n}) - V(\Omega_{i,j}) \right\|^2$. Pixels with similar neighbourhood to $V(\Omega_{m,n})$ will have large weights, which are defined as

$$w_{i,j}^{m,n} = \frac{1}{Z_{m,n}} e^{-\frac{d_{i,j}^{m,n}}{H^2}} \tag{4}$$

where $Z_{m,n}$ is a normalization constant, and the parameter H acts as a filtering degree, that is, it controls the decay of weights as a function of distances. For implementation purposes, a window of size W_1 is used to compute the average with a limited number of neighbours, instead of averaging all pixels of the image. Also, a window of size W_2 is used to define the structure of the neighbourhood and the size of vector $V(\Omega_{k,l})$.

In general, the Non-Local Means algorithm gives good results in terms of noise reduction, however, this does not always happen, especially in images with high salt and pepper noise level.

4 SALT AND PEPPER NOISE DETECTION AND SUPPRESION PROPOSAL

In this work, the Non-Local Means algorithm will be used to provide a preliminary estimate of the restored image, with the aim of detecting salt and pepper noise, even with high density.

In order to consider the presence of objects whose pixels intensity values are equal to the maximum or minimum values on image, in other words, black or white objects in the image without noise, we propose a segmentation, which is performed by grouping neighbouring pixels with similar intensity values, based on a threshold *TI*.

The purpose of making this segmentation is to find a partition $\boldsymbol{S}$ of an image $\hat{\boldsymbol{y}}$ on a set of regions, in such a way that

$$R_{big} \cup R_{small} = \hat{\boldsymbol{y}} \tag{5}$$

where

$$R_{big} = \{R_s : h(R_s) > P\}\text{, and} \tag{6}$$

and,

$$R_{small} = \{R_s : h(R_s) \le P\}, \tag{7}$$

R_{big} is the set of regions considered relevant objects, that is, regions whose number of pixels represent a percentage of image greater than a threshold *P*. Obviously, R_{small} is the set of regions that are, by their size, regarded as details or noise. The function $h(R_s)$ computes the percentage of the image that corresponds to a region R_s. Thus, one can discriminate between pixels that can be considered corrupt and those that belong to an object, although in both cases the pixels intensities are extreme value.

Considering the above exposed, our proposal to detect salt and pepper noise can be described, in general, through the following steps:

```
$\hat{\mathbf{y}}$ =Compute_Estimation($\mathbf{y}$,TW,PAR)
$\mathbf{S}$=Segmentation($\hat{\mathbf{y}}$,TI)
$\alpha$ =Pixels_Classification($\mathbf{y}$,$\mathbf{S}$)
$\hat{\mathbf{x}}$ =Noise_Suppression($\hat{\mathbf{y}}$, $\alpha$, $W_{min}$,$W_{max}$)
```

where PAR is a vector containing the parameters *H*, W_1 and W_2.

The preliminary estimate, described in Figure 1, is performed by calculating the weights of the pixels in the neighbourhood for a corrupt pixel candidate according to expression (4), and its estimated value will be the median value only of intensity values of pixels with weights greater than a threshold *TW*. In this way, only pixels whose structure is similar to the pixel in question are involved. The median value

```
Compute_Estimation($\mathbf{y}$,TW,$PAR_1$)
  for each pixel at position (i,j)
    if ($y_{i,j}=v_{min}$) or ($y_{i,j}=v_{max}$)
      Compute family of weights $w^{i,j}$
      for each weight $w^{i.j}_{m,n}$
        if $w^{i.j}_{m,n} \geq TW$
          $VW \leftarrow VW \cup y_{m,n}$
        end
      end
      $\hat{y}_{i,j} \leftarrow$ MEDIAN(VW)
    else
      $\hat{y}_{i,j} \leftarrow y_{i,j}$
    end
  end
  return($\hat{\mathbf{y}}$)
```

Figure 1: Pseudo-code of estimation computing.

```
Pixels_Clasification($\mathbf{y}$,$\mathbf{S}$)
  for each pixel at position (i,j)
    if ($y_{i,j}=v_{min}$) or ($y_{i,j}=v_{max}$)
      if $(i,j) \in R_{small}$
        $\alpha_{i,j} \leftarrow 1$
      else
        if ($y_{i,j} \neq \hat{y}i,j$)
          $\alpha_{i,j} \leftarrow 1$
        else
          $\alpha_{i,j} \leftarrow 0$
        end
      end
    else
      $\alpha_{i,j} \leftarrow 0$
    end
  end
  return($\boldsymbol{\alpha}$)
```

Figure 2: Pseudo-code of classification step.

is used instead of weighted averaging because its better results with salt and pepper noise.

Subsequently, this estimation will be segmented, as described above, then, a pixel is considered corrupt if its intensity value is an extreme value and the pixel belongs to a region considered irrelevant. If the pixel belongs to a relevant object and its value has changed at estimation, this pixel is also considered corrupted. Figure 2, shows the classification procedure, where if $\alpha_{i,j}=0$, the pixel at position (i,j) is regarded as corrupt and $\alpha_{i,j}=1$ if not.

Once pixels are classified, the algorithm proceeds to noise reduction, by applying an adaptive median filter. This kind of filtering considers geometric closeness in restoration. First, a neighbourhood for corrupt pixels is defined, then uncorrupted pixels belonging to the neighbourhood are stored in a vector, and the central pixel is replaced by median value of this vector. If the mentioned vector is empty, the window size is increased, repeating the procedure. If the size increases to reach a maximum size of window, the central pixel value is replaced by the median value of all pixels in neighbourhood. This procedure is described in Figure 3.

It can be determined that, the proposed algorithm is O(N_D), where N_D is the number of data.

```
Noise_Suppression($\hat{\mathbf{y}}$,$\boldsymbol{\alpha}$,$W_{min}$,$W_{max}$)
  for each pixel at position (i,j)
    if $\alpha_{i,j} = 1$
      $W \leftarrow W_{min}$, witness $\leftarrow$ 0
      do
        $VN \leftarrow \phi$
        for each element in window $\Omega_{i,j}$
        of size $W^2$
          if $\alpha^{i,j}_{m,n} = 0$
            $VN \leftarrow VN \cup \hat{y}_{m,n}$
          end
        end
        if $VN \neq \phi$
          $\hat{x}_{i,j} \leftarrow$ MEDIAN(VN)
          witness $\leftarrow$ 1
        else
          $W \leftarrow W + 2$
        end
      while ($W \leq W_{max}$) & (witness = 0)
      if ($W > W_{max}$) & (witness = 0)
        $\hat{x}_{i,j} \leftarrow$ MEDIAN($\Omega_{i,j}$ of size $W^2_{max}$)
      end
    else
      $\hat{x}_{i,j} \leftarrow \hat{y}_{i,j}$
    end
  end
  return($\hat{\mathbf{x}}$)
```

Figure 3: Pseudo-code of noise suppression.

5 EXPERIMENTAL RESULTS

Experiments show that results obtained by applying the described algorithm are very good. The implementation of the code was made using Matlab 6.5 R13 on a PC 2.20 GHz Core Duo CPU.

The test image (Figure 4) was corrupted with salt and pepper noise with different densities through *imnoise* function of Matlab.

In order to compute the estimation, the values $W_1=3$, $W_2=3$, $TW=0.10$ and $H=10$ were used. For segmentation $T=0.001$ (the intensity values range of image is [0, 1]), and the value of P depends on the

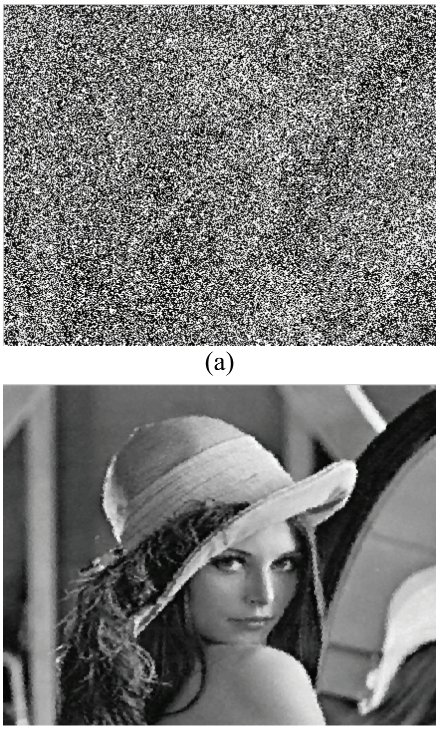


Figure 4: Image 1, (a) Noisy image (80%), (b) Denoised image.

maximum size of a region that can be considered irrelevant, we use *P=0.001*. For noise reduction we used $W_{min}=3$ and $W_{max}=10$.

In order to quantify the performance of the detector, we used the values (Yuan & Tan, 2006) $\Gamma_1 = \gamma_g / \gamma_d$ and $\Gamma_2 = \gamma_g / \gamma_n$, where γ_d is the number of pixels detected with the proposed algorithm, γ_n is the number of pixels that are really corrupted and γ_g is the number of detected pixels that are really corrupted. Tables 1 and 2 show the performance of our proposal, which is identified as 1, for different noise densities on Lena image, compared with algorithm 2 (Chan et al., 2005) and 3 (Wang & Zhang, 1999). Table 3 shows the performance of the algorithm for noise reduction measured by PSNR and UIQI (Wang & Bovik, 2002).

Table 1: Performance of different detectors for Lena image with a noise density of 20%.

Algorithm	Γ_1	Γ_2	Time (s)
1	0.9988	1	17.59
2	0.7922	1	11.66
3	0.9332	0.9529	35.35

Table 2: Performance of different detectors for Lena image with a noise density of 80%.

Algorithm	Γ_1	Γ_2	Time (s)
1	0.9998	1	39.29
2	0.9994	1	26.31
3	0.9293	0.9731	35.20

Table 3: Performance of our noise suppression proposal for Lena image.

Noise (%)	PSNR (dB)	UIQI	Time (s)
20	41.7821	0.9735	21.06
50	34.9502	0.9105	36.45
80	29.4583	0.7786	52.10

6 CONCLUSIONS

This paper has presented an algorithm for detection and suppression of salt and pepper noise in digital images. The obtained results have shown that the proposed algorithm provides good results in acceptable time. In future work, we can consider adapting this algorithm for the suppression of different types of noise and using it as part of an image pre-processing step, in order to perform tasks such as segmentation and object recognition in a robust way.

REFERENCES

Buades, A., Coll, B., Morel, J. M., 2005. A non-local algorithm for image denoising. In *proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Vol. 2, 60-65.

Chan, R. H., Ho, C. W., Nikolova, M., 2005. Salt-and-Pepper Noise Removal by Median-Type Noise Detectors and Detail-Preserving Regularization. *IEEE Transactions on Image Processing*. Vol. 14(10), 1479-1485.

Wang, Z., Bovik, A. C., 2002. A universal image quality index. *IEEE Signal Processing Letters*. Vol. 9(3), 81-84.

Wang, Z., Zhang, D., 1999. Progressive Switching Median Filter for the Removal of Impulse Noise from Highly Corrupted Images. *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing*. Vol. 46(1), 78-80.

Yuan, S. Q., Tan, Y. H., 2006. Difference-Type noise detector for adaptive median filter. *IEEE Electronic Letters*. Vol. 42(8).

PLANNING STACKING OPERATIONS WITH AN UNKNOWN NUMBER OF OBJECTS

Lluis Trilla and Guillem Alenyà
Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Llorens i Artigas 4-6, 08028 Barcelona, Spain
lluistr@gmail.com, galenya@iri.upc.edu

Keywords: Stack planification, Symbolic POMDP, Time-of-flight camera.

Abstract: A planning framework is proposed for the task of cleaning a table and stack an unknown number of objects of different size on a tray. We propose to divide this problem in two, and combine two different planning algorithms. One, plan hand motions in the Euclidean space to be able to move the hand in a noisy scenario using a novel Time-of-Flight camera (ToF) to perform the perception of the environment. The other one, chooses the strategy to effectively clean the table, considering the symbolic position of the objects, and also its size for stacking considerations. Our formulation does not use information about the number of objects available, and thus is general in this sense. Also, it can deal with different object sizes, planning adequately to stack them. The special definition of the possible actions allows a simple and elegant way of characterizing the problem, and is one of the key ingredients of the proposed solution. Some experiments are provided in simulated and real scenarios that validate our approach.

1 INTRODUCTION

Algorithms for planning explicitly considering uncertainty have been widely used in the field of mobile robots (LaValle, 2004; Thrun et al., 2005), but are less common in robotic-arm manipulation and grasping (Hsiao et al., 2007). In that scenario uncertainty is especially important and should be carefully considered because contact takes place between the robot and the world. In this interaction, the position of the object and the robot in the world cannot be precisely known, even more if we consider uncertainty in sensors we use to sense this world.

In this paper we want to explore object grasping and stacking tasks, as they are interesting and challenging skills (Kemp et al., 2007).In order to deal with these problems the *partially observable Markov decision process* paradigm will be used, specifically the *discrete model based POMDP*. It provides the capacity of dealing with uncertainty in observations and actions, usually a robot will have an approximation of reality when is sensing the environment and evaluating the results of the actions completed. POMDP have been used before in the context arm motion control for grasping, however perceptions used are simpler than here, i.e. on/off signals from pressure sensors on the fingers of the hand (Glashan et al., 2007).

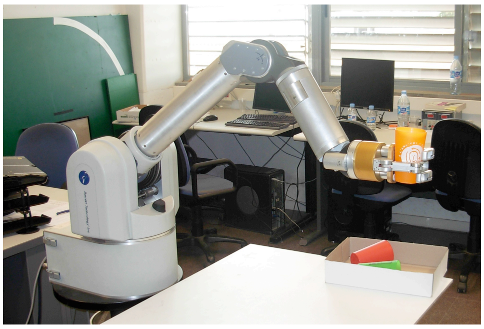


Figure 1: The robotic arm used in the experiments executing the policy computed by the planner.

The system uncertainty is modeled by measuring it in the real system and providing the values to the system, so it can take into account the various difficult situations it could face according to the chosen action. One interesting characteristic of our approach is that two different POMDPs are combined and one of them can control the other one and get feedback from it. We will apply this approach to solve a real situation: to clean a table and stack an unknown number of objects of different size on a tray.

The first POMDP will control robotic arm trajectory to prepare the grasping task, planning in the *space state* formed by the relative coordinates to the target state. This approach is extensively reported in (Trilla, 2009). Here will be briefly introduced. A naive approach to avoid the POMDP complex mechanism is a simple reactive algorithm. However, within this approach is difficult to take into account the uncertainty in the number of stacked objects and the probability of stacks falling down.

The second POMDP will plan symbolically the strategy of the cleaning task and the actions chosen for the target of the first POMDP. The objective of the second POMDP is either to completely clean the table or fill the tray. The planification is symbolic because it does not rely on the coordinates of the objects or its interaction with the world, but on an abstraction layer. Two main considerations are important. First, observations are partial: the number of objects on the table is unknown, i.e. because of possible occlusions, and some objects maybe are pre-stacked on the table and this is difficult to observe. Second, the tray surface is limited so it has to stack objects. Here the planning has to deal with objects of different size.

Perception in grasping applications is generally performed using artificial vision to recognize some object characteristics, and then plan a correct grasp (Saxena et al., 2008). Here we will use a relatively new sensor, a ToF (Time of Flight) camera. This camera delivers 3D images at 25fps, potentially allowing fast perception algorithms and, contrarily to stereo systems, it does not rely on computing depth on texture or other object surface characteristics. Depth information will be used to identify the position of the robot hand in the space, and to easily separate objects from background.

This article is structured as follows. POMDP background is introduced in Section 2. The planification strategy is introduced in Section 3, and in particular the planification of the symbolic steps that are involved in the container-content manipulation (Section 3.1). In Section 4 some experiments are presented, validating our approach in a general stacking case, in the case of different object sizes, and with occlusions between objects. Finally, Section 5 is devoted to the conclusions and future work.

2 POMDP BACKGROUND

A POMDP models a sequence of events in discrete states and time where the agent chooses actions to perform. It is represented by the tuple (S, A, T, R, O) where S is the finite set of *states* and A is a discrete set of *actions*. The *transition model* $T(s, a, s')$ describes the probability of a transition from a state s to s' when the action a is performed. The *reward model* $R(s, a)$ defines the numeric reward given to the agent when it executes an action a being in state s. *Observation model* $O(z, a, s)$ describes the probability of an observation z when the action a is performed, and the state is s. A POMDP handles partially observable environments, there is only an indirect representation of the state of the world. The *belief state b* is the probability distribution over all states in the model. At each time step the belief state is updated by Bayesian forward-filtering.

A decision about which action is most applicable is given by the policy function which contains the information about the best action to perform for any possible belief distribution. The policy balances the probabilities of a future sequence of events with the expected accumulated reward which has to maximize. Computing a policy is highly intractable with classic exact methods like *value iteration* or *policy iteration*. However, some recent work has been devoted to find approximated solutions (Hsu et al., 2007), as *point based value iteration* (Hsu et al., 2008), *discrete Perseus* or HSVI are quite fast and yield good results.

3 PLANNING AND EXECUTING THE MANIPULATION OPERATIONS

We divide the high level task of cleaning a table of several objects in two different levels. First, it is important to decide which object to first manipulate. Then, the next issue is to know the exact position of the object and effectively manipulate it. Here we will present our development for the planning algorithm which deals symbolically with the problem and performs high level task.

The effective manipulation of the objects can be solved by means of a classical control algorithms. Alternatively, we have recently proposed to solve the low level task defining also a POMDP in the space of discretized hand positions (Trilla, 2009). With this approach we are able to deal with robots with low precision or repeatability, and also with mobile robot manipulators that naturally are not exactly placed equally in front of a table.

3.1 Planning the Strategy

We define the problem as follows. The working area is divided in zones: one for the tray and the rest for the table (see Fig 2). Each zone is divided in positions where the objects can be placed. Space state is defined as the set (zo, p, sn, t), where zo (zone) and p (position) identify the object location symbolically, sn indicates the number of objects stacked in the same

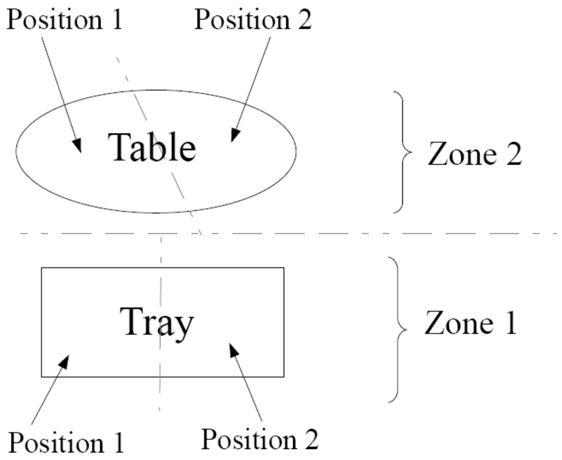


Figure 2: General schema of the working space, detailing the different labeled areas. Table can be divided in several regions in general. Here is divided in two, like in the experiments.

place, and t is the type of the objects in that place.

The actions are defined to move the objects in the following order:

1. move from the table to the first position on the tray
2. move from the table to the second position of the tray
3. stack the objects on the table freeing up a position.

Using only these three actions is possible to successfully complete the cleaning table task. When two objects are of the same size the system not needs to specify which one of the objects on the table is going to be stacked first. The transition model can be designed including any previous criteria so can be moved first the larger objects or those which are in a certain zone (the closest to the tray for instance).

We want to consider an unknown number of objects into the table. This is appropriate for modelling two different situations: when the point of view of the observer doesn't allow to view completely the whole scene, and when occlusions shadow the correct number of objects.

To take into account an unknown number of objects on the table we propose to only consider two of them at each time. The approach can be extended to take into account more objects/positions, but then the complexity of the solution also increases. Once a position on the table is empty that position is assigned to the next object. This assignment could be done by any criteria, proximity, size of the object (first the larger ones then the smaller ones), or randomly as in our experiments. This represents the action uncertainty: when an action is performed the next state can have an empty position or the transition can go to a different state with objects in that position. The objects can be found already stacked on the table, but the probability of success in such a situation decreases, reflecting that is easier to manipulate a unique object that a stack of two or more objects. In this way planning can be done with no previous knowledge about the number of the objects on the table, and lets the planner to adapt to different situations.

Observations indicate the number and type of objects stacked. The uncertainty of the observations is the number of objects stacked in each position and is modeled as a Gaussian probability distribution over the belief state. The container-content problem (which objects can be or not stacked) is related to the type of object and the symbolic POMDP has to chose the proper stacking actions maximizing the available space on the tray.

Finally, the rewards are defined as positive when the objects are placed on the tray, move objects have a cost and stacking movements have a higher cost. The aim is reached when the table is clean or the tray is full. A great advantage of this model is the versatility of its policy so it can adapt to different situations in order to reach the aim for many scenarios.

Observe that our proposal is not assuming neither implicitly nor explicitly the success in the motion operation. As usual, after the execution of each transfer action the state is observed and the appropriate pocily is chosen in accordance. Failed grasping operations are naturally handled, as the state will not change. However, if the object accidentally falls down in the middle of the operation and its position is outside the table area this particular object will not be recovered with the presented approach.

4 EXPERIMENTS

4.1 The Observation Model

For the observations we want to model a SR3000 SwissRanger camera. This sensor delivers images of 160×120 pixels with depth information for each pixel at 25fps using ToF (time-of-flight) principle. ToF cameras emit modulated light, and depth computation is based on measures on the reflected light using the well known time-of-flight principle (Kolb et al., 2008). Figure 3 shows our testing scenario as perceived by the ToF camera. We use depth to easily segment the robot and the object from the background, but depth information is an important cue that allows to compute also several of the required grasping parameters (Kuehnle et al., 2008).

Depth measures from a ToF camera are noisy, and

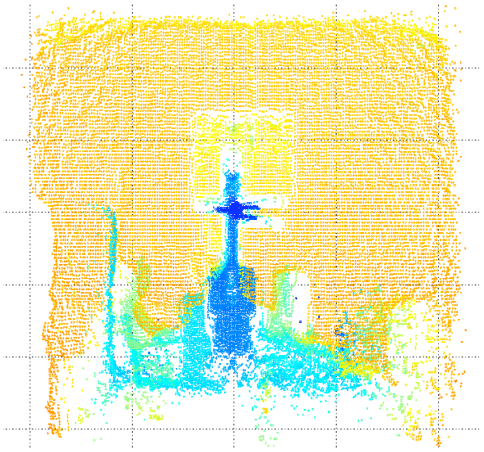


Figure 3: Depth image provided by the ToF camera and used to compute the perceptions. Colors correspond to different depths. Observe the robot horizontally centered in the image, and the hand with its three fingers in the center.

systematical errors are present and depend on the computed depth and on the position of the pixel onto the sensor. To obtain precise measures some calibration procedures have to be performed (Kahlmann et al., 2006; Lindner and Kolb, 2006). Also, some other non-systematic errors affect the camera readings, i.e. surface orientation, irregular illuminated zones. ToF camera calibration is still an active field of research and a lot of work has been done to completely understand the source of errors and compensate them.

This sensor is very interesting as offers the capability of measuring the 3D position of the robot hand as well as the position of the different objects on the table (Figure 3). It is possible without calibrating the camera-robot system to control the motion of the robot (Trilla, 2009). However, the resolution of the sensor is quite small, so it is justifiable to state that it is difficult to determine in our scenario if one glass is alone or it is stacked with other glasses.

The required uncertainties are measured using this scenario.

4.2 Symbolic Planification

Once it is defined with proper values, the symbolic POMDP has been tested with simulations facing it with several scenarios and checking its decisions and the sequence of actions. This experiment is defined with two zones and two positions in each zone. There are two types of glasses: big ones and little ones. The transitions are restricted to stack big glasses with big glasses, the little glasses can be stacked inside the big or the little glasses indifferently. The number of glasses stacked in the same position is restricted to four and only is defined the type of the top and bottom of the glass because they are involved in the container-content problem. The states are determined by all the possible combinations of this variables.

As explained before, choosing this state space the planner becomes able to deal with any quantity of objects on the table. It will not perform the optimal sequence of actions because it has not previous information about the total number of the objects. However, using this method is not necessary to recalculate the policy for different amount of objects and is not limited to a maximum number of them.

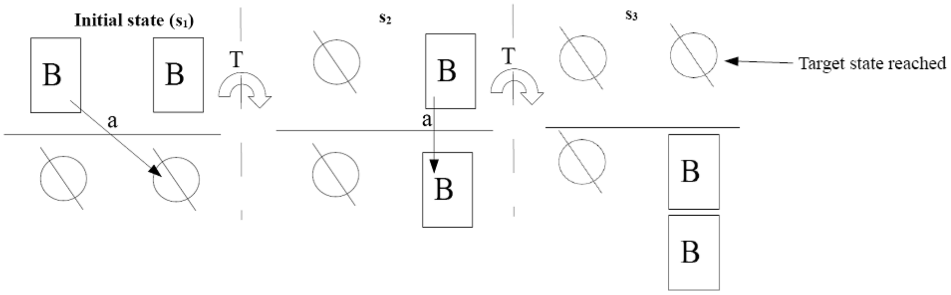
The probability of reaching any of the possible states after performing an action is defined as uniformly distributed in the transition model. Consequently, after moving a glass there is the same probability of reaching a state with an empty position or a full one. This transition is decided randomly and it gives the uncertainty over the actions results. Observations give a probability distribution for the belief state over the quantity of objects stacked in the same position, there is uncertainty about the number of glasses.

The rewards are defined as follows, stacking operations have a cost because the pile becomes more unstable, place the glasses on the tray is rewarded positively and reach the aim has a positive reward as well. Stack four or more glasses has a penalty (more than four glasses is considered the same state) because it is a too unstable pile and it has a high risk of falling over.

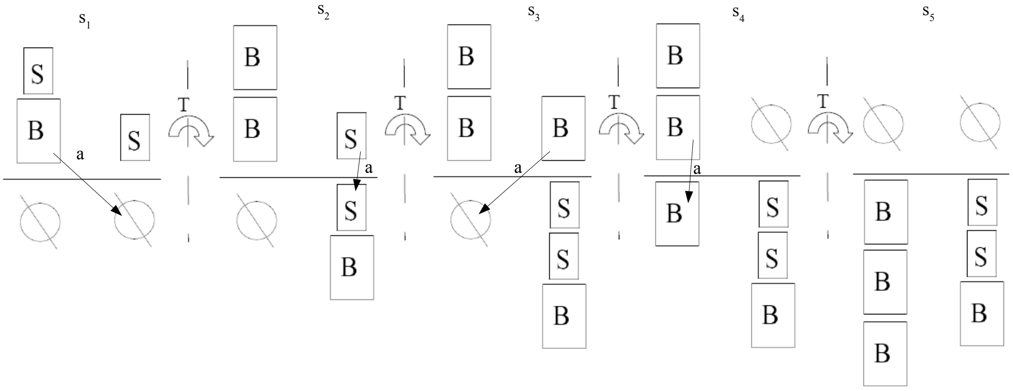
Some results of the carried out experiments are shown in Figures 4(a) and 4(b), that describe how POMDP-S solves the proposed task. In this diagrams, boxes marked with B represent big glasses and boxes with S are the small ones; empty positions are represented by a circle and the action to perform with an arrow; the top side is the table and the bottom side is the tray; states are indicated as s_i; between two states there is a transition T; objects placed together mean that they are stacked vertically.

The first diagram (Figure 4(a)) represents a simple situation where there are only two big glasses on the table. It is worth noting how the planner decides, on the last action s_3, to stack them on the tray instead of placing them separately, even knowing that this action has a higher cost. The explanation is that, as the number of objects is unknown, exists the possibility that more objects appear after the last movement (due to an occlusion), so planner decides that it is safer to keep a free space available on the tray.

In the second diagram (Figure 4(b)) there is a more complex situation, with several objects on the



(a) Simple example, where two big objects are cleaned from the table. Observe that planner decides to create a stack. This is right solution considering that only one object is observed in each zone, as new objects can be observed, i.e. due to occlusions.



(b) A complex example to demonstrate our algorithm capabilities. Regions with several objects are discovered in states s_2 and s_3, before the translations of objects to the tray. The restriction on the size of the object for stacking is correctly handled in transitions s_3 to s_4 and also s_4 to s_5 where big objects are not stacked onto a small one.

Figure 4: Diagram of two experiments results. Boxes marked with B represent big glasses and boxes with S are the small ones. Empty positions are represented by a circle and the action to perform with an arrow. The top side is the table and the bottom side is the tray. States are indicated as s_i. Between two states there is a transition T. Objects placed together mean that they are stacked vertically.

table, several of them already stacked and occluded. An example of this occlusion fact can be seen in the transition from s_1 to s_2. The agent is moving two stacked objects to an empty position on the tray, but in the position that represents this table zone two new stacked objects appear. Note that this is happens also in transition from s_2 to s_3, with a big objects appearing when a small on is moved.

The correct planification using the size of the objects in the stacking strategy can be seen in th transition from s_3 to s_4. Only big objects are present, but in the top of the stack there is a small object, so the empty tray position is preferred. The situation is similar in transition from s_4 to s_5. In our approach, if the agent reaches a state with no possible action (i.e. there exists no transition) the algorithm ends. This could happen prematurely if the objects are not stacked properly. In the next section we will envisage some possible improvements to handle this behavior.

5 CONCLUSIONS AND FUTURE WORK

We have presented a planning framework, combining two different POMDP approaches, that effectively solve the task of cleaning a table and stack an unknown number of objects of different size on a tray

The symbolic POMDP chooses the adequate combination of actions to clean the table. Here the state space design is crucial as we have some restrictions. First, we assumed explicitly that the number of objects on the table was not observable, i.e. due to occlusions between objects, because it is hard to estimate the number of objects when they are stacked, or purely by noise in the observations. Second, the solution to the problem implies stacking objects of different size with some restrictions, i.e, a big object cannot be stacked onto a small one.

Thanks to the proposed space state and action space, the codification of the different positions and sizes of the glasses lets the planner generate a policy able to deal with the container-contents problem stacking the glasses in the best way, maximizing the free space on the tray. The solution we propose to be able to handle the unknown number of objects is based on focus the attention on only one of the objects present in the each one of the different regions on the table, and it has turned out to be particularly adequate and powerful.

5.1 Future Work

Here we have considered that the objects can be already stacked on the table, and that the the robot can perform stacking actions on the table. However this condition has hardly raised in our experiments. One challenge we are facing now is to incentive object stacking actions on the table, before putting them in the tray. A new variable is needed here to balance the cost between two transportation actions and one stacking plus a transportation operation. In the computation of this cost the trajectory from the table to the tray in the transportation action becomes important, as we want to stack closer objects, or more important, stack objects that are in the transportation trajectory to the tray. Our formulation is general in the number defined zones on the table, so we face this new condition as a natural extent of the presented algorithm.

We have considered to add an additional action when observations are not enough to decide for one action, i.e. in order to gather information about the number of glasses stacked on each position. A promising option is an asking action to an operator where the answer could modify the agent's belief state (Armstrong-Crews and Veloso, 2007).

ACKNOWLEDGEMENTS

This work has been partially supported by the Spanish Ministry of Science and Innovation under project DPI2008-06022 and the Generalitat de Catalunya under the consolidated Robotics Group. G. Alenyà was supported by the CSIC under a JAE-Doc Fellowship.

REFERENCES

Armstrong-Crews, N. and Veloso, M. (2007). Oracular partially observable markov decision processes: A very special case. In *Proc. IEEE Int. Conf. Robot. Automat.*, Rome., pages 2477–2482.

Glashan, R., Hsiao, K., Kaelbling, L. P., and Lozano-Pérez, T. (2007). Grasping POMDPs: Theory and experiments. In *RSS Workshop: manip. for human env.*

Hsiao, K., Kaelbling, L. P., and Lozano-p'erez, T. (2007). Grasping POMDPs. In *Proc. IEEE Int. Conf. Robot. Automat.*, Rome., pages 4685–4692.

Hsu, D., Lee, W., and Rong, N. (2007). What makes some POMDP problems easy to approximate? In *Advances in Neural Information Processing Systems (NIPS).*

Hsu, D., Lee, W., and Rong, N. (2008). A point-based pomdp planner for target tracking. In *Proc. IEEE Int. Conf. Robot. Automat.*, Pasadena.

Kahlmann, T., Remondino, F., and Ingensand, H. (2006). Calibration for increased accuracy of the range imaging camera Swissranger^{TM}. In *ISPRS Commission V Symposium*, pp. 136–141, Dresden.

Kemp, C., Edsinger, A., and Torres-Jara, E. (2007). Challenges for robot manipulation in human environments. *IEEE Robot. Automat. Mag.*, 14(1):20–2.

Kolb, A., Barth, E., and Koch, R. (2008). ToF-sensors: New dimensions for realism and interactivity. In *Proc. IEEE CVPR Workshops*, vol. 1-3, pp. 1518–1523.

Kuehnle, J. U., Xue, Z., Stotz, M., Zoellner, J. M., Verl, A., and Dillmann, R. (2008). Grasping in depth maps of time-of-flight cameras. In *Proc. Int. Workshop Robotic Sensors Environments*, pp. 132–137.

LaValle, S.M. (2004). *Planning Algorithms*. Cambridge UP

Lindner, M. and Kolb, A. (2006). Lateral and depth calibration of PMD-distance Sensors. In *Proc. 2nd Int. Sym. Visual Computing*, vol. 4292, pp. 524–533.

Saxena, A., Driemeyer, J., and Ng, A. Y. (2008). Robotic grasping of novel objects using vision. *Int. J. Robot. Res.*, 27:157–173.

Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic Robotics*. MIT Press, Cambridge.

Trilla, L. (2009). Planificació de moviments en entorns amb incertesa per a manipulació d'objectes. Master's thesis, Universitat Politècnica de Catalunya.

MOTION GENERATION FOR A HUMANOID ROBOT WITH INLINE-SKATE

Nir Ziv*, Yong Kwun Lee* and Gaetano Ciaravella**
**Center for Cogntive Robotics, University of Science and Technology (UST) & Korea Institute of Science and Technology (KIST), 39-1 Hawolgol-Dong, Wolsong-Gil 5, Seongbuk-gu, Seoul, South Korea*
***Cognative Center for Robotic Research, Korea Institute of Science and Technology (KIST), Seoul, South Korea*
{nirziv, yklee, cga}@kist.re.kr

Keywords: Humanoid, Inline-Skate, Skating Motion, Wheeled Locomotion.

Abstract: A lot of research has been done for bipedal walking and many positive results have been produced, such as the ASIMO robot from Honda. However, although bipedal walking is a good solution for moving over uneven surfaces; bipedal walking is inefficient over an even surface because the robot's walking speed and stability are limited. Consequently, employing a wheeled locomotion on even surfaces can be advantageous. This paper presents a mathematical model and simulation of wheeled biped robot with two passive wheels on each foot. This enables the robot to move more efficiently over even surfaces. Also, this paper attempts to produce a more human-like inline-skating motion than previously created inline-skating simulations.

1 INTRODUCTION

Over the last few years, research on bipedal walking has progressed and has lead to major successes within the robotics field. One good example of this success is the ASIMO robot, a Honda creation. At the time of writing this paper, ASIMO can run with a speed of 6 km/h and is able to walk at a speed of 2.7 km/h (Honda). When considering the strongest achievements which leg locomotion has achieved until now, one major progressive accomplishment has been the ability for robots to move across uneven surfaces.

However, when considering robotic leg locomotion across only even tarains, the advantages of wheeled locomotion are greater than bipedal (or leg) locomotion. The reason for this is because the wheeled robot's moving speed and efficiancy are stronger and more stable. This is because leg locomotion algorithms are more complex than wheeled motion, making it more difficult for bipedal robots to walk or run quickly. (Jo et al 2008).

Therefore, to combine the advantages of both leg and wheeled locomotion, hybrid robot systems were developed. For example, WorkPartner is a quadruped system that can detect the type of surface it is on, even or uneven, and can then choose accordingly the type of locomotion to use, wheeled or bipedal (Ylonen et al 2002). Another example of hybrid locomotion is WS-2/ WL-16 (Waseda Shoes – Number 2 / Waseda Leg – Number 16) (Hashimoto et al 2005) which also combines bipedal walking with wheeled locomotion. Both WorkPartner and WS-2/WL-16 use DC motorized wheels.

However, because both WorkPartner and WS-2/WL-16 utilize motor wheels, this means that the brake and steering are also motorized. As a result, a robot with motorized wheels has larger wheels and its robotic system becomes heavier and, therefore, less agile.

To solve this issue, passive wheel locomotion has been suggested. One example of a hybrid, passive wheeled robot is the Roller-walker, developed by Hirose and Takeuchi (Hirose et al 1996, 1999, 2000.) In this system, the wheel motion is not generated by a DC motor, instead the robot moves by making a roller skating-like motion. Another example of passive wheel locomotion is the Rollerblader (Chitta et al 2003), which has two passive wheels that move in symmetric and anti-symmetric motions to propel the Rollerblader forward and in a rotary motion respectively.

Similarly, in this paper we present a simulation of a passive wheeled robot to generate a forward motion; our robot consists of two legs with two passive wheels attached along the middle of each foot. However, unlike the Rollerblader, our simulation and model will attempt to create a more human-like motion using 'D' shape movement for the pushing leg where near the end of the movement the pushing leg will be raised in the air.

This paper is organized as follows. The following section will describe the modelling and the equations of the robot's motion. In section 3, we will show the simulation, motion steps and results. We will conclude with a brief discussion about the simulation's results, as well as problems that need to be solved within further research in the future.

2 MODELLING

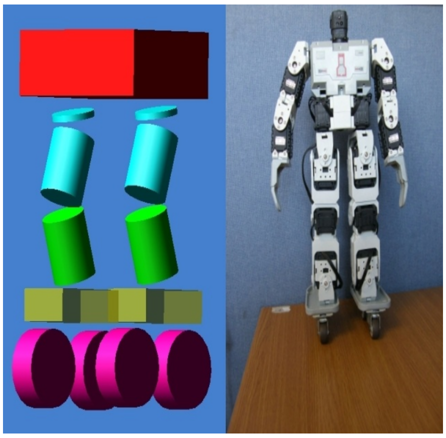


Figure 1: MSC.ADAMS Model and Read Small Size Humanoid.

The modelling of the robot (figure 1) was done by using MSC.ADAMS software. The robot has a total of 12 Degrees Of Freedom (DOF), where each leg contains 6 DOF. Two passive wheels were added to each foot along the middle of the foot.

The forces acting on the robot configuration (figure 2) and a derivation of the equations of motion are shown and explained in more detail in the next section.

2.1 Parametric Modelling

Solving the equations of motion for this system is a very complex process; therefore, the following few assumptions were made:

- Even though the motion is continuous, we will solve it for a time equal to t (where t is a segment of the whole motion.)
- External force, such as air resistance, external push and wind are negligible.
- The simulation motion is a forward motion. This will reduce the amount of forces acting on the model for rotational motion.

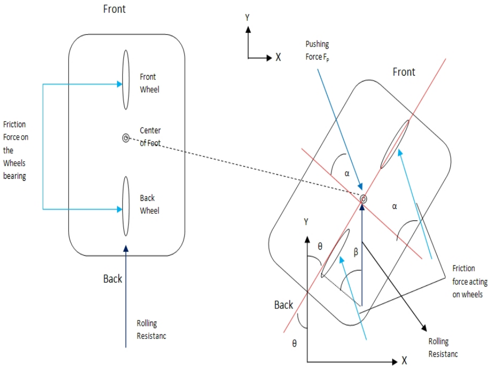


Figure 2: Forces Acting on the Robot.

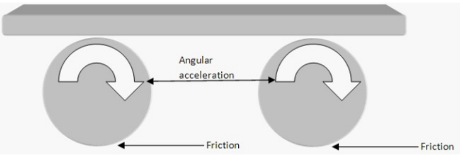


Figure 3: Side View of the Forces Acting on the Robot.

Because we are trying to simulate a forward motion, the forces acting on the model should satisfy the following equations:

$$\sum F_x = 0 \tag{1}$$

$$\sum \mathrm{F_y} = \mathrm{M_t} * \ddot{\mathrm{y}} \tag{2}$$

Where F_x and F_y are the forces acting on the X axis and Y axis respectively. M_t is the total mass and $\ddot{y}$ is acceleration. As figure 2 shows, we will calculate the forces that act on the model.

First, we will analyze the stationary foot. Figure 2 shows the robot's stationary foot as the left foot.

$$\begin{gathered} F_r = C_{rr} * N \\ \mathrm{N} = \mathrm{M_t} * \mathrm{g} \end{gathered} \tag{3}$$

Where F_r, C_{rr} and N are the rolling resistance force, the rolling resistance coefficient and the normal force respectively (Peck et al 1859,National Research council 2006) and, in this instance, the normal force is total mass (M_t) times gravity (g).

$$\begin{gathered} F_{fs} = \mu_f * N \\ N = M_t * g \end{gathered} \tag{4}$$

Where F_{fs} is the frictional force on the stationary foot, μ_f is the friction coefficient, N is the normal force, M_t is the total mass and g is the gravitational force.

Because the stationary foot will always point forward (there is no Z axis rotation) and the foot will be perpendicular to the ground, the rolling resistance force will always act on the Y axis. The frictional force from equation (4) will be used for the wheels' momentum as described next.

The last force acting on the stationary foot is the wheels momentum force. This can be seen in figure 3. The force acting on the wheels can be expressed as follows:

$$MO_w = I * \ddot{\theta} - F_{fs} * r \tag{5}$$

MO_w is single wheel momentum, I is the inertial mass, $\ddot{\theta}$ is angular acceleration F_{fs} is frictional force on the stationary foot and r is the wheel radius. Because we have two wheels on each foot, we can express equation (5) as follows:

$$MO_t = 2 * \left(\frac{I*\ddot{\theta}}{r} - F_{fs}\right) \tag{6}$$

MO_t is the total momentum force acting on the stationary foot's wheels.

Until now we have described the forces which are acting on the stationary foot (or the left foot from figure 2). Now we will describe in the same manner the forces acting on the pushing foot (or the right foot from figure 2).

First, we will look at the pushing force. The pushing force will act on the pushing leg with an angle of α. This angle will vary with time. However, as stated earlier to simplify the equations, we assume that we are looking at a given time t and therefore remove the dependency of time.

Because the pushing force is acting with an angle, we will need to derive the X and Y component of this force. These components are as follows:

$$\begin{aligned} F_{px} &= F_p * \cos\alpha \\ F_{py} &= F_p * \sin\alpha \end{aligned} \tag{7}$$

F_{px} and F_{py} are the pushing forces on the X axis and the Y axis respectively. F_p is the pushing force and α is the angle between the pushing force and the pushing foot.

As a reaction force to the pushing force, the pushing foot will experience a frictional force as well. This force can be written as:

$$\begin{aligned} F_{fp} &= \mu_f * N \\ N &= M_t * g + F_p * \sin\alpha + F_r * \cos\beta \end{aligned} \tag{8}$$

As with the pushing force, we will need to work on the X axis and Y axis components of the friction force. Using equation 8 with angle α produces the following equations:

$$\begin{aligned} F_{fpx} &= F_{fp} * \cos\alpha \\ F_{fpy} &= F_{fp} * \sin\alpha \end{aligned} \tag{9}$$

Where F_{fpx} is the frictional force on the pushing foot on the X axis and F_{fpy} is on the Y axis.

Using equations (1) and (2) and summing all the forces that act on the X axis and Y axis, we can rewrite equations (1) and (2) as follows:

$$\begin{gathered} \sum F_x = 0 \\ \left[F_p - F_{fp}\right] * \cos\alpha = 0 \end{gathered} \tag{10}$$

$$\begin{gathered} \sum \mathrm{F_y} = \mathrm{M_t} * \ddot{\mathrm{y}} \\ MO_t + F_r + \left[F_p - F_{fp}\right] * \sin\alpha = M_t * \ddot{y} \end{gathered} \tag{11}$$

Using equations (3) – (9) and solving equation (11) for $\ddot{y}$, we can rewrite equation (11) as follows:

$$\mathrm{MO_t} + \mathrm{F_r} + \left[\mathrm{F_p} - \mathrm{F_{fp}}\right] * \sin\alpha = \mathrm{M_t} * \ddot{\mathrm{y}} \tag{12}$$

$$\frac{\mathrm{MO_t}}{\mathrm{M_t}} + \frac{\mathrm{F_r}}{\mathrm{M_t}} + \frac{\mathrm{F_p}*\sin\alpha}{\mathrm{M_t}} - \frac{\mathrm{F_{fp}}\sin\alpha}{\mathrm{M_t}} = \ddot{\mathrm{y}} \tag{13}$$

$$\begin{gathered} \mathrm{F_p}\left[\frac{\sin\alpha - \mu_f*\sin\alpha*\sin\alpha}{\mathrm{M_t}}\right] + \frac{2*\mathrm{I}*\ddot{\theta}}{\mathrm{M_t}*\mathrm{r}} + \mu_f * \\ \mathrm{g}\left[-2 + \frac{\mathrm{C_{rr}}}{\mu_f} - \sin\alpha - \mathrm{C_{rr}} * \cos\theta * \sin\alpha\right] = \ddot{\mathrm{y}} \end{gathered} \tag{14}$$

$$\begin{gathered} \mathrm{F_p}\left[\frac{\sin\alpha - \mu_f*\sin\alpha*\sin\alpha}{\mathrm{M_t}}\right] + \mathrm{C_1} * \ddot{\theta} + \\ \mathrm{C_3}[-2 + \mathrm{C_2} - \sin\alpha - \mathrm{C_{rr}} * \cos\theta * \sin\alpha] = \\ \ddot{\mathrm{y}} \end{gathered} \tag{15}$$

$$\begin{gathered} \ddot{y} = \frac{F_p*(\sin\alpha - \mu_f*\sin\alpha*\sin\alpha)}{M_t} + C_1\ddot{\theta} - \\ C_2[2 - C_3 + \sin\alpha + \mathrm{C_{rr}} * \cos\beta * \sin\alpha] \end{gathered} \tag{16}$$

Where $C_1 = \frac{2*I}{M_t*r}$, $C_2 = \mu_f * g$ and $C_3 = \frac{\mathrm{C_{rr}}}{\mu_f}$. For a full derivation of how we developed equation (12), please refer to the appendix.

2.2 Model Analysis

From equation (16) above, we notice that in order to make the acceleration higher, and therefore make the robot move forward faster, we needed to identify a few components of the equation and check the values that qualify our needs.

Equation (16) can be written in simple form as follows:

$$\ddot{y} = A + B * \ddot{\theta} - C \tag{17}$$

Where:

$$A = \frac{F_p * (\sin\alpha - \mu_f * \sin\alpha * \sin\alpha)}{M_t} \quad (18)$$

$$B = \frac{2 * I}{M_t * r} \quad (19)$$

$$C = -C_2[2 - C_3 + \sin\alpha + \mathrm{C_{rr}} * \cos\beta * \sin\alpha] \quad (20)$$

In order to make ÿ bigger, we must make A and B bigger and make C as small as possible. To make equation (18) bigger, we identified α as the variable since $F_{p,}$ M_t and μ are constants. Using this notion, we plotted equation (18) into Matlab. The results are shown in figure 4.

As can be seen in figure 4, which shows when α is static and not dynamically changed by time, the value of α that will be the best in order to make the pushing force greatest (i.e. part A) is α = $\frac{\pi}{2}$ or ,in other words, when α = 90°. At the same time, the best value for μ is 0.2. When looking in detail at equation (20), we identified α and β as the variables because the rest of equation (20) is constant. Because we already determined the value of α from equation (18), we will need to find the best value of β. Using the same value of α with β, we will succeed to minimize equation (20) since cos (90°) will make the last part of C equal to zero.

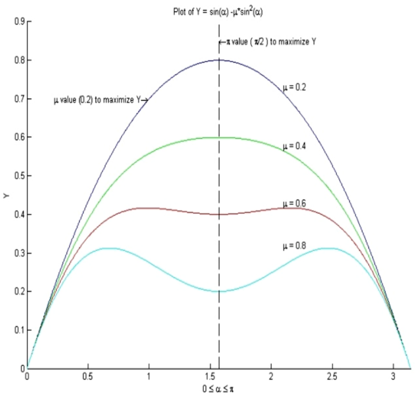


Figure 4: Matlab Plotting the relationship between values of μ and α to maximize pushing force.

The middle part of equation (18) is shown in equation (19) and has been plotted in Matlab; it can be seen in figure 5. In figure 5, we showed the relationship between the angular acceleration, $\ddot{\theta}$, and the robot acceleration, ÿ. As expected, the relationship between the two accelerations is linear, which means that the faster the angular acceleration on the wheels will be, the faster the robot acceleration will be. Also, we verified the equations with different values of the pushing force.

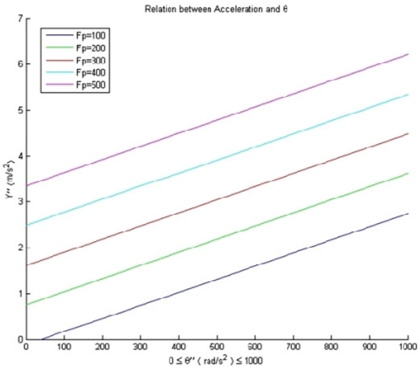


Figure 5: Matlab plotting the relationship between acceleration and angular acceleration.

3 SIMULATION

We used MSC.ADAMS to simulate the skating motion. The model is a simple model of the real humanoid (see figure 1) and the motions were generated by using three general motions that act on the model (one general motion on the hip with relation to the ground, one general motion on the left leg with relation to the hip and the last general motion on the right leg with relation to the hip as well). With these three general motions we controlled the angles or the joint and displacements of the rigid bodies. The skating motion is combined of six steps, three for the right foot, and three for the left foot to produce a full cycle of motion.

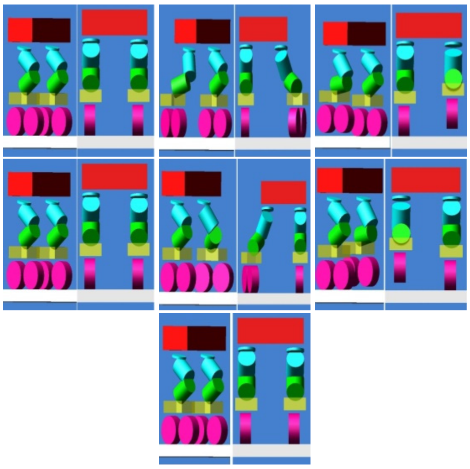


Figure 6: The Simulation Algorithm Sequence. The left image in each square shows the view from the side and the image on the right shows the view from the back.

The shape of the pushing leg motion is like a 'D' shape, starting from the top left corner and can be viewed in two parts. In the first part, the pushing leg is on the ground and is moving in slow motion. In the second part, the pushing leg–is in the air and to maintain balance this step part is faster than the first part of the cycle.

The following are the motion steps (assuming that we begin with the right foot):

1. To start, the right foot is pushed away from the body, creating a ‘D’ shape
2. Before the ‘D’ is completed, the right foot is returned to the ground and a triangle shape between the pushing leg, body and stationary leg is formed.
3. Then the pushing leg is lifted and returned to its starting position, parallel to the stationary leg.
4. Step 1 is repeated, except that the left foot replaces the right foot.
5. Before the ‘D’ is completed, the left foot is returned to the ground and a triangle shape between the pushing leg (this time it is the left leg), body and stationary leg (this time it is the right leg) is formed.
6. Then the pushing leg is lifted and returned to its starting position, parallel to the stationary leg.

One can note that steps 4-6 are a mirror of steps 1-3. Figure 6 shows the images of these 6 steps. Notice the top right image from figure 6 where the triangle between the pushing leg, body and stationary foot is formed.

The time interval for the simulation is 5 seconds with 100 frames. During that simulation the skating motion (steps 1-6) ran twice, this means that each leg creates the pushing force twice. The result of this run was that the robot created enough pushing force to overcome the friction and succeeded to create a forward motion.

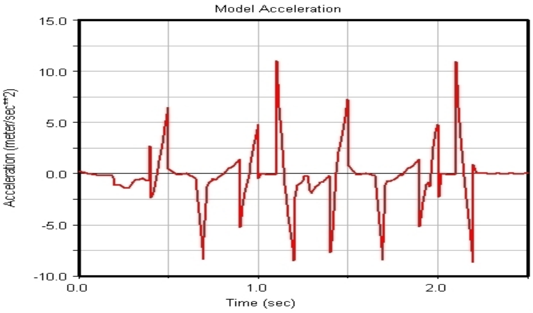


Figure 7: ADAMS Model Acceleration.

Figure 7 and figure 8 are showing the acceleration and angular acceleration of the model and a single wheel of the model respectively. These plots were taken from MSC.ADAMS software. Comparing these values to our Matlab plotting in figure 4 and figure 5 we can see that the results in both MSC.ADAMS simulation and our derivation of the equations of motion agree with one another. Both results show that as the acceleration increases the angular acceleration on the wheels will increase as well.

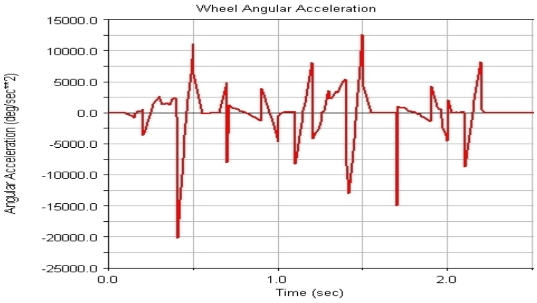


Figure 8: ADAMS Wheel Angular Acceleration.

4 CONCLUSIONS

This paper presents a simulation model with defined motion steps to generate a human-like skating motion. The simulation uses a robot which has two passive wheels on each foot along the middle of the foot. We successfully simulated the skating motion and generated a forward motion.

We also calculated the equations of motions for this model, and used Matlab to find a feasible value for dependent variables. The results were compared between the simulation and the mathematical model successfully. Even though this was a successful simulation, our next step will be to generate this motion in a real humanoid that follows the model, as can be seen in figure 9, figure 10 and figure 11. Additionally, researching a sounder algorithm to produce a more smooth ‘D’ shape would be valuable. Finally, further research would be to create a humanoid based on our model which would have the ability to transform itself between wheeled locomotion and leg locomotion depending on the surface it is crossing (even or uneven).

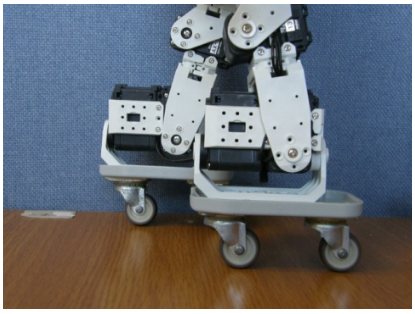


Figure 9: Small Size Humanoid Wheels.

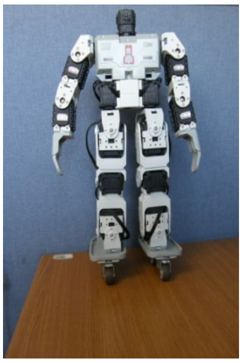


Figure 10: Small Size Humanoid Front View.

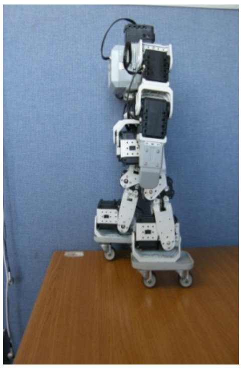


Figure 11: Small Size Humanoid Side View.

REFERENCES

Chitta S., Kumar V., 2003. Dynamics and Generation of Gaits for a Planar Rollerblader. *International Conference of Intelligent Robots and Systems*.

Endo G., Hirose S., 1999. Study on Roller-Walker (System Integration and Basic Experiments.) *International Conference on Robotics and Automation*.

Endo G., Hirose S., 2000. Study on Roller-Walker (Multi-mode Steering control and Self-contained Locomotion.) *International Conference on Robotics and Automation.*

Hashimoto K., Hosobata K, Sugahara Y., Mikuriya Y, Lim H., Takanishi A., 2005. Realization by Biped Leg-wheeled Robot of Biped Walking and Wheel-driven Locomotion. *International Conference on Robotics and Automation.*

Hirose S., Takeuchi H., 1996. Study on roller-walker (Basic Characteristics and its Control.) *International conference on Robotics and Automotion.*

Honda, ASIMO, January, 13th 2010. *The New ASIMO – Major Features Summary*. http://www.hondauk-media.co.uk/uploads/presspacks/bf27134f6692b1c050d8ae9c29bc21840af2a723/Major_Features_Summary.pdf

Jo S., Chu J., Lee Y., 2008. Motion Planning for Biped Robot with Quad Roller Skates. *International conference on Control, Automation and System*.

National Research Council Special Report 286, 2006. Tires and Passengers Vehicle Fuel Economy. Washington D.C.

Peck William Guy , 1859. Element of Mechanics: For the Use of Colleges, Academics and High Schools. A.S. *Barnes & Burr. New York*

Ylonen S. J., Halme A. J., 2002. WorkPartner – Centaur like Service Robot. *Intl. Conference on Intelligent Robots and Systems*.

TOWARDS HUMAN INSPIRED SEMANTIC SLAM

Dominik Maximilián Ramík, Christophe Sabourin and Kurosh Madani
Signals, Images, and Intelligent Systems Laboratory (LISSI / EA 3956), Université Paris-Est
Senart Institute of Technology, Avenue Pierre Point, 77127 Lieusaint, France
{dominik.ramik, sabourin, madani}@univ-paris12.fr

Keywords: SLAM, Semantics, Humanoid robotics, Human inspired, Image segmentation, Scene interpretation.

Abstract: Robotic SLAM is attempting to learn robots what human beings do nearly effortlessly: to navigate in an unknown environment and to map it in the same time. In spite of huge advance in this area, nowadays SLAM solutions are not yet ready to enter the real world. In this paper, we observe the state of the art in existing SLAM techniques and identify semantic SLAM as one of prospective directions in robotic mapping research. We position our initial research into this field and propose a human inspired concept of SLAM based on understanding of the scene via its semantic analysis. First simulation results, using a virtual humanoid robot are presented to illustrate our approach.

1 INTRODUCTION

In mobile robotics, the ability of self-localization is crucial. In fact, knowing precisely where the robot is and what kind of objects surround it in a given moment enables it to navigate autonomously. An informal definition of the Simultaneous Localisation And Mapping (SLAM) says it as a process, in which a mobile robot explores an unknown environment, creates a map of it and uses it simultaneously to infer its own position. The real environment is usually complex and dynamic and it is not easy to interpret. This complexity makes SLAM a challenging task. A comprehensive list of nowadays most common SLAM techniques can be found in (Durrant-Whyte, et al., 2006a), (Durrant-Whyte, et al., 2006b) or (Muhammad, et al., 2009). Although from its beginning a significant advance has been achieved (Thrun, et al., 2008), SLAM is not yet a solved problem. Performing SLAM in dynamic environment (Hahnel, et al., 2003) or understanding the mapped environment by including semantics into maps (Nüchter, et al., 2008) are the actual challenges.

In this paper, the state of the art in SLAM is investigated. A relatively new field of research is identified, which is attempts to perform SLAM with the aid of semantic information extracted from sensors. As one of the research interests of our laboratory (LISSI) is autonomous robotics notably in relation to humanoid robots, we are convinced that the research on semantic SLAM will bring a useful contribution. We position our initial research into this field, drawing our inspiration from the human way of navigation. Contrary the precise and "global" approach to most current SLAM techniques, the human way of doing is based on very fuzzy description of the world and it gives preference to local surroundings of the navigation backdrop. A simulation using a humanoid robot Nao is presented to demonstrate some of the proposed ideas. The real Nao will be used in our further work.

The paper is organized in the following way: section 2 focuses on the state of the art in semantic SLAM. In the third section, our approach to image segmentation and scene interpretation is discussed. Section 4 gives an overview of our robotic humanoid platform. The fifth section presents our initial results and the paper concludes with section 6.

2 SEMANTIC SLAM

One of the latest research directions on the field of SLAM is the so-called semantic SLAM. The concept may be perceived as being very important for future mobile robots, especially the humanoid ones, which will interact with humans and perform tasks in human-made environment. In fact, it is this interaction, which is one of important motives for employing semantics in robotic SLAM as humanoids are particularly expected to share the living space with humans and to communicate with them.

One way of adding semantics to SLAM may be the introduction of human spatial concepts into maps. Humans usually do not use metrics to locate themselves but rather object-centric concepts ("I am near the sink" and not "I am on [12, 59]") and they fluently switch between reference points rather than using global coordinates. Moreover, the presence of certain objects is often an important clue to recognize a place. This problem is addressed in (Vasudevan, et al., 2007). Here, the world is represented topologically; place recognition is performed based on probability of presence of objects in an indoor environment. The work shows a study aimed to understand human concepts of place recognition. It proposes that humans understand places by presence or absence of significant objects. Place classification by presence of objects has been used by (Galindo, et al., 2005), where low-level spatial information is linked to high-level semantics. Their robot has interfaced with humans and performed tasks based on high-level commands involving robots "understanding" of the meaning of place names for path planning. However, object recognition is black-boxed here. In (Persson, et al., 2007) a system is developed to map an outdoor area, generating a semantic map with buildings and non-buildings labelled. In (Nüchter, et al., 2008), a more general system is presented with a robot equipped by a 3D laser scanner evolving in an indoor environment and constructing a 3D semantic map. The processing is based on Prolog clauses enveloping pre-designed prior knowledge about the environment enabling the robot to reason about the environment. In (Ekvall, et al., 2006), object recognition is performed by a robot equipped by a laser range finder and a camera. A semantic structure is extracted from the environment and integrated to robots map. Another semantic mapping technique is shown in (Meger, et al., 2008) including an attention system.

3 IMAGE SEGMENTATION AND SCENE INTERPRETATION

Section 2 showed the pertinence of semantic SLAM for state of the art robotic mapping. It is this field, on which we are focusing our research. Our motivation comes from the natural ability of human beings to navigate seamlessly in complex environments. To describe a place, we use often very fuzzy language expressions and approximation (see (Vasudevan, et al., 2007)) in contrast to current SLAM algorithms. An interesting point is that people are able to infer distance of an object using its apparent size and their experience of object's true size. Recognition of objects and understanding their nature is an integral part of "human SLAM". We believe that application of semantics and human inspired scene description could bring a considerable benefit in development of robust SLAM applications for autonomous robotics.

For scene interpretation, the image has to be segmented first. Although many image segmentation algorithms exist (see (Lucchese, et al., 2001) for a reference), not all are suitable for mobile robotics due to need of real-time processing. We implement a fast algorithm that breaks the input image into parts containing similar colors with less attention to the brightness. We have chosen the YCbCr color model with Y channel dedicated to the luminance component of the image and other two channels Cb and Cr containing respectively the blue and the red chrominance component. Unlike RGB, the YCbCr model separates the luminance and the color into different channels making it more practical for our purposes.

Our algorithm works in a coarse-to-fine manner. First, the contrast is stretched and median filter is applied to the Cr and Cb components. Then the first available pixel not belonging to a detected component is chosen as a seed point. Eq. 1 captures how a seed point is used to extract the segment of interest (S). P stands for all the pixels in the image, whereas p is the actually examined pixel. Predicate C is true if its arguments (p, p_s) are in four-connectivity. I stands for the pixel's intensity. Seed pixel is denoted by p_s. A pixel of the image belongs to the segment S if the difference of intensities of the current and the seed pixel is smaller than a threshold and there exists a four-connectivity between it and the seed pixel

$$\forall p \in P;\ C(p, p_s)\ \&\ |I(p) - I(p_s)| < \varepsilon \rightarrow p \in S\ . \qquad (1)$$

Using this on both chroma sub-images we obtain segments denoted as S_{Cr} and S_{Cb}. A new segment S is then obtained following Eq. 2 as the intersection of segments found on both chroma sub-images without pixels already belonging to an existing segment

$$S = S_{Cr} \cap S_{Cb} - S_{all}. \qquad (2)$$

At the end of the scan, a provisory map of detected segments is available, but the image is often oversegmented. In the second step, all the segments are sorted by their area and beginning with the largest one the segmentation is run again. This time the seed point is determined as the pixel from the skeleton whose distance to its closest contour pixel is maximal. By this step, similar segments from the previous step are merged. The ultimate step is construction of a luminance histogram of each segment. If multiple significant clusters are found in the histogram, the segment is broken-up accordingly to separate them.

Now, the segments are labeled with linguistic terms describing their adjacency to each other horizontal and vertical position and span on the image. The average color, its variance and the compactness (Q) of the segment is computed following Eq. 3, where n denotes the area of the segment and o the number of contour pixels.

$$Q = 4\pi n / o^2. \qquad (3)$$

These features are used in a set of linguistic rules - the prior knowledge about the world. The aim is to determine the nature of segments and their appurtenance to an object of the perceived environment. E.g. a compact segment found in mid-height level surrounded by the wall is considered as a "window", small compact segments adjacent to the floor are denoted a "box", wide span grayish segment adjacent to the ceiling is labeled a "wall" etc.

4 NAO, THE HUMANOID ROBOT

The robotic platform we use is described in this section. It is based on Nao, a humanoid robot manufactured by Aldebaran Robotics[1]. The robot is about 58cm high with weight slightly exceeding 4kg with 25 DOF. Among others sensors it is equipped with two non-stereo 640x480px CMOS cameras. For simulations, a virtual version of Nao is available for the Webots simulation program developed by Cyberbotics[2]. For development purposes, we have chosen URBI language created by Gostai[3] and aimed specially to robotics. It allows fast development of complex behaviours for robots and provides a simple way of managing parallel processes. LibURBI connectors allow user to develop own objects using so called UObject architecture and to plug them into the language. These objects can be developed in C++, Matlab or Java code. For the demo simulation presented in the next section, we used the simulated robot mentioned above and we are going to use the real one in our further research.

The task itself may be not perceived as being strictly specific for humanoid robots. However, the motivation to use humanoid robots comes from the fact, that they are specially designed with the aim to interact with humans and to act in a human-made environment. The concepts we are exploiting here come from human approach to navigation and orientation in the space, thus embedding such human inspired semantic SLAM capabilities onto a humanoid robotic platform seems pertinent to us.

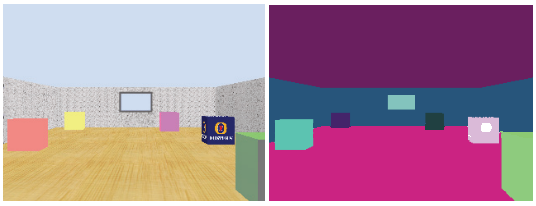


Figure 1: A view of the robot's random walking sequence. The left image is the original one. The right one shows segments detected during the segmentation phase.

5 RESULTS

As a demonstration of some of the mentioned principles, we present a simulation using Webots, where a virtual Nao is walking through a room with objects (cubes) of different colors inside it. The YCrCb image, acquired by Nao's front camera is segmented using our fast segmentation algorithm described in the precedent section (see Fig. 1). The processing speed is several tens of ms for a 320x240 frame on a 2GHz CPU Intel C2D.

After having the image segmented, all segments are labeled and interpreted by a set of prior knowledge rules. Segments can be even merged using these rules to cope with partial occlusions. The "semantic" information is used to approximate the actual distance of objects. Having an object of type "window", its typical size is looked up in the memory (at this stage, the dimensions are known a-priori as the actual learning of object sizes is supposed to be addressed in the future work). The size information is used along with the apparent size of the object to compute its approximate distance (see Fig. 2). This is described by Eq. 4 (simplified for horizontal size only). The distance d to an object is the product of estimated real width w_{real} of the object and tangent of its width in pixels w_{px} on the image multiplied by fraction of the horizontal field of view φ and the width w_{img} of the image in pixels

$$d = w_{real} * \tan (w_{px} * \varphi / w_{img}) . \qquad (4)$$

[1] http://www.aldebaran-robotics.com

[2] http://www.cyberbotics.com/

[3] http://www.gostai.com/

The aim of this calculation is absolutely not to infer the exact distance of an object, but rather to determine whether it is “far” or “near” in the context of the world. This can help in further process of creation of the map of the location. Resigning to precise metric position of every object in the mapped world and replacing it only by rough metric and human expressions like “near to” or “beside of” is believed to enable us to create faster and more robust algorithms for SLAM. Using “object landmarks” to navigate in an environment is certainly more meaningful that using e.g. simple points as in case of classical SLAM.

Precise metric information of course has still its role here, but only in some specific cases like close obstacle avoidance or disclosure to grasp an object and notably when the robot is learning typical sizes of objects to enable inference of their distance when they are seen again.

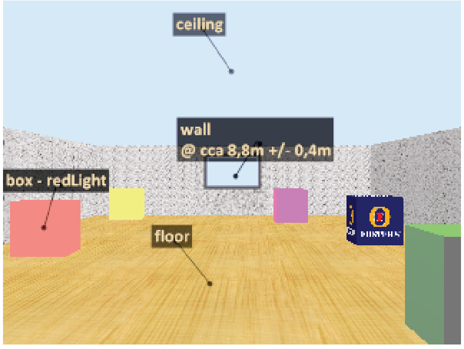


Figure 2: The same view as in case of Fig. 1 after the interpretation phase. Some of the detected objects are labeled. The opposing wall is labeled also with its approximate distance with respect to the robot.

6 CONCLUSIONS

State of the art techniques have been discussed in this paper. In spite of a great advance in past years, a generally usable SLAM solution is still missing. We identify the pertinence of semantic SLAM for the future of mobile robotics and we present our initial research on this field inspired by the human way of navigation and place description. We show a concept of a prospective semantic SLAM algorithm driven by object recognition and the use of human spatial concepts.

For description of a scene by semantic means, a fast and efficient algorithm for image segmentation is an important starting point. A part of our future work will be dedicated to further development of such an algorithm. Another part of our future work will be focused on development of algorithms of semantic SLAM we outlined in this paper. They will be consequently implanted and verified in an indoor environment on the real Nao robot.

REFERENCES

Durrant-Whyte, H., et al. Simultaneous Localisation and Mapping (SLAM): Part I The Essential Algorithms. Robotics and Automation Magazine. 2006a, Vols. 13, No 2, pp. 99-110.

Simultaneous Localisation and Mapping (SLAM): Part II State of the Art. Robotics and Automation Magazine. 2006b, Vols. 13, No 3, pp. 108-117.

Ekvall, S., Jensfelt, P. and Kragic, D. Integrating Active Mobile Robot Object Recognition and SLAM in Natural Environments. International Conference on Intelligent Robots and Systems, 2006 IEEE/RSJ. Beijing : IEEE, 2006, pp. 5792-5797.

Galindo, C., et al. Multi-Hierarchical Semantic Maps for Mobile Robotics. International Conference on Intelligent Robots and Systems (IROS 2005). Edmonton : IEEE, 2005, pp. 2278- 2283.

Hahnel, D., et al. Map Building with Mobile Robots in Dynamic Environments. Proceedings of the IEEE International Conference on Robotics and Automation. Taipei : IEEE, 2003, Vol. 2, pp. 1557-1563.

Lucchese, L. and Mitra, S. K. Color image segmentation: A state-of-the-art survey. Proc. Indian Nat. Sci. Acad. (INSA-A). 2001, Vols. 67-A, pp. 207-221.

Meger, D., et al. Curious George: An attentive semantic robot. Robotics and Autonomous Systems. Amsterdam : North-Holland Publishing Co., 2008, Vol. 56, pp. 503-511.

Muhammad, N., Fofi, D. and Ainouz, S. Current state of the art of vision based SLAM. Image Processing: Machine Vision Applications II, Proceedings of the SPIE. 2009, Vol. 7251, pp. 72510F-72510F.

Nüchter, A. and Hertzberg, J. Towards semantic maps for mobile robots. Robotics and Autonomous Systems. Amsterdam : North-Holland Publishing Co., 2008, Vol. 56, pp. 915-926.

Persson, M., et al. Probabilistic Semantic Mapping with a Virtual Sensor for Building/Nature detection. International Symposium on Computational Intelligence in Robotics and Automation. Jacksonville : IEEE, 2007, pp. 236-242.

Thrun, S. and Leonard, J. J. Simultaneous Localization and Mapping. [ed.] B. Siciliano and O. Khatib. Springer Handbook of Robotics. Berlin Heidelberg : Springer-Verlag, 2008, 37.

Vasudevan, S., et al. Cognitive maps for mobile robots-an object based approach. Robotics and Autonomous Systems. Amsterdam : North-Holland Publishing Co., 2007, Vol. 55, pp. 359-371.

HOMOTHETIC APPROXIMATIONS FOR STOCHASTIC PN

Dimitri Lefebvre
GREAH – University Le Havre, France
dimitri.lefebvre@univ-lehavre.fr

Keywords: Stochastic Petri Nets, Continuous Petri Nets, Fluidification, Steady State, Reliability Analysis.

Abstract: Reliability analysis is often based on stochastic discrete event models like stochastic Petri nets. For complex dynamical systems with numerous components, analytical expressions of the steady state are tedious to work out because of the combinatory explosion with discrete models. For this reason, fluidification is an interesting alternative to estimate the asymptotic behaviour of stochastic processes with continuous Petri nets. Unfortunately, the asymptotic mean marking of stochastic and continuous Petri nets are mainly often different. This paper proposes a geometric approach that leads to a homothetic approximation of the stochastic steady state in specific regions of the marking space.

1 INTRODUCTION

Reliability analysis is a major challenge to improve the safety of industrial processes. For complex dynamical systems with numerous interdependent components, such studies are mainly based on stochastic discrete event models like Markov models (Rausand et al., 2004) or stochastic Petri nets (SPNs) (Molloy, 1982). Such models are mathematically well founded and lead either to analytical results or numerical simulations. But in case of large systems, the combinatory explosion limits their use. In this context, fluidification can be discussed as a relaxation method.

This paper is about the approximation of the SPNs asymptotic mean markings and average throughputs by mean of continuous Petri nets (CPNs) under infinite server semantic (Vazquez et al., 2008; Lefebvre et al., 2009). The limits of the fluidification of SPNs are discussed according to the partition in regions of the reachability state space. A characterization of the regions is proposed that leads to a homothetic approximation of the stochastic steady state. The proposed results are not constructive but concern the existence of solutions. They may be helpful to investigate the properties of a considered SPN and they may lead for example to the design of observers or controllers for stochastic processes.

2 FLUIDIFICATION OF SPN

2.1 Stochastic Petri Nets

A Petri net (PN) is defined as $<\boldsymbol{P}, \boldsymbol{T}, W_{PR}, W_{PO}>$ where $\boldsymbol{P} = \{P_i\}$ is a set of n places and $\boldsymbol{T} = \{T_j\}$ is a set of q transitions, $W = W_{PO} - W_{PR} \in (\mathbf{Z})^{n\times q}$ is the incidence matrix, $M(t)$ is the PN marking vector and M_I the PN initial marking (David *et al.,* 1992). Depending on the incidence matrix, PNs may have P-semiflows. A P-semiflows $y \in (\mathbf{Z}^+)^n$ is a non-zero solution of equation $y^T.W = 0$. Let define $Y = \{y_1,..., y_h\}$ as a basis of W^T kernel, composed of h minimal P-semiflows. For simplicity, the basis Y will be represented as a matrix $Y \in (\mathbf{Z}^+)^{n \times h}$ that satisfies (1):

$$Y^T.M(t) = Y^T.M_I = C, \quad t \geq 0 \qquad (1)$$

Let define, for each minimal P-semiflow $y_i \in Y$, its support as the subset $\boldsymbol{P}(y_i) \subset \boldsymbol{P}$ of places that belong to the corresponding marking invariant. For each $\boldsymbol{P}(y_i)$ it is possible to select a single place and to recover the marking of this place from the marking of the other places in $\boldsymbol{P}(y_i)$. Let define as a consequence the subset $\boldsymbol{P_2} \subset \boldsymbol{P}$ of h places whose markings may be recovered from Y, and the subset $\boldsymbol{P_1} \subset \boldsymbol{P}$ of n - h other places. The permutation matrix D defined according to $\boldsymbol{P_1}$ and $\boldsymbol{P_2}$ leads to the marking $M' = ((M'_1)^T\ (M'_2)^T)^T = D.M$.

A stochastic Petri net (SPN) is a timed PN whose transitions firing periods are characterized a firing rate vector $\mu = (\mu_j) \in (\mathbf{R}^+)^q$ (Molloy, 1982). The marking and mean marking vectors of a SPN at time t will be referred as $M_s(t)$ and $MM_s(t)$. The SPNs considered in this paper are bounded, reinitialisable, with infinite server semantic, race policy and resampling memory. As a consequence, the considered SPNs have a reachability graph with a finite number N of states and their marking process is mapped into a Markov model with state space isomorphic to the reachability graph (Bobbio *et al.,* 1998). The Markov model has an asymptotic state propability vector $\Pi_{ss} = (\pi_{ss\ k}) \in [0, 1]^{1 \text{ x } N}$ and the asymptotic mean marking M_{mms} of SPNs depends from Π_{ss}:

$$m_{mms\ i} = \sum_{k=1,\ldots,N} m_{k\ i}.\pi_{ss\ k}, i = 1,..,n \tag{2}$$

2.2 Continuous Petri Nets and Regions

CPNs have been developed in order to provide continuous approximations of the discrete behaviours of PNs (David *et al.,* 1992; Silva and Recalde*,* 2004). A CPN is defined as < *PN*, X_{max} > where *PN* is a Petri nets and $X_{max} = \text{diag}(x_{max\ j}) \in (\boldsymbol{R}^+)^{qxq}$ is the diagonal matrix of maximal firing speeds $x_{max\ j}$, $j = 1,\ldots q$. $M_c(t)$ is the marking vector and $X_c(t) = (x_{cj}(t)) \in (\boldsymbol{R}^+)^q$ is the firing speeds vector that satisfy $dM_c(t) / dt = W.X_c(t)$. For CPNs with infinite server semantic, $X_c(t)$ depends continuously on the marking of the places according to $x_{cj}(t) = x_{max\ j}.$ min $(m_k(t) / w^{PR}_{kj})$, for all $P_k \in {}^\circ T_j$, where ${}^\circ T_j$ stands for the set of T_j upstream places. A marked CPN has a steady state if the marking vector $M_c(t)$ tends to a finite limit M_{mmc} in long run.

According to the function "min(.)", the marking space of CPNs is divided into K regions $\mathbf{A_k}$ (eventually empty) with $K = |{}^\circ T_1| \text{ x} \ldots \text{ x } |{}^\circ T_q|$. Each region $\mathbf{A_k}$ is defined by its PT-set (Julvez *et al.* 2005) defined according to (3):

$$\text{PT-set}(\mathbf{A_k}) = \{(P_i, T_j) \text{ s.t. } \forall\ M_c(t) \in \mathbf{A_k}, x_{cj}(t) = x_{max\ j}(t).m_{ci}(t)/w^{PR}_{ij}\} \tag{3}$$

The place P_i such that i = argmin $(m_k(t) / w^{PR}_{kj})$ for all $P_k \in {}^\circ T_j$ is the critical place for transition T_j at time t. A constraint matrix $A_k = (a^k_{ij}) \in (\boldsymbol{R}^+)^{q \text{ x } n}$, $k = 1,\ldots,K$, $i = 1,\ldots, q$ and $j = 1,\ldots, n$ is defined for each region $\mathbf{A_k}$ according to the corresponding PT-set: $a^k_{ij} = 1/w^{PR}_{ji}$ if $(P_i, T_j) \in$ PT-set$(\mathbf{A_k})$ and $a^k_{ij} = 0$ elsewhere. For each region $\mathbf{A_k}$, equation (4) holds:

$$\forall M_c(t) \in \mathbf{A_k},\ dM_c(t) / dt = W.\ X_{max}.A_k.M_c(t) \tag{4}$$

Definition: A region $\mathbf{A_k}$ is critical if there exists two transitions T_j and T_k that have the same critical place P_i in region $\mathbf{A_k}$.

Proposition 1: Marking $M_c \in \mathbf{A_k}$ iff M_c satisfies (5):

$$\begin{pmatrix} -I_{\text{n}} \\ A(k) \\ Y^T \\ -Y^T \end{pmatrix}.M_c \leq \begin{pmatrix} 0 \\ 0 \\ \text{C} \\ -\text{C} \end{pmatrix} \tag{5}$$

with In and the identity matrix of size n and:

$$A(k) = (\quad \ldots \quad) \in (Z^+)^{q.(K-1)\text{x}\, n} \tag{6}$$

Proof: Equation (5) results from the definition of PT-sets and P-semiflows of the PN. The equation $-I_n.M_c \leq 0$ stands for the positivity of the marking. The equation $A(k).M_c = 0$ defines the region borders according to the "min" functions. Finally, the equation $Y^T.M_C \leq C$ and $-Y^T.M_C \leq -C$ result from the P-semiflows.□

2.3 Continuous Approximation of SPNs

Numerous structural and behavioural properties are not preserved with fluidification (Silva and Recalde 2004). The average throughput and mean marking of a CPN are mainly not identical to the ones of a discrete PN (Julvez *et al.,* 2005, Lefebvre et al., 2009). Concerning SPNs, the steady state is mainly often different from the one of a CPN with same parameters ($x_{maxj} = \mu_j$ $j = 1,\ldots,q$). The asymptotic mean markings of SPNs can be approximated with the steady state of CPNs if all transitions remain enabled with degree at least 1 in long run and the marking vector does not leave the region of initial marking in long run (Vasquez et al, 2008). These conditions limit strongly the interest of fluidification. In our preceding works, we have investigated the limit of fluidification for the approximation of SPNs. CPNs with a modified set of maximal firing speed can be used to approximate the mean marking in non critical regions (Lefebvre and Leclercq 2010). In the next section, we continue this investigation for critical regions.

3 HOMOTHETIC ESTIMATION

Let consider the problem to reach Mmms when Mmms ∈ Ai (eventually critical) and MI ∈ Ak (non critical) with Ai ≠ Ak. The proposition 2 provides conditions to work out admissible but partial homothetic transformations of ratio α such that (α.(M’mms1)T $(M''_{mms2})^T)^T$ ∈ $\mathbf{A_k}$. Then a CPN with modified constant maximal firing speeds ($x_{maxj} \neq \mu_j$ $j = 1,\ldots,q$) is worked out with proposition 3. This CPN approximates $\alpha.M'_{mms1}$.

Proposition 2: Let define $M'_{mms} = D.M_{mms}$ and M''_{mms2} such that the $Y^T.(\alpha.(M'_{mms1})^T\ (M''_{mms2})^T)^T = C$. The condition $(\alpha.(M'_{mms1})^T\ (M''_{mms2})^T)^T \in \mathbf{A_k}$ holds if α satisfies (7):

$$\begin{pmatrix} -I_{n-h} \\ A(k) \\ Y^T \\ -Y^T \end{pmatrix}.D^{-1}.\begin{pmatrix} \alpha.M'_{mms1} \\ M''_{mms2} \end{pmatrix} \leq \begin{pmatrix} 0 \\ 0 \\ C \\ -C \end{pmatrix} \qquad (7)$$

***Proof*:** Proposition 2 results from proposition 1 by replacing M_c by $D^{-1}.D.M_c$ and by considering the partial homothetic transformations of ratio α. Proposition 2 characterises the intersection of the region $\boldsymbol{A}_{\mathbf{k}}$ and the direction M_{mms}.

***Proposition 3*:** Consider a SPN with $M_I \in \boldsymbol{A}_{\mathbf{k}}$ (non critical) and $M_{mms} \in \boldsymbol{A}_{\mathbf{i}}$ (eventually critical) with $\mathbf{A_i} \neq \mathbf{A_k}$. Let define the CPN with same structure and initial marking. $M_c(t)$ tends asymptotically to M_{mmc} such that $M'_{mmc1} = \alpha.M'_{mms1}$ if there exist X_{max} such that $M_c(t)$ satisfies the proposition 1 for all $t \geq 0$ and equation (8) holds:

$$W.X_{max}.A_k.D^{-1}.\begin{pmatrix} \alpha.M'_{mms1} \\ M''_{mms2} \end{pmatrix} = 0 \qquad (8)$$

***Proof*:** Proposition 3 results from the steady state solution of equation (4) and from the partial homothetic transformations of ratio α..

The propositions 2 and 3 lead to a 4-stages algorithm for estimating Mmms in critical regions.

Work out the transformation matrix D.

List the conditions to be satisfied by α, so that the partial homothetic transformation of Mmms and MI are in the same non critical region.

Work out the modified constant firing speeds that drive Mc(t) to Mmmc st M’mmc1 = α.M’mms1.

Recover the asymptotic stochastic mean marking Mmms with (1).

4 EXAMPLE

Consider for example the marked SPN described in fig.1 (Julvez *et al.* 2005). This PN has 2 P-semiflows: $Y = ((0\ 0\ 0\ 1\ 1)^T\ (1\ 1\ 2\ 1\ 0)^T)^T$ and $C = (4\ 5)^T$. The subsets of places $\boldsymbol{P_1} = \{P_1, P_2, P_3\}$ and $\boldsymbol{P_2} = \{P_4$, P5} are defined according to Y and lead to the trivial transformation matrix D = I5 (i.e. M’ = M).

The fig. 2 illustrates stochastic mean markings that are reached from MI = (5 0 0 0 4)T according to various transitions firing rate vectors μ ∈ [0 : 10]4.

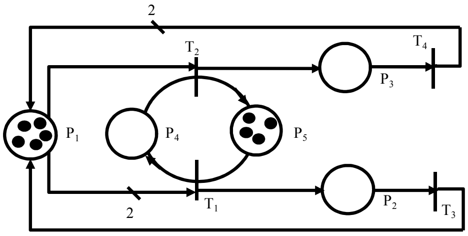


Figure 1: An example of SPN with $M_I = (5\ 0\ 0\ 0\ 4)^T$.

If the PN of fig. 1 is considered as a CPN, 4 regions $\mathbf{A_1}$ to $\mathbf{A_4}$ exist. The regions are defined by the constraint matrices A_1 to A_4.

$$A_1 = \begin{pmatrix} 1/2 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} A_2 = \begin{pmatrix} 1/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} A_3 = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} A_4 = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

The regions are also depicted in figures 2 to 4 according to the full lines (reachable area limits) and dotted lines (regions intersections).

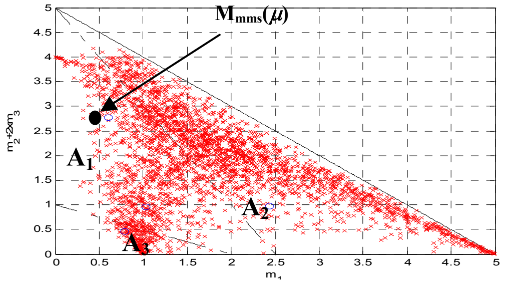


Figure 2: Projection in plan (m_1, $m_2 + 2.m_3$) of M_{mms} for the SPN of fig. 1 with various vectors $\mu \in [0 : 10]^4$.

The CPN as a single critical region $\mathbf{A_1}$ and $M_I \in \mathbf{A_2}$. The fig. 3 illustrates the asymptotic continuous mean markings that are reached from M_I and according to various maximal firing speeds $x_{max\ j} \in [0.1 : 10]$, $j = 1,\ldots,4$. Some areas in the critical region $\mathbf{A_1}$ are reachable with SPNs and not with CPNs. For examples, the asymptotic mean markings $M_{mms}(\mu) = (0.6\ 0.2\ 1.3\ 1.6\ 2.4)^T$ obtained with $\mu = (6\ 2\ 3\ 0.5)^T$ is in critical region $\mathbf{A_1}$ and is not reachable with CPN (figs. 2 and 3): $X_{max} = \text{diag}(6\ 2\ 3\ 0.5)^T$ leads to $M_{mmc}(X_{max}) = (0.1\ 0.1\ 0.4\ 3.9\ 0.1)^T$ and any other maximal firing speeds also fail.

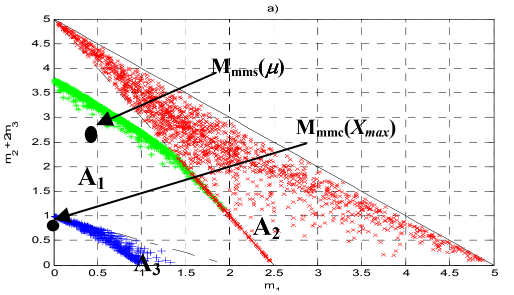


Figure 3: Projection in plan (m_1, $m_2+2.m_3$) of M_{mmc} for the CPN of fig. 1 with various $x_{max\,j} \in [0.1 : 10]$, j=1,…,4.

The propositions 2 and 3 are used to work out the admissible ratio α and the maximal firing speeds that lead to homothetic approximations of $M_{mms1}(SPN)$. In the region $\mathbf{A_2}$, (7) leads to (10):

$$\begin{pmatrix} & & -I_5 & & \\ -1 & 0 & 0 & 1 & 0 \\ 1/2 & 0 & 0 & 0 & -1 \\ & & Y^T & & \\ & & -Y^T & & \end{pmatrix} . \begin{pmatrix} \alpha.m'_{mms1} \\ \alpha.m'_{mms2} \\ \alpha.m'_{mms3} \\ m''_{mms4} \\ m''_{mms5} \end{pmatrix} \leq \begin{pmatrix} 0_{5x1} \\ 0 \\ 0 \\ 4 \\ 5 \\ -4 \\ -5 \end{pmatrix} \tag{9}$$

and then to the admissible interval $\alpha \in [5/(2.m_{mms1}+m_{mms2}+2.m_{mms3}) : 5/(m_{mms1}+m_{mms2}+2.m_{mms3})]$. For the considered example $\alpha \in [1.26: 1.48]$. The figure 4 illustrates various homothetic marking trajectories for SPN obtained for some values of parameter α in admissible interval in order to reach $\alpha.M'_{mms1}(\mu)$.

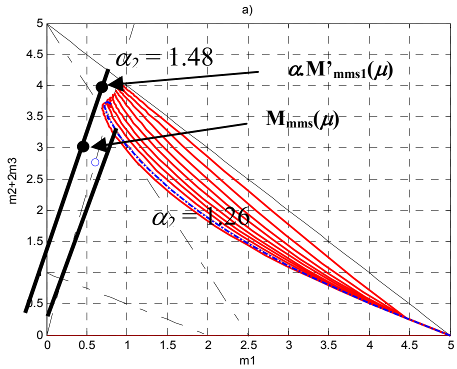


Figure 4: Projection in plan (m_1, $m_2+2.m_3$) of the homothetic convergence to $\alpha.M'_{mms}(\mu)$.

The proposition 3 is used to work out the set of the admissible maximal firing speeds that depend on the parameter α such that the CPN with same structure and initial marking tends to M_{mms}:

$$\begin{aligned} x_{max1} &= 2.x_{max4}.m_{mms3} / m_{mms1} \\ x_{max2} &= \alpha.x_{max4}.\ m_{mms3} / (5-\alpha.(m_{mms1}+m_{mms2}+2.m_{mms3})) \\ x_{max3} &= x_{max4}\ m_{mms3} / m_{mms2} \end{aligned} \tag{10}$$

where x_{max4} is a dof. For example, consider the particular homothetic ratio α = 4/3. The trajectory (dotted line in figure 4), obtained for X_{max} =(4.25, 3.41, 6, 1) results in asymptotic marking m'_{mmc1} = 0.80, m'_{mmc2} = 0.28, m'_{mmc3} = 1.71. From this approximation, it is easy to recover the asymptotic stochastic mean marking $M_{mms}(\mu)$.

5 CONCLUSIONS

This paper has proposed partial homothetic transformations of the SPN mean marking to approximate. The proposed results concern the existence of solutions but are not constructive in the sense that the asymptotic stochastic mean markings to estimate by CPNs must a priori be known. The selection of the best projectors and ratios will be investigated in our further works. Our future work is also to investigate continuous approximations directly derived from the SPNs transition firing rates.

REFERENCES

Bobbio A., Puliafito A., Telek M., Trivedi K. (1998) Recent Developments in Stochastic Petri Nets, *J. of Cir., Syst., and Comp.*, Vol. 8, No. 1, pp. 119--158.

David R., Alla H., (1992) *Petri nets and grafcet – tools for modelling DES*, Prentice Hall, London.

Júlvez G., Recalde L. Silva M. (2005) Steady-state performance evaluation of continuous mono-T-semiflow Petri nets, *Automatica*, 41 (4), pp. 605-616.

Lefebvre D., Leclercq E., Khalij L., Souza de Cursi E., El Akchioui N. (2009), Approximation of MTS stochastic Petri nets steady state by means of continuous Petri nets: a numerical approach, Proc. IFAC ADHS, pp. 62-67, Zaragoza, Spain.

Lefebvre D., Leclercq E., (2010), Approximation of Stochastic Petri Nets steady state by CPNs with constant and piecewise-constant maximal firing speeds, MOSIM 2010, Hammamet, Tunisia.

Molloy M.K., (1982) Performance analysis using stochastic Petri nets, *IEEE Tran. on Computers C*, vol. 31, pp. 913 – 917.

Rausand M. and Hoyland A. (2004), *System reliability theory: models, statistical methods, and applications*, Wiley, Hoboken, New Jersey.

Silva M. and Recalde L. (2004) On fluidification of Petri Nets: from discrete to hybrid and continuous models, *An. Reviews in Control*, Vol. 28, no. 2, pp. 253-266.

Vazquez R., Recalde L., Silva M., (2008) Stochastic continuous-state approximation of markovian Petri net systems, Proc. IEEE – CDC08, pp. 901 – 906, Cancun, Mexico.

FDI WITH NEURAL AND NEUROFUZZY APPROACHES
Application to Damadics

Y. Kourd, N. Guersi
Department of Control Engineering, Faculty of Science and Engineer, Mohamed Khider Biskra University, Algeria
Department d'Electronique, Université Badji Mokhtar Annaba, Algeria
{ykourd, Guerssi54}@yahoo.fr

D. Lefebvre
GREAH – Université Le Havre, 25 rue Philippe Lebon, 76058 Le Havre, France
dimitri.lefebvre@univ-lehavre.fr

Keywords: Fault Diagnosis, Modelling, Residual Generation, Residual Evaluation, Neural Classifier, Neurofuzzy Classifiers.

Abstract: Fault diagnosis is a major challenge for complex systems as long as it increases the safety and productivity. This work concerns faults diagnosis, based on artificial intelligence, neural networks, and fuzzy logic. Thanks to an associative memory, neural networks have good capacities of organization, approximation and classification. Combined with fuzzy logic, neural networks are an effective tool for system modelling, fault detection and fault diagnosis. This paper illustrates the potential of these tools for the modelling and the diagnosis of an industrial actuator (DAMADICS benchmark).

1 INTRODUCTION

Fault detection and isolation (FDI) is a major issue for complex systems as long as it increases the safety and productivity of these systems. Its first vocation is the detection and the isolation of system failures. The necessity to detect and isolate early the failures calls upon techniques of the artificial intelligence. These techniques have been recently developed and improved by many researchers. The point is that artificial intelligence makes easier the task carried out by the operators as long as the observation of symptoms and the data analysis or information interpretation is carried out by the diagnosis system.

Several methods exist for the diagnosis of dynamical systems. Basically, model-based and data-based methods can be distinguished (Chow, 1980; Patton et al. 1989; Gertler, 1991; Willsky, 1976). Model – based methods compare the measured data with the knowledge provided by the model of theconsidered system in order to detect and isolate the faults that disturb the process. Such techniques require a sufficiently accurate mathematical model of the process.Data-based methods require a lot of process measurements and can be divided into signal processing methods and artificial intelligence approaches. Model and data based methods are used to design residual signals. The fault detection results from the comparison of the residuals with arbitrary thresholds: a fault is detected each time one residual ccross over the threshold. This comparison is calculated on line. To isolate the faults, residuals are structured to be robust and sensitive to some specific sets of faults.

In this context, our study concerns the investigation of model-based FDI methods with artificial intelligence, particularly neural networks and fuzzy logic. Fuzzy logic can be used to describe the system behaviours according to linguistic rules and fuzzy sets. The advantage of fuzzy logic is that it can be used in presence of uncertainties. The drawback is that the number and expression of the rules and also the parameters of the membership functions that define the sets are not easy to be work out. In that case, neural networks are helpful to identify the unknown parameters according to measured data and to learning algorithms.

This paper concerns the application of neural networks, fuzzy logic and neurofuzzy systems (ANFIS) for an industrial actuator from the sugar factory in Lublin, Polen (Damadics, 2004).

2 FDI METHODS

The proposed approach can be presented with 3 stages (1) the design of a data – based model; (2) the fault detection according to a residual generator; (3) the fault isolation thanks to neural or neurofuzzy classifiers.

2.1 Reference Model Design

In the following we consider dynamic systems with q inputs $u_i(t)$ and n outputs $y_j(t)$ and it is assumed that the state variables are no measurable. Such systems exhibit often complex dynamics, with strong nonlinearities. As a consequence, knowledge –based models are not easy to obtain. Another approach lies in the data–based models. Artificial neural networks (ANN) are often used for that purpose (Juditsky et al. 1995). The goal is to design a model that will be used for the generation of residuals (figure 1).

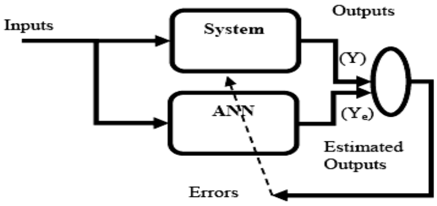


Figure 1: Data-based model design.

In order to get the best ANN architecture, several configurations are tested according to a trial – error processing that uses pruning to eliminate the useless nodes. The learning of the ANN is obtained according to the Levenberg-Marquardt algorithm with early stopping. This algorithm is known for its rapid convergence. During learning stage, the ANN is trained with data collected during the normal functioning of the system. Then the ANN reference model is validated with another set of data.

2.2 Fault Detection

The considered system may be affected by p faults Fi with the assumption that simultaneous faults do not occur. The vector r(t) of n residuals $r_i(t)$ is calculated according to the difference between the outputs vector of the system y(t) and the output vector of the ANN model $y_e(t)$. As long as the system has no fault, the estimated output $y_e(t)$ remains in the neighbourhood of the actual output y(t) and the residual r(t) is near zero. When a fault occurs, at least one estimated output becomes different from and the actual one and the corresponding residual is no longer near zero.

2.3 Fault Isolation with ANN

A neural classifier has been developed to isolate the faults after detection (Kourd et al., 2008). This classifier is a multilayer Perceptron ANN (figure 2). The inputs are the n residuals $r_i(t)$ and the outputs are the p signatures f_i of the faults F_i that are under consideration.

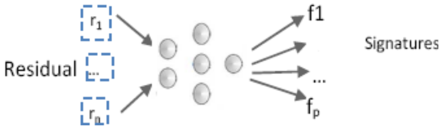


Figure 2: Neural classifier.

The neural classifier is trained and validated with a learning algorithm similar to the one used for reference model design.

2.4 Fault Isolation with ANFIS

In order to deal with improve the isolation, a neurofuzzy classifier has also been developed (figure 3). Such a classifier has an hybrid architecture that takes advantages from fuzzy logic and neural networks (Nauck et al. 1995). This classifier is design as a double Takagi- Sugeno ANFIS networks. The inputs are the n residuals $r_i(t)$ and the outputs are the p signatures f_i of the faults F_i that are under consideration.

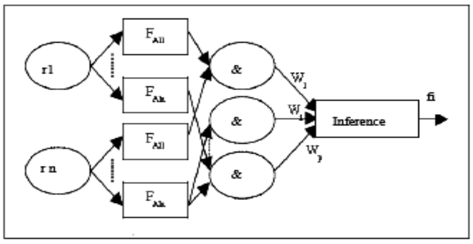


Figure 3: Double neurofuzzy ANFIS classifiers.

3 APPLICATION TO DAMADICS

3.1 System Description

The system under consideration is the DAMADICS valve (figure 4). It is composed of a pneumatic servomotor and a controller that drives the valve.

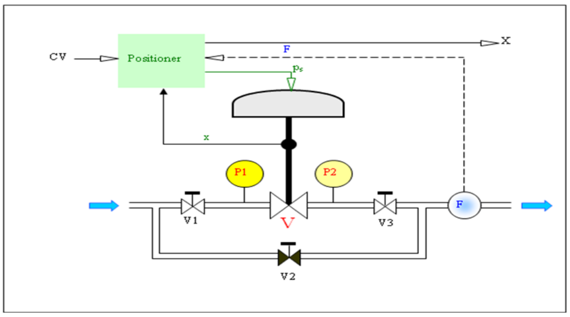


Figure 4: Actuator shema.

This system has four input variables (CV, P1, P2, T1) and 2 outputs variables (X, F) that are described in table 1 (DAMADICS 2004). The other variables are not considered in our application.

Table 1: Input and output variables.

Input	Range	Unit	Description
CV	[0,1]	-	control signal from external PI controller
P1	[2000, 4e+6]	Pa	Inlet liquid pressure
P2	[2000, 4e+6]	Pa	Outlet liquid pressure
T1	[30, 110]	C °	Liquid temperature
Output	**Range**	**Unit**	**Description**
X	[0,1]	-	Position of the rod
F	[0,1]	-	Average flow

There exist 20 possible faults that may affect the functioning of the actuator (DAMADICS 2004). Some faults may be abrupt or incipient ones.

3.2 Model Design

The actuator is modeled with two multilayer perceptrons ANN that represent the interaction between the inputs and the outputs according to (1):

$$\begin{aligned} netX &= netX\ (CV, P1, P2, T1) \\ netF &= netF(X, P1, P2, T1) \end{aligned} \tag{1}$$

To select the structure of the neural networks netX and netF, numerous tests have been carried out to obtain the best architectures (i.e. number of hidden layers and number of neurons by layer) in order to model the operation of the actuator. The training and test data were generated by the simulation of the Matlab-Simulink actuator model (Kourd et al. 2008).

From table 2, we notice that netX(6,3,1) and netF(6,3,1) give the best results. When the training is over, the ANN netF provides estimated outputs that are not far from the actual ones. Validation is done with the measured data provided by the 'Lublin Sugar Factory in 2001 (DAMADICS, 2004). Validation is illustrated on figure 5.

Table 2: netX and netF neural networks structure.

netX	Layer 1	Layer 2	Output layer	MSE
Operation 1	6	3	1	0.00033
Operation 2	10	8	1	0.00149
Operation 3	21	12	1	0.00491
netF	Layer 1	Layer 2	Output layer	MSE
Operation 1	6	3	1	0.000199
Operation 2	10	8	1	0.000849
Operation 3	21	12	1	0.00949

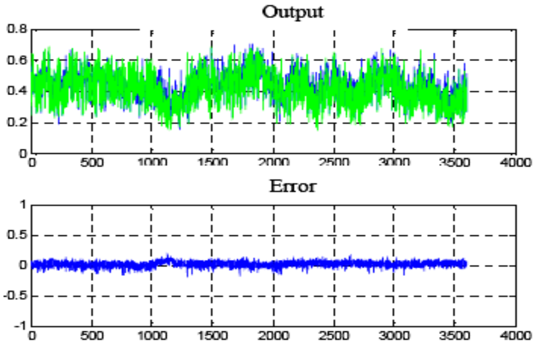


Figure 5: Actual output F and estimated output netF (up); Instantaneous error (down).

The modelling error is acceptable. Similar conclusions are obtained with ANN netX. Let notice that the output of netX is an input for netF and the sensitivity of the estimation depends strongly on the error on netX. As a conclusion, both ANNs provide a good approximation of the actuator dynamics.

3.3 Fault Detection

In the following four faults will be considered: F7 (medium cavity or critical flow) F10 (servo-motor's diaphragm perforation) F15 (positioner spring fault) and F17 (positioner supply pressure drop) in order to illustrate the efficiency of the proposed approach. Two residual are designed according to (2):

$$\begin{aligned} rF &= F - netF \\ rX &= X - netX \end{aligned} \tag{2}$$

During normal functioning the residuals remain near zero: their magnitude is in range [-0,2; 0,2]. The value 0.2 will be used as detection threshold (Emami-Naeini, 1988; Ding and Frank, 1991). Let us notice that a low pass filter is used to remove high frequency noises. In figure 6, the residuals are worked out when the fault F7 is simulated during interval [500 1000] time units (times units are in seconds).

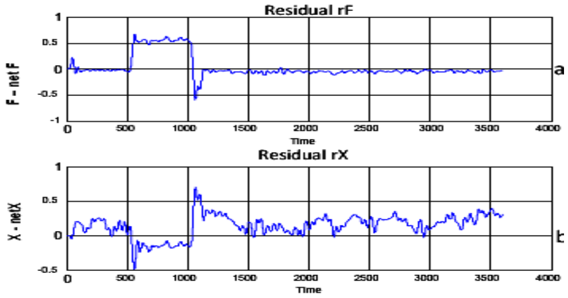


Figure 6: Residual in presence of fault F7; Residual rF (up); Residual rX (down).

3.4 Fault Isolation

The classifiers presented in section 2 are trained with a set of simulated faults. Then they are validated according to the real data collected on the Sugar factory:

- During period [500, 1100], the fault F7 occurs.
- During period [4100, 4600], the fault F10 occurs.
- During period [7700, 9000], the fault F15 occurs.
- At times 11300 and 11850 the fault F17 occurs.

The ANN classifier presented in section 2.3 receives two inputs: the residuals rX and rF and delivers four outputs that are the signatures f7, f10, f15, f17 of the faults F7, F10, F15, F17. The signature f7 is given in figure 7. The neural classifier gives acceptable results in the sense that the signature of each fault is far from zero when the considered faults occur. But misclassifications may occur.

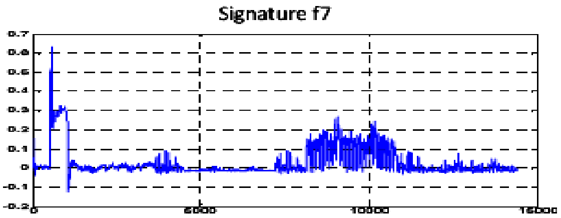


Figure 7: Magnitude of the fault signatures f7 in function of time for ANN classifier.

The ANFIS classifier presented in section 2.4 has also two inputs and four outputs. The resulting signature f7 is given in figure 8.

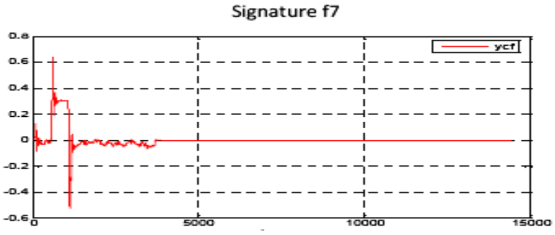


Figure 8: Magnitude of the fault signature f7 in function of time for ANFIS classifier.

The use of ANFIS classifier improves the classification results. The number of misclassifications decreases and the quadratic mean square on error of the residuals decreases (table 3).

Table 3: Quadratic mean square error.

	F7	F10	F15	F17
Neural classifier (x 10^{-3})	3.8	10.0	43.9	23.2
Fuzzy Neural classifier (x 10^{-3})	2.5	4.6	30.6	3.3

4 CONCLUSIONS

This paper uses neural networks and fuzzy logic for the fault diagnosis. The neural networks are good tools for the modelling and diagnosis of non linear processes, but some problems remain in the selection of the optimal architecture as well as the number of layers, and the numbers of neurons in each layer. The uses of neurofuzzy networks improve the classification of faults.

In our further works we will consider numerical criteria to compare both classifiers, we will also compare our results with the existing results and we will improve the neurofuzzy diagnosis system.

REFERENCES

Chow, E. Y., Failure detection system design methodology, Thesis, Lab. Information and Decision system, M.I.T, Cambridge, 1980.

Ding, X., Frank P. M., Frequency domain approach and threshold selector for robust model-base fault detection and isolation, *Proc. SAFEPROCESS'91,* Baden-Baden, Germany, vol. 1, pp. 307-312, 1991.

Emami-Naeini, A. E. A., Effect of model uncertainty of failure detection: the threshold selector, *IEEE-TAC*, 33, pp. 1106- 1115, 1988.

Gertler, J. J., Analytical redundancy methods in fault detection and isolation – survey and synthesis, *Proc.*

SAFEPROCESS'91, Baden-Baden, Germany, pp. 9-21, 1991.

Juditsky, A., Hjalmärsson, H., Benveniste, A., Delyon, B., Ljung, L., Sjöberg, J., Zhang, Q., Nonlinear black-box modelling in system identification: mathematical foundations. *Automatica*, 31, pp. 1725-1750, 1995.

Nauck, D., Kruse, R., Nefclass – A neurofuzzy approach for the classification of data, Proc. Symp. on Applied Computing – ACM, Nashville, USA, 1995.

Patton, R. J., Frank, P. M., Clarck, R. N., Fault diagnosis in dynamic systems, Prentice Hall, 1989.

Willsky, A. S., A survey of design methods for failure detection in dynamic systems, Automatica 12, pp. 601-611, 1976.

Kourd, Y., Guersi, N., Lefebvre, D., A two stages diagnosis method with neuronal networks, Proc. ICEETD 2008, Hammamet, Tunisie.

DAMADICS (2004): Website of DAMADICS: Development and Application of Methods for Actuator Diagnosis in Industrial Control Systems. http://diag.mchtr.pw.edu.pl/damadics/.

CONSENSUS PROBLEM OF MULTI-AGENT SYSTEMS WITH MARKOVIAN COMMUNICATION FAILURE

Yuebing Hu, James Lam
Department of Mechanical Engineering, University of Hong Kong, China
yuebing@hkusua.hku.hk, james.lam@hku.hk

Jinling Liang
Department of Mathematics, Southeast University, Nanjing 210096, China
jinlliang@seu.edu.cn

Keywords: Consensus control, Markov chain, Mean-square stability, Communication failure, Multi-agent systems.

Abstract: This paper studies the consensus problem of multi-agent systems with Markovian communication failure which may be caused by limited communication capacity. The occurrence of the failures is modeled by a discrete-time Markov chain. A consensus sufficiency condition is established in terms of linear matrix inequalities (LMIs). Based on this condition, a new controller design method is provided. A numerical example is utilized to illustrate the effectiveness of the proposed approach.

1 INTRODUCTION

In the past few years, multi-agent system (MAS) has sparked the interest of researchers. Up to now, results of networked MASs have been broadly applied to biology, physics and engineering, such as the study of swarming behavior (Liu and Passino, 2004), automated highway systems (AHSs) (Bender, 1991) and congestion control in communication (Paganini et al., 2005). In the cooperative behaviors, consensus, which means making a group of agents to reach an agreement on certain quantity of interest that depends on the states of all agents, is a fundamental topic in MASs fields and has been studied recently (Olfati-Saber et al., 2007).

Due to the special property of MASs, the interconnected communication network among agents plays an important role in the consensus reaching problem and is usually described by Laplacian graph. Depending on applications, the network topologies of multiple agents are either fixed or switched, while the latter is more practical due to the limited or imperfect communication channel, noises or some special objectives. Results on switching topology have been provided in recent articles such as (Olfati-Saber and Murray, 2004).The physical systems are usually of big complexity, thus some dynamic processes are described by time-varying linear model. Particularly, for systems subject to randomly changing parameters, Markov jump linear system (MJLS), which is a hybrid system composed of a finite number of subsystem modes, is an appropriate class of models and has been extensively studied (Xiong and Lam, 2007). However, to the best of our knowledge, although MASs are usually treated as networked systems, the issue of Markovian topology switching processes has not been fully investigated and fruitful results of MJLS were not applied to MASs until now.

In this paper, we investigate the consensus control problem of MASs with communication failure. By modeling the communication process in a Markovian process, a new sufficiency condition of the consensus problem is established in terms of linear matrix inequalities (LMIs) which can be easily solved. Based on this condition, a state-feedback controller is designed such that the consensus of the closed-loop system is mean square stable (MSS) with known communication failure processes.

Notation. Throughout this paper, $\mathbb{R}^n$, $\mathbb{R}^{n\times m}$, $\mathbb{S}^{n\times n}$ represent the n-dimensional Euclidean space, the set of all $n \times m$ real matrices and the $n \times n$ real symmetric positive definite matrices, respectively; $\mathbb{Z}_+$ is the set of non-negative integers; $(\Omega, \mathcal{F}, P)$ denotes a complete probability space; the superscript "T" represents the transpose; for Hermitian matrices $X = X^T \in \mathbb{R}^{n\times n}$ and $Y = Y^T \in \mathbb{R}^{n\times n}$, the notation $X \geq Y$ (respectively, $X > Y$) means that the matrix $X - Y$ is positive semi-definite (respectively, positive definite); I_n is the $n \times n$ identity matrix; $\mathbf{E}(\cdot)$ denotes the expectation operator with respect to some probability measure; $(M)_{ij}$ refers to the ith row, jth column element of matrix M; $\|\cdot\|$ represents the Euclidean norm for a vector and the spectral norm for a matrix; the symbol $\otimes$ denotes the Kronecker product; trace$(\cdot)$ is the trace of matrix; diag$(M_1, M_2, \ldots, M_N)$ is a block-diagonal matrix with diagonal blocks $M_1, M_2, \ldots, M_N$; $\mathbf{1}_N$ is defined as $\mathbf{1}_N = (1, 1, \ldots, 1)^T \in \mathbb{R}^N$ and $\mathbf{0}_N$ is defined similarly, that is, $\mathbf{0}_N = [0, 0, \ldots, 0]^T \in \mathbb{R}^N$.

2 PRELIMINARIES

A *directed graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of a finite *vertex set* $\mathcal{V}$ and an *edge set* $\mathcal{E} \subset \mathcal{V}^2$. Suppose there are n vertices in $\mathcal{V}$, then the graph has an *order* n and each vertex can be uniquely labeled by an integer i belonging to a finite index set $\mathcal{I} = \{1, 2, \ldots, n\}$. Each edge can be denoted by an ordered pair of distinct vertices (v_i, v_j) where v_j is the head and v_i is the tail, that is, the edge points from v_i to v_j with no self-loop. Each edge $(v_i, v_j) \in \mathcal{E}$ corresponds to the information transmission from agent j to agent i. The graph with the property that for any $(v_i, v_j) \in \mathcal{E} \Leftrightarrow (v_j, v_i) \in \mathcal{E}$ is said to be *symmetric* or *undirected*. The *in (out)-degree* of v_i, denoted by $d_i(v_i)$ $(d_o(v_i))$, is the number of edges with v_i as its tail(head). If $(v_i, v_j) \in \mathcal{E}$, then v_j is one of the *neighbors* of v_i. The set of neighbors of v_i is denoted by $\mathcal{N}_i = \{v_j \in \mathcal{V} : (v_i, v_j) \in \mathcal{E}\}$. An *adjacency matrix* of graph $\mathcal{G}$ with order n is an $n \times n$ matrix $\mathcal{A} = \{a_{ij}\}$ defined as

$$a_{ij} = \begin{cases} 1 & \text{if } (v_i, v_j) \in \mathcal{E}, \\ 0 & \text{otherwise.} \end{cases}$$

An *in-degree matrix* of graph $\mathcal{G}$ with order n is an $n \times n$ matrix $\mathcal{D} = \text{diag}\{d_{11}, d_{22}, \ldots, d_{nn}\}$ where $d_{ii} = \sum_{v_j \in \mathcal{N}_i} a_{ij}$. A *Laplacian matrix* $\mathcal{L}$ of graph $\mathcal{G}$ with order n is an $n \times n$ matrix defined as follows:

$$\mathcal{L} = \mathcal{D} - \mathcal{A}.$$

Let us consider a MAS with n agents. The discrete-time linear dynamics of agent i can be described by the following equation:

$$x_i(k+1) = Ax_i(k) + Bu_i(k), \quad i \in \mathcal{I} \tag{1}$$

where $x_i(k) \in \mathbb{R}^m$ is the system state, $u_i(k) \in \mathbb{R}^l$ is the control input, $A \in \mathbb{R}^{m \times m}, B \in \mathbb{R}^{m \times l}$, $k \in \mathbb{Z}_+$ is the time step, $x_i(0) \triangleq x_{i0}$ is the initial state.

Suppose the communication failure between agents i and j of the MAS (1) behaves in an independent way, that is, agent i can receive data from agent j does not necessarily mean agent j can receive data from agent i. The control input of the ith agent with communication failure is

$$u_i(k) = K \sum_{v_j \in \mathcal{N}_i} \gamma_{ij}(k)(x_i(k) - x_j(k)) \tag{2}$$

where $K \in \mathbb{R}^{l \times m}$ is the controller gain to be designed, $\gamma_{ij}(k)$ denotes the communication status from agent j to i at time k (1 for successful communication, 0 for unsuccessful communication). The communication status process is assumed to be a discrete-time homogeneous Markov chain taking values in a finite set $\mathcal{W} = \{0, 1\}$ with transition probability matrix

$$\Pi_{ij} = \begin{bmatrix} 1 - \beta_{ij} & \beta_{ij} \\ \alpha_{ij} & 1 - \alpha_{ij} \end{bmatrix}, \tag{3}$$

where $0 \leq \Pr(\gamma_{ij}(k+1) = 0 \mid \gamma_{ij}(k) = 1) = \alpha_{ij} \leq 1$ and $0 \leq \Pr(\gamma_{ij}(k+1) = 1 \mid \gamma_{ij}(k) = 0) = \beta_{ij} \leq 1$ are called the failure probability and the recovery probability, respectively. To simplify the expression, $\mathscr{A} \triangleq (\alpha_{ij})$, $\mathscr{B} \triangleq (\beta_{ij})$ are used to denote the failure probability matrix and recovery probability matrix, respectively. Notice that the communication failure model in (2) indicates that the error signal $x_i(k) - x_j(k)$ will not be employed by the controller u_i at time k when the j communication channel fails.

Under the above formulation, if there is no communication channel from agent j to i, that is, no edge (v_i, v_j) in the graph and $a_{ij} = 0$ in the Laplacian matrix, then the absence channel is treated as an 'ineffective' channel with $\gamma_{ij}(k) = 0$ for all k, with the failure and recovery probabilities assigned to be $\alpha_{ij} = 1$ and $\beta_{ij} = 0$, respectively. By treating the absent communication channels this way, the original problem is equivalent to considering the communication failure problem of an MAS with a complete graph governed by known communication failure probability $\gamma_{ij}(k)$ in the communication channel from j to i at time k. Consequently, the Laplacian matrix at time k can be rewritten as

$$L(k) = \begin{bmatrix} \sum_{j \neq 1} \gamma_{1j}(k) & -\gamma_{12}(k) & \ldots & -\gamma_{1n}(k) \\ \ldots & \ldots & \ldots & \ldots \\ -\gamma_{n1}(k) & -\gamma_{n2}(k) & \ldots & \sum_{j \neq n} \gamma_{nj}(k) \end{bmatrix} \tag{4}$$

where $L(k) \in \mathcal{L}^0 \triangleq \{\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_d\}$ such that $\mathcal{L}^0$ contains all possible Laplacian matrices of the MAS. Here, $\max d = 2^{\tilde{d}}$ where $\tilde{d} \triangleq \sum_{i=1}^n d_{ii}$ is the total number of effective communication channels in the graph (that is, the number of edges of the complete graph subtracting those 'ineffective' edges).

Definition 1. *A communication failure process is said to be Markovian if it is a discrete-time homogenous Markov chain defined in a complete probability space*

$(\Omega, \mathcal{F}, P)$, and takes value in $\mathcal{W}$ with known transition probability matrix $\Pi_1 \triangleq (\lambda_{ij}) \in \mathbb{R}^{s_0 \times s_0}$.

Denote by X the concatenation of vectors $x_1, x_2, \ldots, x_n$, that is, $X = (x_1^T, x_2^T, \ldots, x_n^T)^T$, by (1) and (2), the closed-loop dynamics of MASs (1) can be rewritten in a matrix form

$$X(k+1) = (\bar{A} + \bar{B}(k))X(k), \tag{5}$$

where $\bar{A} = I_n \otimes A$, $\bar{B}(k) = L(k) \otimes (BK)$, $L(k)$ takes values from $\mathcal{L}^0$. In this paper, we always assume that every agent could receive information of some other agents with a non-zero probability during the whole control process.

The mode transition probability $\pi_{\mathcal{L}_a\mathcal{L}_b}$ from $\mathcal{L}_a$ to $\mathcal{L}_b$ is given by

$$\pi_{\mathcal{L}_a\mathcal{L}_b} = \prod_{i,j=1, i\neq j}^{n} (\Gamma_{\mathcal{L}_a\mathcal{L}_b})_{ij} \tag{6}$$

for all $a, b = 1, \ldots, d$, which satisfies $\sum_{b=1}^{d} \pi_{\mathcal{L}_a\mathcal{L}_b} = 1$, where

$$(\Gamma_{\mathcal{L}_a\mathcal{L}_b})_{ij} = \begin{cases} 1-\alpha_{ij} & \text{if } (\mathcal{L}_b)_{ij} = (\mathcal{L}_a)_{ij} = -1, \\ 1-\beta_{ij} & \text{if } (\mathcal{L}_b)_{ij} = (\mathcal{L}_a)_{ij} = 0, \\ \alpha_{ij} & \text{if } (\mathcal{L}_b)_{ij} = 0, (\mathcal{L}_a)_{ij} = -1, \\ \beta_{ij} & \text{if } (\mathcal{L}_b)_{ij} = -1, (\mathcal{L}_a)_{ij} = 0. \end{cases}$$

3 STABILITY ANALYSIS AND CONTROLLER SYNTHESIS

In this section, we first give definitions on stability and consensus. Then we consider the consensus reaching and controller design problems for MASs with communication failure characteristics described in Section 2.

Definition 2. *MAS (5) with Markovian communication failure process (2) is said to be mean square stable (MSS) if*

$$\lim_{k\to\infty} \mathbf{E}(\|X(k)\|^2 \mid X(0)) = 0 \tag{7}$$

for any initial state $X(0) \in \mathbb{R}^{mn}$.

Definition 3. *Agents of MAS (5) with Markovian communication failure process (2) are said to reach consensus if*

$$\lim_{k\to\infty} \mathbf{E}(x_i(k) - x_j(k)) = 0 \tag{8}$$

for all $i, j \in \mathcal{I}$.

Now we are in the position to present the main contribution of this paper.

Theorem 1. *Consider MAS (5) with Markovian communication failure (2), given the controller gain matrix K, consensus is reached if there exist real matrices* $P_{\mathcal{U}} \in \mathbb{S}^{(n-1)m\times(n-1)m}$, $\mathcal{U} \in \mathcal{L}^0$, *such that*

$$\sum_{\mathcal{V}\in\mathcal{L}^0} \pi_{\mathcal{U}\mathcal{V}} \Phi_{\mathcal{V}}^T P_{\mathcal{V}} \Phi_{\mathcal{V}} - P_{\mathcal{U}} < 0, \quad \forall \mathcal{U} \in \mathcal{L}^0, \tag{9}$$

where $\Phi_{\mathcal{V}} = I_{n-1} \otimes A + \Lambda_{\mathcal{V}} \otimes BK$, $\Lambda_{\mathcal{V}} = T_o^T \mathcal{V} T_o$ *and* T_o *is the orthogonal basis for the null space of* $\mathbf{1}_n$.

Proof. Construct orthogonal matrix $T = \begin{bmatrix} \frac{1}{\sqrt{n}}\mathbf{1}_n & T_o \end{bmatrix} \in \mathbb{R}^{n\times n}$, where T_o is the orthogonal complement of $\mathbf{1}_n$ satisfying $T_o^T T_o = \mathbf{I}_{n-1}$. Since T_o is the orthogonal complement of $\mathbf{1}_n$, then

$$\mathbf{1}_n^T T_o = \mathbf{0}_{n-1}^T, T^T T = \begin{bmatrix} \frac{1}{\sqrt{n}}\mathbf{1}_n^T \\ T_o^T \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{n}}\mathbf{1}_n & T_o \end{bmatrix} = \mathbf{I}_n;$$

which means T is an orthogonal matrix and $T^T \mathcal{L} T$ is a similarity transformation. Partition the Laplacian matrix $\mathcal{L}$ and matrix T^T conformably:

$$T^T = \begin{bmatrix} \frac{1}{\sqrt{n}} & \frac{1}{\sqrt{n}}\mathbf{1}_{n-1}^T \\ T_{o1}^T & T_{o2}^T \end{bmatrix},$$

then

$$T^T \mathcal{L} T = \begin{bmatrix} 0 & A_{\mathcal{L}} \\ \mathbf{0}_{n-1} & \Lambda_{\mathcal{L}} \end{bmatrix}, \tag{10}$$

where $A_{\mathcal{L}} = \frac{1}{\sqrt{n}}\mathbf{1}_n^T \mathcal{L} T_o$ and $\Lambda_{\mathcal{L}} = T_o^T \mathcal{L} T_o$.

If the graph is undirected, that is, $\mathcal{L} = \mathcal{L}^T$. Thus

$$T^T \mathcal{L} T = \begin{bmatrix} 0 & (\frac{1}{\sqrt{n}} T_o^T \mathcal{L} \mathbf{1}_n)^T \\ \mathbf{0}_{n-1} & T_o^T \mathcal{L} T_o \end{bmatrix} = \begin{bmatrix} 0 & \mathbf{0}_{n-1}^T \\ \mathbf{0}_{n-1} & \Lambda_{\mathcal{L}} \end{bmatrix},$$

Denote $\tilde{X}(k) = (T \otimes I_m)^T X(k)$, the dynamics of the closed-loop system can be described by

$$\begin{aligned} &\tilde{X}(k+1) \\ &= (I_n \otimes A)\tilde{X}(k) \quad + \begin{bmatrix} 0 \otimes I_m & A_{L(k)} \otimes (BK) \\ \mathbf{0}_{n-1} \otimes I_m & \Lambda_{L(k)} \otimes (BK) \end{bmatrix} \tilde{X}(k), \end{aligned}$$

where $A_{L(k)} = \frac{1}{\sqrt{n}}\mathbf{1}_n^T L(k) T_o$, $\Lambda_{L(k)} = T_o^T L(k) T_o$.

Define

$$\tilde{X}(k+1) = \begin{bmatrix} X_1(k+1) \\ X_2(k+1) \end{bmatrix},$$

where

$$\begin{aligned} X_1(k+1) &= AX_1(k) + (A_{L(k)} \otimes (BK))X_2(k), \quad (11) \\ X_2(k+1) &= \Phi_{L(k)} X_2(k). \quad (12) \end{aligned}$$

where $\Phi_{L(k)} = I_{n-1} \otimes A + \Lambda_{L(k)} \otimes BK$. Let

$$z_i(k) = \sum_{v_j\in\mathcal{N}_i} \gamma_{ij}(k)(x_i(k) - x_j(k)),$$

$$Z(k) = (z_1^T(k), z_2^T(k), \dots, z_n^T(k))^T,$$

then

$$Z(k) = (L(k) \otimes I_m)X(k)$$

where $L(k)$ is given by (4), and

$$(T^T \otimes I_m)Z(k) = \begin{bmatrix} A_{L(k)} \otimes I_m \\ \Lambda_{L(k)} \otimes I_m \end{bmatrix} X_2(k),$$

where $T^T L(k) T = \begin{bmatrix} 0 & A_{L(k)} \\ \mathbf{0}_{n-1} & \Lambda_{L(k)} \end{bmatrix}$. Thus, $Z(k)$ is MSS if $X_2(k)$ is MSS, that is, the consensus of MASs (5) is reached if (12) is MSS.

Define a Lyapunov function as follows:

$$V(k, L(k)) = X_2^T(k) P_{L(k)} X_2(k)$$

where $P_{L(k)} > 0$ are matrices to be determined.

Let $\mathcal{U} = L(k)$, $\mathcal{V} = L(k+1)$, then

$$\begin{aligned} & \mathbf{E}(V(k+1, L(k+1))|L(k) = \mathcal{U}) - V(k, \mathcal{U}) \\ = & \mathbf{E}(X_2^T(k+1) P_{L(k+1)} X_2(k+1)|L(k) = \mathcal{U}) \\ & -X_2^T(k) P_{\mathcal{U}} X_2(k) \\ = & X_2^T(k) \left[\sum_{\substack{\mathcal{V} \in \mathcal{L}^0 \\ j=1,\dots,s_0}} \pi_{\mathcal{U}\mathcal{V}} \Phi_{\mathcal{V}}^T P_{\mathcal{V}} \Phi_{\mathcal{V}} - P_{\mathcal{U}} \right] X_2(k) \\ < & 0 \end{aligned}$$

for any $X_2(k) \neq 0$ if inequality (9) holds. Hence $\lim_{k\to\infty} \mathbf{E}(V(k, L(k))) = 0$, which ensures

$$\lim_{k\to\infty} \mathbf{E}(\|X_2(k; x_0)\|^2) = 0,$$

that is, system (12) is MSS. This completes the proof. □

Now, based on Theorem 1 and Schur complement Lemma, a controller design method is readily derived in the following theorem.

Theorem 2. *Consider MAS (5) of n agents with Markovian communication failure (2), consensus of the closed-loop system is reached if there exist $\mathcal{X}_i \in \mathbb{S}^{(n-1)m\times(n-1)m}$, $i = 1, 2, \dots, d$, $\mathcal{Y} \in \mathbb{R}^{l\times m}$ and $G \in \mathbb{R}^{m\times m}$ such that $\forall i = 1, 2, \dots, d$,*

$$\begin{bmatrix} -I_{n-1} \otimes (G + G^T) + \mathcal{X}_i & \Theta_i^T \\ \Theta_i & -\Lambda \end{bmatrix} < 0 \qquad (13)$$

where

$$\begin{aligned} \Lambda &= diag(\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_d), \\ \Theta_i &= \left[\sqrt{\pi_{i\mathcal{L}_1}} \Xi_{\mathcal{L}_1}^T \quad \sqrt{\pi_{i\mathcal{L}_2}} \Xi_{\mathcal{L}_2}^T \quad \cdots \quad \sqrt{\pi_{i\mathcal{L}_d}} \Xi_{\mathcal{L}_d}^T \right]^T, \\ \Xi_{\mathcal{V}} &= I_{n-1} \otimes AG + \Lambda_{\mathcal{V}} \otimes B\mathcal{Y}, \\ \Lambda_{\mathcal{V}} &= T_o^T \mathcal{V} T_o, \quad \mathcal{V} \in \mathcal{L}^0 \end{aligned}$$

and T_o is the orthogonal basis for the null space of $\mathbf{1}_n$. Moreover, a consensus controller gain matrix in (2) is given by $K = \mathcal{Y} G^{-1}$.

Proof. Define
$\Psi_i \triangleq \left[\sqrt{\pi_{i\mathcal{L}_1}} \Phi_{\mathcal{L}_1}^T \quad \sqrt{\pi_{i\mathcal{L}_2}} \Phi_{\mathcal{L}_2}^T \quad \cdots \quad \sqrt{\pi_{i\mathcal{L}_d}} \Phi_{\mathcal{L}_d}^T \right]^T$, where $\Phi_{\mathcal{V}} = I_{n-1} \otimes A + \Lambda_{\mathcal{V}} \otimes BK$. Pre- and post-multiplying inequality (13) by $\left[\Psi_i \quad I_{(n-1)md}\right]$ and its transpose, respectively, then

$$\Psi_i \mathcal{X}_i \Psi_i^T - \Lambda < 0,$$

that is,

$$\begin{bmatrix} -\mathcal{X}_i^{-1} & \Psi_i^T \\ \Psi_i & -\Lambda \end{bmatrix} < 0. \qquad (14)$$

Thus, inequality (14) is equivalent to inequality (9) by replacing P_i by $\mathcal{X}_i^{-1}$. This completes the proof. □

4 CONCLUSIONS

In this paper, a consensus control problem of MASs with communication failure between agents has been studied. A sufficient condition on consensus reaching problem is established in terms of the feasibility of some LMIs. In addition, a state-feedback consensus controller is designed to make the closed-loop system reach consensus. A numerical example has been given to demonstrate the effectiveness of the proposed results.

ACKNOWLEDGEMENTS

The research was supported by HKU CRCG 200907176129.

REFERENCES

Bender, J. G. (1991). An overview of systems studies of automated highway systems. *IEEE Trans. Veh. Technol.*, 40(1):82–99.

Liu, T. F. and Passino, K. M. (2004). Stable social foraging swarms in a noisy environment. *IEEE Trans. Automat. Control*, 49(1):30–44.

Olfati-Saber, R., Fax, J. A., and Murray, R. M. (Jan, 2007). Consensus and cooperation in networked multi-agent systems. In *Proceedings of the IEEE*, volume 95, pages 215–233.

Olfati-Saber, R. and Murray, R. M. (2004). Consensus problems in networks of agents with switching topology and time-delays. *IEEE Trans. Automat. Control*, 49(9):1520–1533.

Paganini, F., Wang, Z. K., Doyle, J. C., and Low, S. H. (2005). Congestion control for high performance, stability, and fairness in general networks. *IEEE/ACM Trans. Networking*, 13(1):43–56.

Xiong, J. and Lam, J. (2007). Stabilization of linear systems over networks with bounded packet loss. *Automatica*, 43(1):80–87.

MANIPULATOR-DEPLOYED SYSTEMS FOR SURFACE DECONTAMINATION IN NUCLEAR FACILITIES

Jan Bremmer, Sascha Gentes and Nadine Gabor
Karlsruhe Institute of Technology (KIT), Institute for Technology and Management in Construction (TMB)
Technology and Management for the Decommissioning of Nuclear Facilities
Am Fasanengarten, Geb. 50.31, 76131 Karlsruhe, Germany
{jan.bremmer, sascha.gentes, nadine.gabor}@kit.edu

Keywords: AMANDA, MANOLA, Manipulator, Vacuum Technology, CompactRIO.

Abstract: Due to the phasing out of nuclear energy in Germany there are a growing number of nuclear facilities that have to be decommissioned within the next years. In this context a multitude of surfaces in nuclear facilities have to be decontaminated. Manipulator-deployed systems offer a suitable solution and are properly designed for this kind of task. Beyond that they can be used for the processing of surfaces in civil as well as industrial fields of application, e.g. the stripping of coatings on metal surfaces. By the use of a suitable supporting system various attachments like a milling machine or laser can be carried and operated on walls and ceilings. Vacuum suction plates guarantee the interconnection between the supporting system and the object to be treated. Due to the intricate processes as well as the required flexibility arising from the multi purpose use with milling or laser attachments a robust control system with a high performance and a high level of customization is required. The following article introduces to you the systems in detail.

1 INTRODUCTION

Due to the phasing out of nuclear energy in Germany there are a growing number of nuclear facilities that have to be decommissioned within the next years. In order to successfully dismantle and decommission these facilities a great amount and variety of processes and actions is required.

Among other things these include the qualified decontamination of surfaces. The associated tasks are manifold, ranging from simple cleaning to complete surface ablation. The latter is particularly challenging due to the multitude of surfaces in nuclear facilities that have to be decontaminated.

In this regard, effectiveness and economic efficiency play an important role as well as the avoidance of secondary waste and cross contamination (Gentes, 2006: 416, 417). Therefore the selection of an appropriate process and system is crucial.

But only a few of the currently available processes for the decontamination of surfaces fully meet the required criteria and are applicable for this kind of task. Therefore the nuclear industry calls for more suitable and efficient decommissioning technologies.

The Institute of Technology and Management for the Decommissioning of Nuclear Facilities at the Karlsruhe Institute of Technology (KIT) has addressed itself to this task and thus is conducting research as well as is developing innovative processes and machines for this kind of purpose. All this is done in co-operation with industry partners that are focused on carrying out decommissioning activities in the field. This kind of co-operation helps to generate feedback in order to improve the machines.

2 THE BASIC IDEA

The decontamination of surfaces in nuclear facilities is an intricate and time consuming process. It is very labour intensive, because most techniques are hand-operated or difficult to handle by just one person. Compared to the output of a machine the performance of labour is very low. Thus more labour is required in order to be cost-effective.

Furthermore the timeframe for works conducted in the hot zone, which is the controlled area inside of a nuclear facility, is very limited per worker and shift (HVBG, 2004: app. 2), due to the required use of breathing apparatuses. Because of that several teams have to be employed on alternating shifts to ensure a proper exchange of labour which finally results in a better practical performance.

Another important aspect is the high effort that comes along with many of the techniques being used for the decontamination of surfaces. In numerous cases scaffolding is needed, especially in rooms with big ceiling heights. Besides that extra time is required for the set-up of equipment or machinery. But the time spent on works preparation reduces the overall period available for decontamination. Thus it makes even suitable techniques less effective.

Based on these facts the idea was born to construct and build a manipulator that is able to climb on walls and ceilings autonomously, in order to decontaminate surfaces by means of a milling attachment or laser. The manipulator requires no scaffolding or elaborate set-up. Besides that there is only one operator needed to run the machine. Finally the use of a manipulator results in a greater economic efficiency.

3 MACHINE SPECIFICATIONS

Several requirements have to be met for the successful decontamination of surfaces in nuclear facilities. Besides the requirements requested by the clients, e.g. regarding safety and economic efficiency of the particular technique, there are requirements that originate from the structural conditions of a nuclear facility.

The relevant criteria have to be taken into account when setting up a manipulator system. Hence the following specifications have to be implemented:

The manipulator is required to have a low self-weight. This enables the operator to move the system through the facility. In some cases the support of a trolley might be useful.

Furthermore a modular layout allows the operator to move the manipulator through small and narrow openings by disassembling it partly.

In this case a low single-weight of each module is essential to allow lifting them by one person only.

Moreover the disassembly as well as assembly of the manipulator should be feasible in a very short period of time. Therefore the use of quick fasteners is recommended. The entire system has to be flexible and rugged.

In order to meet the performance requirements an output of at least 10 m^2 per hour is essential and the minimum requirement of the clients.

Beyond that an autonomous operation of the manipulator is advisable to keep the labour costs on a low level.

Not less important are safety features that assure a safe operation of the entire system. Here an emergency shut-off is the minimum requirement.

But due to the fact that the manipulator can be operated on walls and ceilings by means of vacuum technology it is also a prerequisite to prevent the system from falling off the object that is being treated.

In this regard pressure monitoring and the use of check valves is important.

Furthermore the control system needs a proper set up that includes safety procedures.

4 MACHINE BUILD-UP

4.1 Basic System AMANDA I

According to the basic idea, a first manipulator system was built in the course of the research project AMANDA.

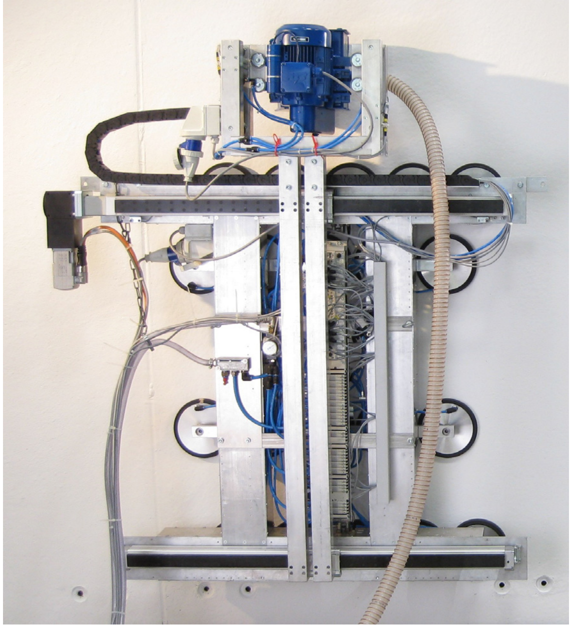


Figure 1: AMANDA I in wall operation.

AMANDA stands for **A**utonomous **Man**ipulator for **D**econtamination **A**ssignments. The system can be operated on walls and ceilings by using vacuum technology. It is equipped with a milling attachment for decontamination purposes.

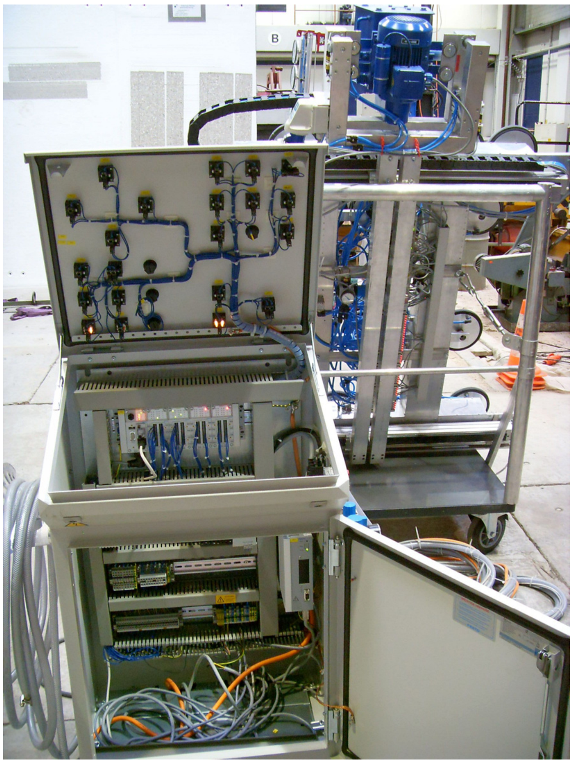


Figure 2: AMANDA I with control unit.

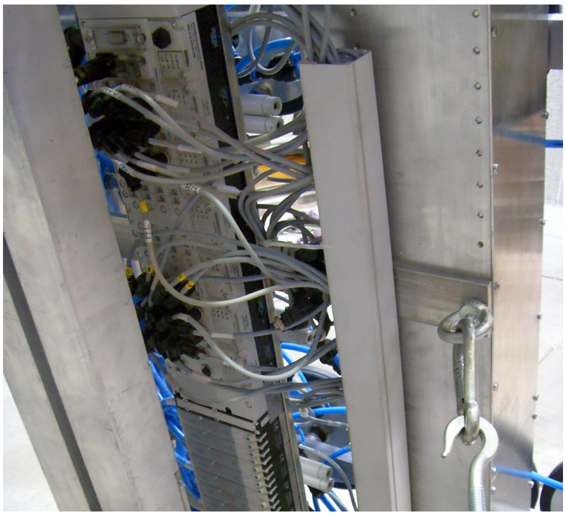


Figure 3: CPX Valve Terminal of AMANDA I.

The control unit which is shown in figure 2 is based on a FESTO SPS. It is mounted in a switch box underneath the control panel. The SPS is linked with the CPX valve terminal on the manipulator via cable. All pneumatic valves as well as all sensors for the positioning of the pistons and for pressure monitoring are connected to the CPX valve terminal. The valve terminal is shown in figure 3.

In contrast to the valves and sensors, the servo drive which moves the linear drive unit is directly linked with the SPS by cable.

4.2 Successor System MANOLA

Based on the principle of the stand-alone manipulator for decontamination assignments AMANDA I the successor system MANOLA is currently under construction at the KIT. MANOLA is funded by the German Ministry for Education and Research under the reference key 02S8548.

MANOLA stands for **Man**ipulator **O**perated **L**aser **A**blation. The support system is operated with vacuum technology as well. Instead of a milling attachment MANOLA carries a laser system that is used for the ablation of both, contaminated and uncontaminated concrete surfaces and coatings. MANOLA is built according to the specifications listed in section 3.

To allow the transportation of MANOLA inside of a nuclear facility the system is equipped with a trolley. The trolley consists of an undercarriage fitted with rubber tracks, and a loading platform.
Figure 2 shows a visualization of MANOLA in wall operation. Beyond that, MANOLA can be operated on ceilings as well.

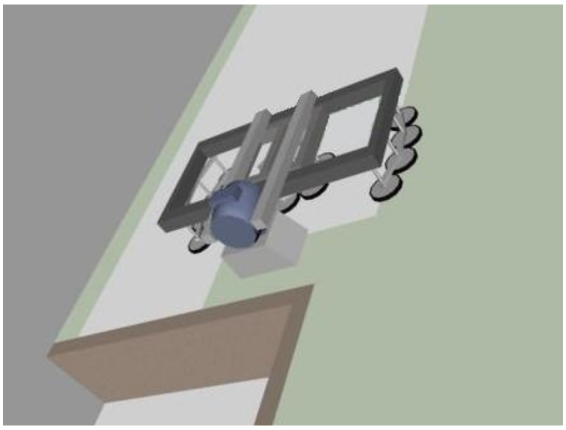


Figure 4: MANOLA in wall operation.

5 MANOLA CONTROL UNIT

5.1 Processor And Chassis

The control unit of MANOLA is based on the CompactRIO System that is distributed by National Instruments. ‘National Instruments CompactRIO is a small rugged industrial control and acquisition system powered by reconfigurable I/O (RIO) FPGA technology for ultrahigh performance and customization. NI CompactRIO incorporates a real-time processor and reconfigurable FPGA for reliable stand-alone embedded or distributed applications, and hot-swappable industrial I/O modules with built-in signal conditioning for direct connection to sensors and actuators' (National Instruments, 2009).

The setup of the system applicable for MANOLA consists of an embedded real-time processor with 800 MHz, 512 MB DDR2 RAM, and 4 GB storage. The processor is connected to a 4-slot Virtex-5 LX50 reconfigurable chassis that will be installed on the MANOLA trolley. Furthermore two 8-slot deterministic chassis will be installed on the manipulator itself for embedding all sensors and actuators of MANOLA. Both chassis on the manipulator will be linked with the main chassis on the trolley in daisy chain mode via Ethernet cable, and the main chassis on the trolley is linked via WLAN with a laptop that serves as the MANOLA Control Panel. Beyond all these components there are WLAN cameras installed on each, the trolley and the manipulator, for process monitoring by the operator.

5.2 Sensors

Various sensors are part of the MANOLA control system. They are used for different purposes like positioning, vacuum and pressure monitoring, position monitoring of pistons as well as scanning and evaluating the object which has to be treated. All sensors provide necessary input to the manipulator for a proper operation.

First, there are four distance laser sensors used as a simplified positioning system for MANOLA. The four sensors are fixed to a traverse that carries the laser processing head and runs over the main frame of the manipulator. By moving the distance laser sensors over the entire frame, many different reading points can be generated and used to map the borderlines given by adjacent parts of the building.

Second, a laser scanner is used for scanning and evaluating the surface area of the object to be treated. The laser scanner is part of the processing head which also includes the optics of the laser unit for the ablation process, and it enables the machine to detect disruptive objects like offsets on the surface, nails, screws, pipes, etc. All these different kinds of objects may interfere with the operation of the laser. Thus it is very important to detect them in order to protect the laser processing head.

Third, there are two ultrasonic sensors attached to the processing head. The ultrasonic sensors are used for the detection of obstacles that may appear in front of the processing head during treatment of the surface area. In any case of interference the process will be paused immediately in order to protect the processing head. Then the processing head will be moved up by an electric drive and the suspect area will be scanned with the laser scanner if applicable. Depending on the size of the obstacle the manipulator will be moved forward or around the localised object. The integrated decision making process is based on the input of the various sensors of the processing head.

Fourth, pressure and vacuum monitoring is an essential part of the control system. Due to the fact that the manipulator is operated with vacuum technology it is security-relevant and prevents MANOLA from falling off the wall. The vacuum for the suction plates is produced by sending pressurized air through vacuum generators that operate according to the Venturi Principle. Because the connection of the suction plates to the object depends on the pressure as well as on the vacuum, both, pressure and vacuum, need to be monitored.

Fifth, several small size sensors are required for monitoring the exact positions of the pistons of all pneumatic cylinders. Every time when a piston is moved the control system asks for a feedback, if the piston reached its destination. If this case is true, further operations can proceed. In a false case, further operations have to be put on hold due to the occurrence of an error. Only after assessing and removing the error by the operator further operation may proceed.

5.3 Actuators

Besides the sensors MANOLA includes different actuators. The actuators are used for the operation of the supporting frame, the traverse and the processing head. The actuators of MANOLA are a rack drive, a rotary module, a servo drive as well as a couple of pneumatic control valves.

The rack drive joins two important tasks. It moves the traverse and processing head for ablation processing as well as the supporting frame which includes the main frame and sub frame for pacing. By linking the rack drive to the component that has to be moved only one drive is needed. Thus the overall weight of the manipulator is kept on a low level.

In order to turn the manipulator by 45, 90 or any degrees to change the direction of processing a rotary module is built-in to the center of the sub frame. When the manipulator is rotated the suction plates of the main frame are released from the object, e.g. a wall, and the rotary module can be activated. After the manipulator is moved into its final direction, the suction plates of the main frame are sucked to the object again. During the whole rotation process the suction plates of the sub frame stay in contact with the object (wall, ceiling, etc.).

As described in the previous section, the processing head is moved down for processing the surface area and moved up when the processing is paused or finished. Lifting and kneeling is executed by a servo drive. Therefore the servo drive is mounted between the outer end of the traverse and the processing head. Input generated by the laser scanner and the ultrasonic sensors provide necessary input for the control system to actuate the servo drive. Beyond that further input is provided via control panel by the operator.

Last but not least a couple of pneumatic valves are integral part of the pneumatic system and the vacuum system. During pacing the relevant pneumatic valves are activated. They control the airflow necessary for the pneumatic cylinders and the vacuum generators. Owing to the fact that the pneumatic valves must not change their operating status in case of a drop of voltage, bistable valves are being used.

5.4 NI-Modules And Connectors

All actuators and sensors have to be linked with the control unit. Due to the great variety of connector types this requires an adaptable system. This is implemented by using the cRIO-System with its many different NI-Modules that all fit in the same type of chassis. The cRIO chassis provides the chance to build in the relevant type of NI-Modules depending on the attachment and its relevant NI-Modules.

Table 1: MANOLA NI-Modules.

Pos.	Module	Connector Type	Device
1	NI 9201	Analog IN	Sensor
2	NI 9205	Analog IN	Sensor
3	NI 9263	Analog OUT	Actuator
4	NI 9421	Digital IN	Sensor
5	NI 9425	Digital IN	Sensor
6	NI 9472	Digital OUT	Actuator
7	NI 9477	Digital OUT	Actuator
8	NI 9853	CAN BUS	Actuator
9	NI 9870	RS 232	Sensor
10	NI 9871	RS 485	Actuator

As quick as the attachment can be changed the NI-Modules can be switched as well, even during operation. That keeps the system flexible and aligned with the customers needs. The following table provides you an overview of all existing connector types and the relevant NI-Modules.

6 CONCLUSIONS

Especially in areas where people are exposed to great hazards, the use of a manipulator provides an interesting solution. Nuclear facilities are one of those fields of application. Beyond that there are other areas as well where the use of a manipulator can be valuable. For example this applies to high structures that require costly scaffolding or other areas that are difficult to access.

By providing a compact, light and robust unit equipped with a milling attachment or laser, the customer disposes of a suitable machine for the treatment of various kinds of surfaces. Furthermore the manipulator comes with an intelligent and adaptable control system that provides a costumer friendly operability as well as expandability.

Relating to feedback of project managers in nuclear facilities under decommission, the use of manipulators is more than welcome and thus will be common in the future. The same is expected for other fields of use. But in order to meet these needs the current manipulator needs to be further developed. Thus the system provides a platform which is capable of being extended in the future.

REFERENCES

Gentes, S. (2006) *Bautechnik 83, Heft 6: Duennschichtiger Oberflächenabtrag von Beton in kerntechnischen Anlagen*, Berlin: Ernst & Sohn

HVBG (2004) *BGR 190:* Benutzung von Atemschutzgeraeten, HVBG

National Instruments (2009) *NI CompactRIO Reconfigurable Control and Acquisition System*, [Online], Available: *http://zone.ni.com/devzone/cda/tut/p/id/2856* [05 Jan 2010]

DESIGN AND EXPERIMENTAL VERIFICATION OF POWER-ASSISTED SMART DOOR SYSTEM FOR PASSENGER VEHICLE

Kum-Gil Sung, Min-Kyu Park
School of Mechanical and Automotive Engineering Technology
Yeungnam College of Science and Technology, Daegu 705-703, Korea
{kgsung, mk_park}@ync.ac.kr

Byoungsoo Lee
Department of Mechnical and Automotive Engineering, Keimyung University, Daegu 704-701, Korea
blee@kmu.ac.kr

Keywords: Smart Door, Power-assist, Passenger Vehicle, Human Friendly.

Abstract: In this study, to enhance the ease of opening and closing the doors of the passenger vehicle, a smart door with a power assist mechanism consisting of a motor and clutch was developed and tested. A power assist mechanism mounted within the vehicle's door is proposed and modeled. The required force necessary to control the designed mechanism during the vehicle's roll, pitch and the opening angle of the door has been established. Finally, the improvement in the ease of opening and closing doors by utilizing the designed ower assist mechanism was tested and proven through an experimental verification.

1 INTRODUCTION

Recently, in the automotive industry, research in creating intelligent parts by integrating various functions is actively conducted. The developed technologies are being commercialized rapidly in North America, Europe, and Korea. Among such intelligent parts, auto sliding door and electric power door systems are developed for doors of vehicles (Yoon, 2002, Grujicic, 2009). However, in most of the conducted research and products in the process of commercialization, studies on mechanisms for compensating changes in the necessary force to open and close doors of a vehicle parked at an angle is limited.

Depending on the structure, shape, and opening & closing force of the vehicle's door, the perceived convenience and safety by the user is affected greatly and it is very frequently used by passengers. Thus, when the door does not operate as intended by the passenger, the inconvenience is drastically felt. Traditionally, a part called the door checker which is attached between the door and the vehicle frame prevented the door from opening instantaneously when the vehicle is parked at an angle.

However, in limited space or when the vehicle is parked at an angle forward or sideways due to the weight of the door it will not become stationary at the passenger's desired position. Therefore, a new power-assist smart door which adjusts the opening and closing force depending on the roll and pitch angle of the vehicle is suggested. It utilizes a motor and clutch to power-assist the passenger while opening or closing the door of the vehicle. When this system is used no matter how the vehicle is positioned the door can be conveniently opened or closed without exerting oneself. The convenience of this kind of a system becomes more important when the weight of the door is heavy or for large passenger vehicles. In order to commercialize such a power-assist smart door system, in depth research in various areas such as power assist mechanism, controlling the opening and closing force, mechanism for understanding the intention of the passenger, dedicated motor and clutch development must be conducted. Among these, for developing the power-assist equipment, the mechanism for understanding the passenger's intention when getting on or off the vehicle and to actively control the necessary force to open or to close the door must be organized.

In this study, to enhance the ease of opening and closing the doors of the passenger vehicle, a smart door with a power-assist mechanism consisting of a motor and clutch is developed and a smart door with this mechanism will be designed. A power assist mechanism which can be mounted inside the door is suggested and modeling of a vehicle door with this mechanism conducted. Based on this, a power assist mechanism which is suitable for passenger vehicle is designed. A computer simulation is used to derive the necessary force to open or close the door for a given door opening angle at a given vehicle roll or pitch angle.

2 MODELING OF SMART DOOR

An illustration of the new power assist smart door which can alter the necessary force to change the force required to open or close the door when the vehicle is parked at a roll or pitch angle is provided in Figure 1. It consists of a rack and pinion, a clutch, and a reduction gear and motor. When this mechanism is applied, even when the vehicle is parked at an angle, the necessary force required to open or close the door will be the same as if the vehicle is parked on a flat level surface.

To establish the equation of motion for the power-assist mechanism attached smart door, in the model, as shown in Figure 2, the degrees of freedom was established by the roll ϕ_x and pitch ϕ_z angle of the vehicle and the angle of the door opening θ. The equation of motion is derived based on moment from the weight of the door and the assumption that the external force generated moment by the power assist mechanism.

Initially, as indicated in Figure 3 to obtain the moment of the door due to its own weight, it is assumed that the roll and pitch angle of the vehicle frame is fixed and that the roll and pitch angle(actually the reverse direction of the vehicle frame angle) is created relative to the gravity vector. Thus, the moment by the door's own weight $\tilde{T}_g$ can be summarized as the following.

$$\tilde{T}_g = \tilde{r}_g \times m\tilde{g}_{rot} \qquad (1)$$

Here, $\tilde{g}_{rot}$ is the rotated gravity vector caused by the roll and pitch angle and can be represented as equation (2).

$$\tilde{g}_{rot} = \begin{bmatrix} -g\sin\phi_z\cos\phi_x \\ -g\cos\phi_z\sin\phi_x \\ g\sin\phi_x \end{bmatrix} \qquad (2)$$

When equation (1) is reorganized by the y directional component, the moment in the y direction by the weight of the door becomes the following equation.

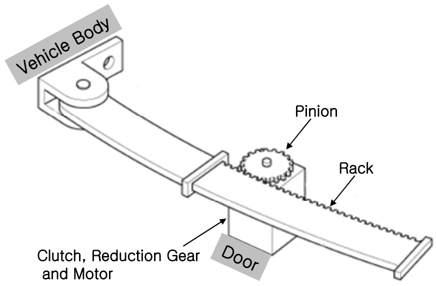


Figure 1: The configuration of the power-assist mechanism for the proposed smart door system.

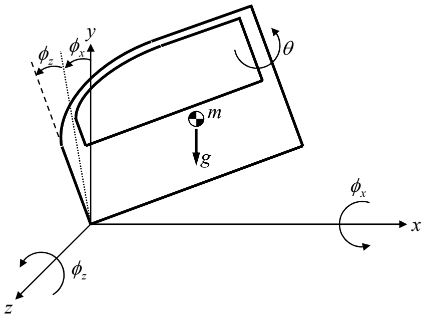


Figure 2: Coordinates for the dynamic model.

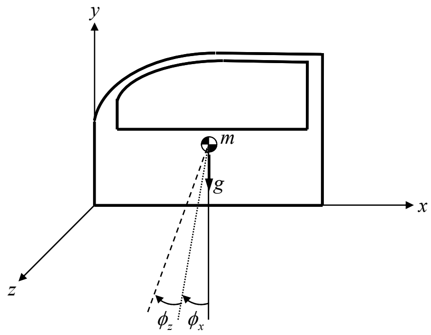


Figure 3: Gravity vector considering the roll and pitch angle of vehicle body.

$$T_{g\,y} = r_g mg\left(\sin\theta\sin\phi_z\cos\phi_x - \cos\theta\sin\phi_x\right) \qquad (3)$$

Next, as indicated in Figure 4 the moment of the door $\tilde{T}_F$ created by force of the power assist mechanism(F) is arranged as equation (4).

$$\tilde{T}_F = \tilde{r}_g \times \tilde{F} = F\tilde{r} \times \tilde{e}_t \tag{4}$$

Here, $\tilde{e}_t$ is the directional unit vector created by the power assist mechanism and can be represented as the following.

$$\tilde{e}_t = \begin{bmatrix} r\cos\theta - \varepsilon_x \\ 0 \\ -r\sin\theta - \varepsilon_z \end{bmatrix} / \sqrt{(r\cos\theta - \varepsilon_x)^2 + (r\sin\theta - \varepsilon_z)^2} \tag{5}$$

The following equation is the rearrangement of the moment in the y direction of the door caused by the power assist mechanism from equation (4).

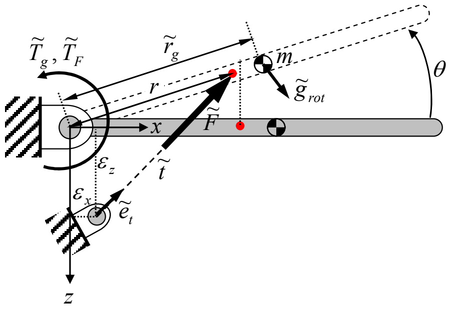


Figure 4: Free body diagram for dynamic model of the proposed smart door system.

$$T_{Fy} = \frac{rF(\varepsilon_x \sin\theta + \varepsilon_z \cos\theta)}{\sqrt{(r\cos\theta - \varepsilon_x)^2 + (r\sin\theta + \varepsilon_z)^2}} \tag{6}$$

When equation (2) and (6) is applied to the equation of motion it becomes the following.

$$\begin{aligned} J_y\ddot{\theta} + c\dot{\theta} &= T_{Fy} + T_{gy} + T_{fric} \\ &= \frac{rF(\varepsilon_x \sin\theta + \varepsilon_z \cos\theta)}{\sqrt{(r\cos\theta - \varepsilon_x)^2 + (r\sin\theta + \varepsilon_z)^2}} \\ &\quad + r_g mg(\sin\theta \sin\varphi_z \cos\varphi_x - \cos\theta \sin\varphi_x) + T_{fric} \end{aligned} \tag{7}$$

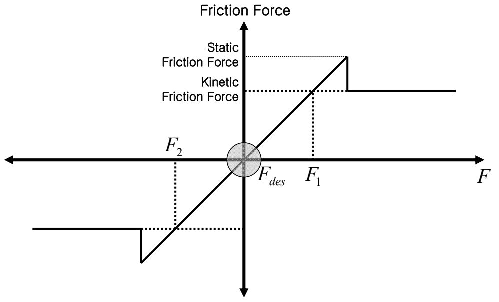
Here, J_y is the mass moment of inertia in the y direction, c is the damping coefficient of the door hinge, and T_{fric} is the frictional force that occurs when the door opened or closed.

3 DESIGN AND MANUFACTURE OF SMART DOOR

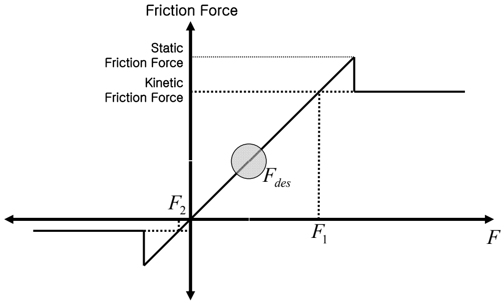
In order to design the power assist mechanism which will be actually mounted in the vehicle, it becomes necessary to estimate the maximum force necessary to open or close the door of the vehicle when it is in a pitch or roll position. Depending on the door opening position the desired control force F_{des} must be estimated. In this mechanism, the maximum force needed to open or close the door during a vehicle's pitch or roll position can be measured experimentally, however, the required control force of the mechanism must be set up depending on the pitch, roll, and door angle. As indicated in Figure 5, in this research, to set up the control force of the pitch, roll, and door opening angle the respective dynamic frictions F_1 and F_2 have been measured. Then the average value has been calculated and the control force has been set up as shown in Figure 6. Thus, as shown in Figure 5(a) only the forces needed to open or close the door when the vehicle is on level ground($F_1 - F_{des}$ and $F_{des} - F_2$) is required for vehicle in a slanted position as indicated on Figure 5(b).

The required control forces for roll ϕ_x, pitch ϕ_z, and door opening angle θ are within the range of -30~30°, -50~50° and 0~80° respectively as shown in Figure 6. As it can be verified from the results the required maximum control force is approximately 1486.3N.

In this study the maximum required control force is used as the boundary condition for the design and the motor, reduction gear, rack and pinion, and selection of parts were conducted.



(a) case I (flat road)



(b) case II (slope road)

Figure 5: The desired actuating force with respect to the vehicle body angle.

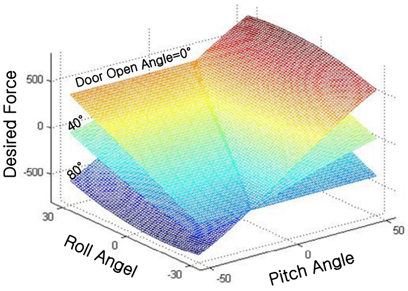


Figure 6: The required control force characteristics for designing power-assist device.

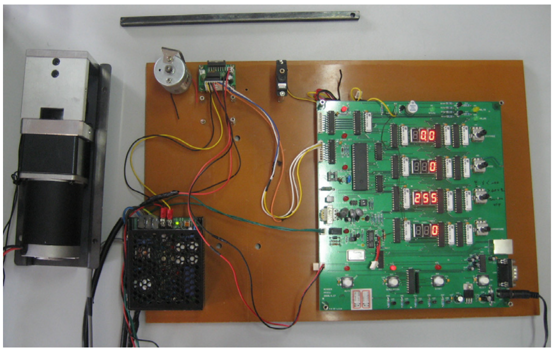


Figure 7: Power-assist mechanism control board.

In order to manufacture a suitable smart door, the door of an actual vehicle has been measured and the parameters are indicated. Based on these values, a control block diagram to control the smart door system is structured. The power assist mechanism control board is shown in Figure 7 and it is attached to the smart door system.

4 EVALUATION OF SMART DOOR

To test and evaluate the designed smart door, the test set up indicated in Figure 8 is constructed. An actual door of a passenger vehicle is attached to the frame. This frame can be set up in a vehicle roll position. The force generated by the power assist system is controlled by the computer D/A signal, the vehicle roll angle ϕ_x, door opening angle θ, and angular velocity $\dot{\theta}$ are measured by using sensors.

In order to investigate the characteristics of the smart door system, the roll angle of the vehicle was set at 15° as indicated in Figure 8 and the required opening and closing forces are actually measured. As it can be found from the experiment in Figure 9 when the door opening angle is adjusted between 0~80 ° if the power assist mechanism is not activated, the necessary force is 60~180N. But when the power assist mechanism is activated approximately 20~80N was required to open or close the door.

5 CONCLUSIONS

In this study, to enhance the ease of opening and closing the doors of the passenger vehicle, a smart door with a power assist mechanism consisting of a motor and clutch was developed and tested. For this purpose a power assist mechanism which can be mounted inside the door is suggested and the smart door is modeled and tested. The required control force for controlling the mechanism due to the vehicle's pitch, roll, and door opening angle is established.

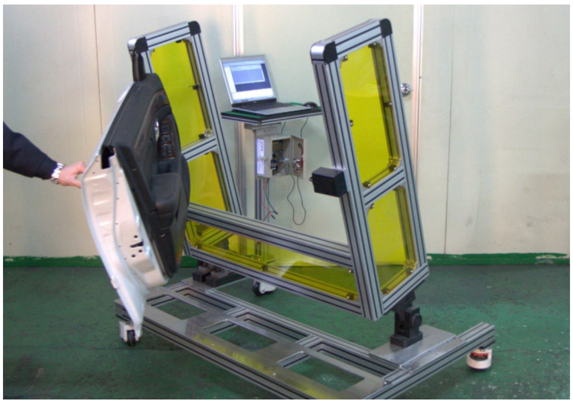


Figure 8: Experimental setup for the power-assisted smart door system.

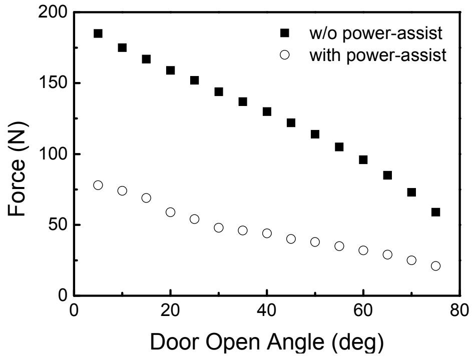


Figure 9: The control responses of the smart door system.

Based on the established results the power assist mechanism with the adequate opening and closing force was designed. A computer simulation was used to evaluate the designed power assist mechanism. The designed power assist mechanism installed smart door will be mounted on a vehicle and will be tested in the near future to prove the usefulness and convenience of the system.

REFERENCES

Grujicic, M., Arakere, G., Sellappan, V., Ziegert, J. C., Kocer, F. Y., and Schmueser, D., 2009. Multi-Disciplinary Design Optimization of a Composite Car Door for Structural Performance, NVH, Crashworthiness, Durability and Manufacturability. In *Multidiscipline Modeling in Materials and Structures*. Vol. 5, No. 1, pp. 1-28.

Yoon, D. H., Lee, J. H. and Yoo, J. H., 2002. Development and Optimization of Automobile Parts for Door Opening Angle Control System. In *Proceedings of the KSAE Annual Spring Conference*. pp. 1235~1240.

IMPACT OF DIFFERENT BIT RATES ON PERFORMANCE CHARACTERISTICS OF INDUSTRIAL WLAN SOLUTIONS

André Schimschar and Lutz Rauchhaupt
Institut für Automation und Kommunikation e.V. Magdeburg, Werner-Heisenberg-Str. 1, Magdeburg, Germany
{andre.schimschar, lutz.rauchhaupt}@ifak.eu

Keywords: Communication Systems, Performance Analysis, Reliability Analysis, Transmission Characteristics.

Abstract: Wireless communication is already widely used in industrial automation applications. Many solutions are available which consider industrial related requirements more or less. Thus, reliability and performance with respect to the specific requirements are the main concerns of industrial automation system users. Some of the wireless solutions base on the well-known Wireless LAN. The development of IEEE 802.11 was driven by the demand on highest possible data rates e.g. for video streaming applications. This paper proves whether the increased bit rates are appropriate to the requirements of industrial automation applications. A test methodology is introduced which can be used to get the necessary characteristic parameters with respect to the application field - industrial automation. The impact of bit rates on packet loss probability and therefore on the performance is analysed. Together with the transmission time for a successful transmission with one attempt a cost function has been developed. It shows for WLAN solutions, that the highest bit rate is not adequate to meet the reliability and performance requirements of industrial automation applications.

1 INTRODUCTION

Wireless communication technologies are widely spread in daily life. The price of wireless products is thereby the main design aspect with respect to the consumer market. Reliability is one of the minor design goals. Therefore, almost everyone has had negative experiences with such technologies and has developed concerns regarding the usage of wireless in industrial communication.

Also the most popular wireless communication technology WLAN, which uses the probabilistic media access control CSMA/CA, has disadvantages concerning time behaviour and reliability of packet transmissions. With setting up the bit rate the end-user has to find a trade-off between transmission time and loss probability of transmitted packets. Following the common sense the user takes a high bit rate for fast transmissions or a low bit rate for a high reliability of communication.

In this paper a test methodology is described which allows the investigation of wireless solutions with respect to industrial automation application requirements and conditions. This approach is used to define appropriate test conditions and to set-up a test system in order to investigate the influence of different bit rates on performance and reliability.

The transmission time is measured and the number of retransmission is investigated. Furthermore the loss probability is derived from the number of retransmission. Finally, a cost function has been developed in order to provide a guideline to find an appropriate trade-off between performance and reliability.

The paper is structured as follows. In chapter 2 the test methodology is presented and the main important influencing parameters are discussed and its values are defined. In chapter 3 the characteristic values are introduced which have been used for the investigations and the test system and its components are explained. The test results are discussed in chapter 4. The cost function is derived in chapter 5 and the conclusions out of it are presented in chapter 6.

2 TEST METHODOLOGY AND CONDITIONS

Because of the special requirements and conditions of industrial automation applications a test methodology has been developed by the authors (see Rauchhaupt, L., Krätzig, M., 2008.). It considers the application field, but is independent of a certain wireless technology. All relevant influencing parameters are taken into account (see Gnad, A., Krätzig, M., Rauchhaupt, L., Trikaliotis, S., 2008). These parameters are configured well-directed or if not possible documented. Here only the main important influencing parameters are described because of the limited space.

The focus lies on the impact of different bit rates on the performance and reliability of WLAN solutions. For WLAN systems according to IEEE 802.11 this is related implicitly to the physical layer coding and modulation schemes as discussed later on.

Although the WLAN implementation (hardware and software) has a noticeable influence on the time behaviour (see e.g. Rauchhaupt, L., Krätzig, M., 2008, Rauchhaupt, L., 2009) it is not considered here. However, an industrial WLAN solution was selected which provides the best performances in terms of jitter of the transmission time (i.e. span) and in terms of minimum outliers which means in number and in value.

Also interferences of other wireless systems are not considered. They are well investigated as described e.g. in ZVEI, 2009.

Thus, a simple point-to-point topology is used for the investigations consisting of a WLAN access point and one client. The attenuation between the two devices was 60 dB which is according to about 10 m line of sight. The WLAN channel 1 (2412 MHz) was chosen.

The test packets have been generated by the client. They had a user data length of 64 octets which is the minimum length for an Ethernet packet and a typical length in Ethernet based automation applications.

In previous investigations the dependency of the performance from the value of the transmit time interval became obvious. Therefore in these tests the client generated the test packets with a random transmit time interval between 15 ms and 25 ms. The random generation is based upon a uniform distribution function.

The sample size of each test case was 30,000 packet transmissions which resulted in test durations of about 10 minutes.

NOTE: With the chosen implementation a bit rate of 5.5 MBit/s was not adjustable. When this bit rate has been configured the WLAN system showed the same behaviour as for automatic bit rate control mode. Therefore, the bit rate 5.5 MBit/s is not considered here.

3 CHARACTERISTIC PARAMETERS AND TEST SYSTEM

The analysis of literature concerning the usage of characteristic parameters to describe and assess communication behaviour has shown that there are remarkable differences. Moreover, the definitions come mostly from the application field of Ethernet, Internet or telecommunication which does not fit to the application field of industrial automation (e.g. in (EN 61491, 1999), (EN 61209, 200)).

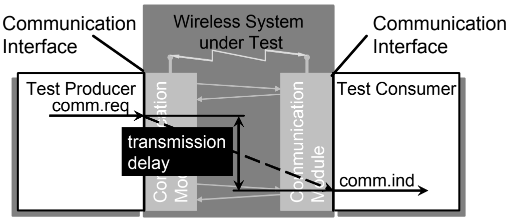


Figure 1: Definition of the transmission delay.

That is why it was necessary to find appropriate definitions. The following characteristic parameters are proposed to assess wireless communication systems with respect to industrial automation applications:

- Transmission time
- Update time
- Response time
- Data throughput
- Packet loss rate
- Residual error rate
- Activation time after energy saving mode
- Energy requirements

It has to be mentioned that it is not required nor recommended to use all parameters at the same time to characterise a communication solution for a certain application. The definitions of the listed characteristic parameters can be found in (VDI/VDE 2185, 2007).

In this paper we focus on the transmission time. It is related to event driven applications. For example when a work piece reaches a certain position in order to be machined or when a fluid reaches a defined level in a tank. In these cases it is of interest as to how long it takes to transfer the information from sensor to the control unit e.g. programmable control logic (PLC). The appropriate characteristic parameter of a communication system to assess its behaviour is the transmission time.

The definition of transmission time is based on a producer consumer model as shown in Figure 1. It is the time duration from the beginning of the handing over of the first user data byte of a packet at the communication interface in the test producer, up to the handing over of the last user data byte of the same packet at the communication interface at the test consumer. It may be necessary to transmit several telegrams between the communication modules e.g. for acknowledgment. Furthermore, network elements such as base stations may be involved in the communication producing additional delays. All these delays are covered by the transmission time.

In order to assess the reliability the retry rate is analysed. It is defined as the number of WLAN packets that are necessary to transmit the content of one generated test packet.

The system under test implements a widely used radio technology - Wireless LAN and its application interface is very common - Ethernet. Therefore, standard measurement equipment can be used to implement the above mentioned test methodology.

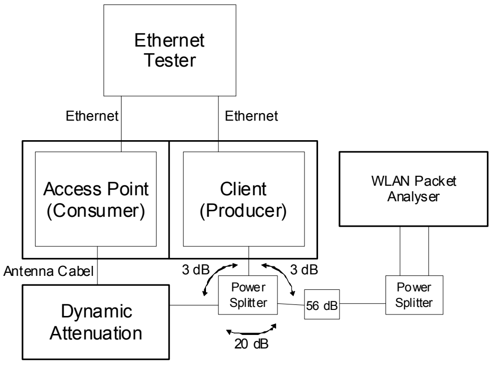


Figure 2: Test architecture.

The test architecture is shown in Figure 2. The devices under test (WLAN client and WLAN access point) are put into separate radio proof enclosures. The test boxes achieve an attenuation of 90 dB against the outer environment. The radio interfaces of access point and client are connected by an antenna cable. It is led via a power splitter and a dynamically changeable attenuator. This approach excludes interferences to the radio communication.

The dynamic attenuation has been configured to a constant value of 57 dB. Together with the attenuation of splitter and antenna cable the total attenuation is 60 dB as specified for the test cases. Besides the relevance of this value for automation applications it has been shown that in this way the transmit signal is low enough to avoid an overmodulation and high enough to be unaltered. The second output of the power splitter is connected to another static attenuation of 56 dB in order to connect a WLAN packet analyser. A second power splitter is used in order to provide the signals to two channels of the WLAN packet analyser. This increases the reliability of the packet monitoring. The total attenuation between client and WLAN analyser is 62 dB. The total attenuation between access point and analyser is more than 133 dB. Thus the analyser captures only the packets transmitted by the client.

The test packets are generated by an Ethernet tester which transfers the data to the WLAN client. As mentioned before the user data length is 64 octets. The generated packets are compliant to PROFINET-IO telegrams. The advantage is the specific frame type and content which simplifies the identification of packets during analysis.

The packets transferred by the WLAN client are monitored by WLAN analyser. If the packets are successfully received by the WLAN access point they are transferred to the Ethernet tester and are monitored there also.

During analysis in a first step the user data packets are filtered out of the packets monitored by the WLAN analyser. Since every user data packet can be identified by a unique payload it can be found within the user data packets monitored by the WLAN analyser. If a user data packet is listed more than ones it means retries has been initiated by the WLAN client because of missing acknowledgements. This way the retry rate can be calculated.

In addition a timestamp is included within the user data. With help of this timestamp the Ethernet tester is able to calculate the transmission time for every successfully received packet.

4 TEST RESULTS

4.1 Transmission Delay

The measured transmission time consists of random and constant components (see also Rauchhaupt, L., Krätzig, M., 2008.). Examples for constant components are the frame spacing times or the signal propagation delay on the wireless medium. These components have the same values in every measurement of a sample. The random nature of the transmission time is being caused by latency of application interface and implementation, by the technology variable, the number of packet retries and the time allocation for additional connections.

Therefore, stochastic methods are required in order to analyse the measurement results. Statistic parameters such as mean value, median, standard deviation, 95%-percentile (p95), maximum and minimum value of a test case can be analysed.

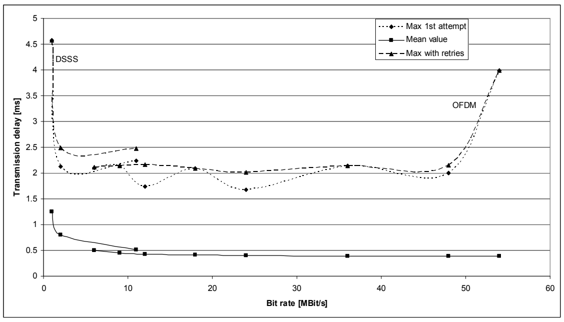


Figure 3: Transmission time for first attempt and including all retries.

In Figure 3 then mean value of the transmission time including all packets retries is depicted. Furthermore the maximum transmission time values out of all value which are received after the first attempt is shown as well as the maximum transmission time values of all successfully received packets.

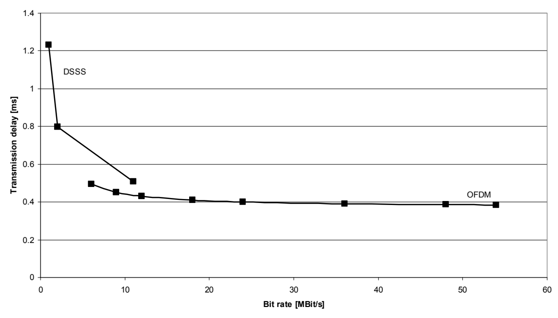


Figure 4: Delay for transmissions without retries.

The first result is that the maximum values for a certain bit rate are almost the same independent how often a packet is retransmitted. Since there is no contention on the medium it can be concluded that the maximum values of the transmission time are determined in the first line by the implementation including interfaces from and to Ethernet.

The second result is that the value of the outliers increases dramatically for 54 Mbit/s, while the mean value is similar to all other bit rates.

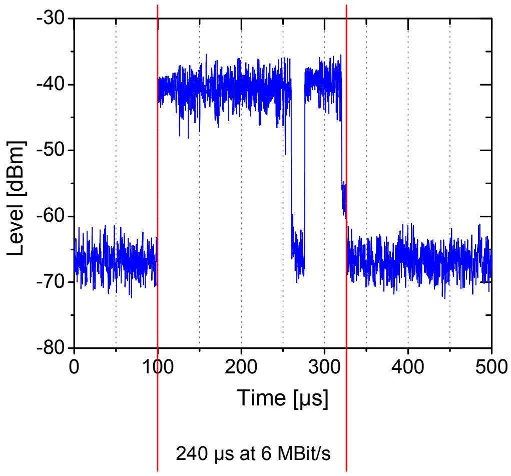


Figure 5: Medium utilisation time at 6 Mbit/s.

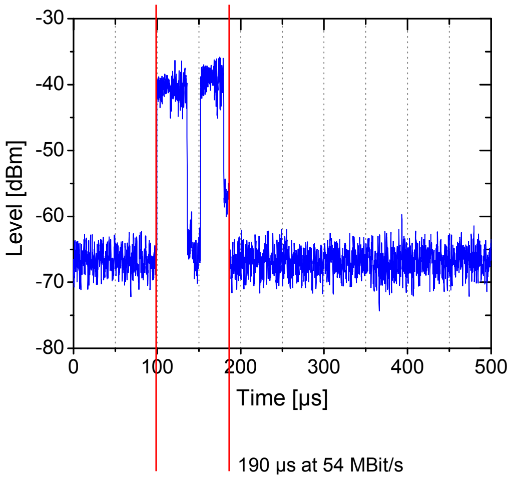


Figure 6: Medium utilisation time at 54 Mbit/s.

Since the influence of the number of retransmissions is not very significant in Figure 4 only the mean values of the transmission time are depicted for those packets which are received at the first transmission attempt.

The diagram shows that at low bit rates the differences between the neighbouring values are high. The differences become smaller with higher bit rates. However, from 12 Mbit/s on the increase of the mean values is very low. The reason for that behaviour is the time for a packet on the medium.

Figure 5and Figure 6 show the signal power level of a WLAN data packet and an acknowledge frame during transmission on the medium. It can be seen that the increase of the bit rate by 9 (from 6 Mbit/s to 54 Mbit/s) causes only a decrease of the medium utilisation time by about 1.3 (from 240 µs to 190 µs). The reason is that because of the backward

compatibility of the WLAN standard parts of a packet are transferred with 1 Mbit/s or 2 Mbit/s. With 64 octets user data only this aspect has remarkable influence on the transmission time.

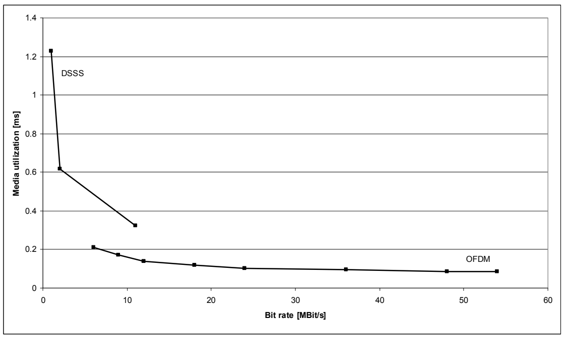


Figure 7: Media utilization time for different bit rates.

Figure 7 illustrates the medium utilisation time over the bit rate. As shown in Figure 5and Figure 6 the medium utilisation time is defined as time duration from the beginning of a data packet transmission up to the end of the related acknowledgement. Retransmissions are not considered. The curves in Figure 4 and Figure 7 show a similar characteristic. This underlines the influence of the medium utilisation time on the transmission time.

In addition the curves in Figure 7 indicate also another effect. The WLAN standard IEEE 802.11 specifies different coding and modulation schemes as listed in Table 1.

Table 1: Overview of the transmission method and modulation for each data rate.

Data rate	Method	Modulation
1 MBit/s	DSSS	DBPSK
2 MBit/s	DSSS	DQPSK
5.5 MBit/s	DSSS	DQPSK
6 MBit/s	OFDM	BPSK
9 MBit/s	OFDM	BPSK
11 MBit/s	DSSS	DQPSK
12 MBit/s	OFDM	QPSK
18 MBit/s	OFDM	QPSK
24 MBit/s	OFDM	16 QAM
36 MBit/s	OFDM	16 QAM
48 MBit/s	OFDM	64 QAM
54 MBit/s	OFDM	64 QAM

Theses schemes are related to the configurable bit rates. In particular the coding scheme Direct Sequence Spread Spectrum (DSSS) and Orthogonal Frequency Division Multiplex (OFDM) have different impacts on the medium utilisation time. Therefore, the curves are separated for theses coding schemes in Figure 3, Figure 4 and Figure 7.

4.2 Number of Retransmissions

The bar chart in Figure 8 depicts the number of packet transmissions needed for a correct reception by the access point. A logarithmic scale is chosen for the number of transmissions in order to see the small number of retransmissions. The interesting fact of this picture is that while using bit rates related to DSSS coding the maximum number of retransmission is one. Obviously this coding scheme is more robust than OFDM.

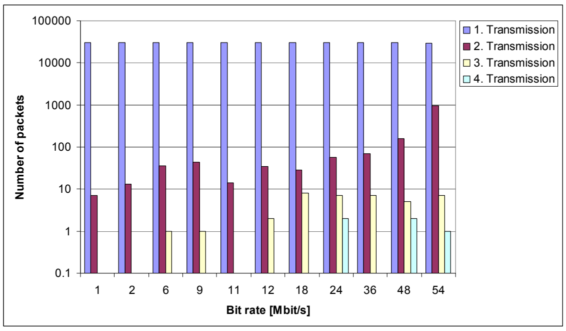


Figure 8: Number of packet retransmissions over bit rate.

For all bit rates related to DSSS the entire number one retransmission is about ten. In test cases with bit rates related to OFDM coding a number of packet transmissions with at least two retransmissions can be noticed. Especially for higher bit rates also up to four retransmissions can be seen.

The diagram in Figure 9 illustrates the total number of PROFINET-IO packet transmissions sent by the WLAN client via the medium. This means the transmission of all data packets including all retransmissions. Figure 9 shows almost constant values for DSSS bit rates. They are concentrated next to the lower limit of 30000 packets. The OFDM bit rates do not have such a constant behaviour. However, for low OFDM bit rates only little differences exist. At higher bit rates the number of packet transmission increases. Especially for 54 Mbit/s the number of required transmission is remarkable high. For this bit rate 31000 WLAN packets are required to transmit 30000 Ethernet packets. This results in a retry overhead of about 3.3 percent.

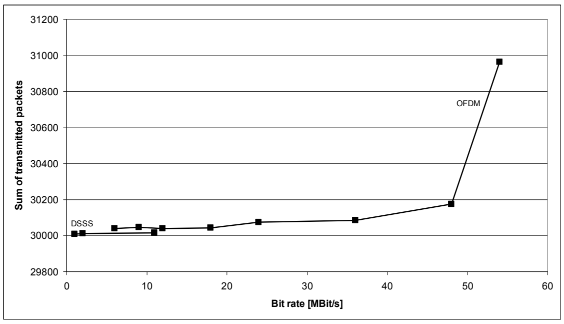


Figure 9: Total number of packet transmissions.

4.3 Loss Probability

Based on the discussion of retransmissions the loss probability shall be determined. Out of the results shown in Figure 8 a geometric distribution is assumed. The geometric distribution deals with the number of independent trials required for a single success. A famous example of this distribution is tossing a coin until it lands on heads.

The formula given in equation 1 describes the expectation value E(X) of required transmissions for each bit rate. Whereas N is the sample size, i is the indicator of the transmission attempt and k_i is the frequency of an attempt within a test case.

$$E(X) = \frac{1}{N} \cdot \sum_{i=1}^{n} k_i \cdot i \tag{1}$$

N in formula 1 has not been calculated as shown in equation 2 since the sample size is well known for the test cases.

$$N = \sum_{i=1}^{n} k_i \tag{2}$$

The estimation of the success probability p of a packet transmission is equivalent to the reciprocal of the expectation value E(X) of required transmissions (Formula 3).

$$\hat{p} = \frac{1}{E(X)} \tag{3}$$

With the well-known equation 4 it is easy to get the loss probability out of the success probability.

$$\hat{q} = 1 - \hat{p} \tag{4}$$

Figure 10 illustrates the loss probability for different bit rates, which has been calculated with the last-mentioned equations. In this picture we differentiate again between DSSS and OFDM method. The diagram is similar to the one of Figure 9. The only difference is the normalised ordinate. Additionally, the figure presents low loss probabilities for DSSS bit rates. On the other hand values for OFDM bit rates are essentially higher. However, for a wide range the values are almost constant. Furthermore, the abovementioned extreme value at 54 MBit/s exists also in this figure. In summary the diagram underlines the results concerning the robustness of DSSS coding scheme. OFDM bit rates are not recommended with respect to a robust communication.

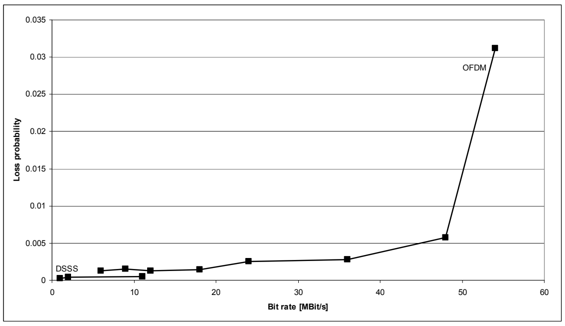


Figure 10: Loss probability for different data rates.

5 COST FUNCTION

In order to consider all the above mentioned aspects for the assessment of a wireless system a cost function has been developed. It is a trade of between (a high) transmission time and (a low) loss probability. Therefore the values of the transmission time measured in section 4.1 and the loss probability values calculated in section 4.3 are used. For further discussions the cost function shall depend on the bit rate. In addition the values should not have a dimension unit.

Therefore, the mean values of the transmission time for packets without retransmissions are used. Thus the influence of retransmissions on the time behaviour is avoided. In order to get values free of dimension units the transmission time values are put in a ratio to the shortest transmission time mean value. This value belongs to a bit rate of 54 MBit/s. Thus the value of bit rate 54 MBit/s is assigned to the ratio of one in the cost function. All other bit rates have values greater than one. The ratio for 1 MBit/s is the highest and is almost three times greater than for the reference bit rate.

The second part of the cost function is the loss probability. It is multiplied by the transmission time ratio. The complete formula is given in equation 5.

$$C(DR) = \frac{\overline{t_{TD1}(DR)}}{t_{TD1}(54MBit/s)} \cdot \hat{q}(DR) \qquad (5)$$

The cost for different bit rates is illustrated in the diagram of Figure 11.

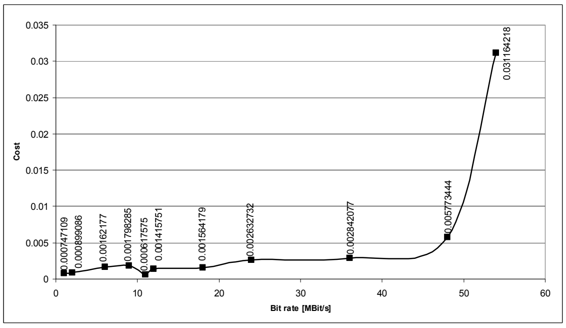


Figure 11: Cost function of WLAN packet transmissions.

The diagram shows a general trend. The cost values increase with higher bit rates. This means better communication behaviour can be expected with lower bit rates.

Taking a deeper look to the cost function the minimum is visible at 11 MBit/s. As in the figures before this diagram also offers the maximum at 54 MBit/s.

Bit rates lower than 24 MBit/s are almost in the same range of values. However, there is no constant increasing of cost values with higher bit rates. At bit rates greater or equal than 24 MBit/s the trend of the cost function rises noticeably.

6 CONCLUSIONS

The focus of this paper was the investigation of the influence of bit rates on the reliability and performance of WLAN systems with respect to industrial automation applications. Thus the test setup and the values of the influencing parameters reflect the communication requirements of this application field.

A first result of the measurements is the confirmation of the generally way of thinking. Higher bit rates cause shorter transmission time values. On the contrary lower bit rates cause less packet retransmissions. However, in the context of industrial automation a more detailed look at the results is necessary.

Thus there is an important finding, that there are differences between the coding schemes DSSS and OFDM. Obviously DSSS shows a more robust behaviour than OFDM. DSSS is included in IEEE 802.11 b. In contrast IEEE 802.11 g uses OFDM in favour. However, in order to be compatible with IEEE 802.11 b devices IEEE 802.11 g also supports bit rates using DSSS. With this background the user can choose DSSS for more reliable connections or OFDM for faster transmissions.

Another result is that the maximum values of the transmission time are more influenced by the implementation than by the number of retransmissions. Thus optimisation potential of the WLAN implementation can be detected using the presented test approach. In fact there are remarkable differences between industrial WLAN solutions as investigated by the authors in other tests projects.

However, also the retransmissions play an important role when a congested medium is taken into account what was excluded in the given investigations.

Therefore a cost function was introduced that opens the possibility to consider transmission time and required retransmissions for a successful communication. It can help to find the appropriate trade-off between a fast and a reliable wireless connection. As shown in Figure 11 a transmission with 11 MBit/s has the lowest cost value and is therewith the best trade-off with respect to considered application field. Also other bit rates of the IEEE 802.11 b specification, which uses DSSS and the corresponding modulations, showed an appropriate behaviour. The worst behaviour offer transmissions with 54 MBit/s. This fact disagrees with the popular opinion that a high bit rate is the best solution real-time automation applications. In addition the investigations show that even the medium utilisation with 54 Mbit/s is the highest because of the retransmissions. This is an important outcome with respect to the required efficient spectrum use in order to support coexistence between wireless systems.

Other interesting points of the explained measurements are the lower costs of 12 Mbit/s and 18 MBit/s in comparison to 6 Mbit/s and 9 Mbit/s. The rates of 24 Mbit/s and 36 MBit/s are also acceptable.

As a conclusion the best bit rates for industrial automation applications are the ones of the IEEE 802.11 b specification. With the focus on fast transmissions bit rates between 12 Mbit/s and 36 MBit/s could be chosen. The bit rates 6 Mbit/s, 9 Mbit/s, 48 Mbit/s and 54 MBit/s should be avoided.

REFERENCES

EN 61491, 1999. *Electrical equipment of industrial machines– Serial data link for real-time communication between controls and drives*

EN 61209, 2000. *Maritime navigation and radiocommunication equipment and systems-Integrated bridge systems (IBS)- Operational and performance requirements, methods of testing and required test results*

VDI/VDE-Guideline 2185, 2007. *Radio based communication in industrial automation*

Dzung, D., Apneseth, C., Endresen, J., Frey, J.-E., 2005. *Design and Implementation of a Real-Time Wireless Sensor/Actuator Communication System*, 10.1109/ ETFA.2005.1612710, Volume: 2, pp. 10 – 19, ISBN: 0-7803-9401-1

Siemens, 2007. *Industrial mobile communication*, Siemens AG, Order No. 6ZB5530-1AM02-0BA2

Rauchhaupt, L., Krätzig, M., 2008. *How to Assess Reliability of Industrial Wireless Solutions*, 5th International Conference on Informatics in Control, Automation and Robotics (IFAC), 11-15 May, 2008, Funchal, Madeira, In: Proceeding , Volume 1 Robotics and Automation, Pages 122-130, ISBN: 978-989-8111-31-9, CD

Rauchhaupt, L., 2009. *Radio Technology for Real-Time Application - The flexWARE Project*, ETSI Wireless Factory Starter Group Meeting N° 1, ETSI, Sophia Antipolis, France, 20 - 21 October 2009, In: http://portal.etsi.org/docbox/Workshop/2009/200910_WIFA [access 24.10.2009]

Gnad, A., Krätzig, M., Rauchhaupt, L., Trikaliotis, S., 2008. *Relevant Influences in Wireless Automation*, 7th IEEE International Workshop on Factory Communication Systems, COMMUNICATION in AUTOMATION (WFCS 2008), May 20-23, 2008, Dresden, Germany, ISBN: 978-1-4244-2350-7, CD

ZVEI, 2009. *Coexistence of Wireless Systems in Automation Technology - Explanations on reliable parallel operation of wireless radio solutions.* Frankfurt, Germany, 2009, https://www.zvei.org/fileadmin/user_upload/Fachverbaende/Automation/Publikation/ZVEI_Coexistence_of_Wireless_Systems_in_Automation_Technology.pdf, [access 01.02.2010]

Trsek, H., 2008. *Reproduzierbare Leistungsmessung und Konformitäts/Interoperabilitätstests von IEEE 802.11 WLAN Implementierungen*. 10th Wireless Technologies Congress, Bochum, Germany

SIMULTANEOUS LEARNING OF PERCEPTIONS AND ACTIONS IN AUTONOMOUS ROBOTS

Pablo Quintía, Roberto Iglesias, Miguel Rodríguez
Department of Electronics and Computing, Universidade de Santiago de Compostela, Santiago de Compostela, Spain
{pablo.quintia, roberto.iglesias, miguel.rodriguez}@usc.es

Carlos V. Regueiro
Department of Electronics and Systems, Universidade da Coruña, A Coruña, Spain
cvregueiro@udc.es

Keywords: Reinforcement learning, State representation, Autonomous robots, Fuzzy ART.

Abstract: This paper presents a new learning approach for autonomous robots. Our system will learn simultaneously the perception – the set of states relevant to the task – and the action to execute on each state for the task-robot-environment triad. The objective is to solve two problems that are found when learning new tasks with robots: interpretability of the learning process and number of parameters; and the complex design of the state space. The former was solved using a new reinforcement learning algorithm that tries to maximize the *time before failure* in order to obtain a control policy suitable to the desired behavior. The state representation will be created dynamically, starting with an empty state space and adding new states as the robot finds them, this makes unnecessary the creation of a predefined state representation, which is a tedious task.

1 INTRODUCTION

Robots must be able to adapt its behaviour to changes in the environment if we want them operating in real scenarios, dynamic environments or human's common workplaces. Because of this in this paper we describe a model free learning algorithm, able to adapt the behaviour of the robot to new situations and that not relies on any predefined knowledge. Our system will learn simultaneously how to translate the perceptions of the robot into a finite state space and the actions to perform at each state to achieve a desired behaviour. We are not aware of any other publications with the same objectives.

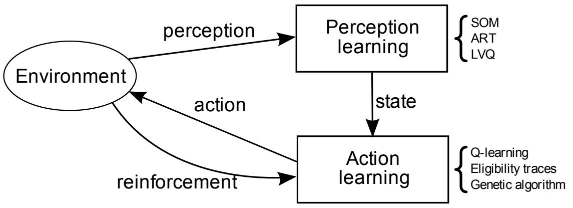


Figure 1: General schema of our proposal with the two modules for perception learning and action learning.

We propose the combination of reinforcement learning (Sutton and Barto, 1998) to learn the actions and a Fuzzy ART network (Carpenter et al., 1991) to learn the states (Fig. 1). Our system will not only learn the actions to execute on each state but also it will learn to classify the situations the robot finds during its operation. Our reinforcement learning based algorithm will be simpler and easier to interpret than other approaches, and the dynamic representation of states will create the state space from an empty set of states. This eliminates the burden of creating an adhoc representation for each task. Thanks to this combination of reinforcement learning and Fuzzy ART we will achieve a technique able to learn on-line, adapting the behaviour of the robot to the changes that may occur in the environment or in the robot itself.

2 ACTION LEARNING

Sutton and Barto developed reinforcement learning as a machine learning paradigm that determines how an agent ought to take actions in an environment so as to maximise some notion of long-term reward (Sutton and Barto, 1998).

Reinforcement learning is a very interesting strategy, since all the robot needs for learning a behaviour is a reinforcement function which tells the robot how good or bad it has performed, but nothing about the set of actions it should have carried out. Through a stochastically exploration of the environment, the robot must find a control policy – the action to be executed on each state – which maximises the expected total reinforcement it will receive:

$$E[\sum_{t=0}^{\infty} \gamma^t r_t] \tag{1}$$

where r_t is the reinforcement received at time t, and $\gamma \in [0,1]$ is a discount factor which adjusts the relative significance of long-term versus short-term rewards.

Q-learning (Watkins, 1989) is one of the most popular reinforcement learning algorithms, although it might be slow when rewards occur infrequently. What is termed Eligibility Traces (Watkins, 1989) expedite the learning by adding more *memory* into the system. One problem of this algorithms is their dependence of the parameters used, that usually need to be set after a trial an error process.

In this work we present a new learning algorithm based on reinforcement. Our algorithm will provide a prediction of how long the robot will be able to move before it makes a mistake. This raises clear and readable systems where it is easy to detect, for example, when the learning is not evolving properly: basically a high discrepancy between the time before failure predicted and what is actually observed on the real robot. Another advantage of our learning proposal is that it is almost parameterless, so it minimises the adjustments needed when the robot operates in a different environment or performs a different task. The only parameter needed is a learning rate which is not only easy to set, but it is often the same value, regardless of the task to be learnt.

Since we wish to use the experience of each state transition to improve the robot control policy in real time, we shall apply Q-learning, but redefining the utility function of states and actions. Q(s,a) will be the expected time interval before a robot failure when the robot starts moving in s, performs action a, and follows the best possible control policy thereafter:

$$Q(s,a) = E[-e^{(-Tbf(s_0=s,a_0=a)/50T)}], \tag{2}$$

where $Tbf(s_0,a_0)$ represents the expected time interval (in seconds) before the robot does something wrong, when it executes a in s, and then follows the best possible control policy. T is the control period of the robot (expressed in seconds). The term $-e^{-Tbf/50T}$ in Eq. 2 is a continuous function that takes values in the interval $[-1,0]$, and varies smoothly as the expected time before failure increases.

Since $Q(s,a)$ and $Tbf(s,a)$ are not known, we can only refer to their current estimations $Q_t(s,a)$ and $Tbf_t(s,a)$:

$$Tbf_t(s,a) = -50 * T * Ln(-Q_t(s,a)), \tag{3}$$

The definition of $Q(s,a)$, Tbf, and the best possible control policy, determine the relationship between the Q-values corresponding to consecutive states:

$$Tbf_t(s_t,a_t) = \begin{cases} T & \text{if } r_t < 0 \\ T + max_a\{Tbf_t(s_{t+1},a)\} & \text{otherwise} \end{cases} \tag{4}$$

r_t is the reinforcement the robot receives when it executes action a_t in state s_t. If we combine Eq. 3 and Eq. 4, it is true to say:

$$Q_{t+1}(s,a) = \begin{cases} -e^{-1/50} & \text{if } r_t < 0 \\ Q_t(s_t,a_t) + \delta & \text{otherwise} \end{cases} \tag{5}$$

where,

$$\delta = \beta(e^{\frac{-1}{50}} * max_a Q_t(s_{t+1},a) - Q_t(s_t,a_t)). \tag{6}$$

$\beta \in [0,1]$ is a learning rate, and it is the only parameter whose value has to be set by the user.

3 PERCEPTION LEARNING

In reinforcement learning the state space definition is a key factor to achieve good learning times. The state space must be fine enough to distinguish the different situations the robot might find, but at the same time it must have a reduced size to avoid the curse of dimensionality.

The design of the state space is a delicate task, and it is dependent on the problem the robot has to solve. We propose a dynamic creation of the state space as the robot explores the environment (Fig. 1). For this task we have chosen to use a Fuzzy ART artificial neural network (Carpenter et al., 1991). This kind of networks are able to perform an unsupervised online classification of the input patterns without any previous knowledge.

Of the three parametres that are involved in the Fuzzy ART algorithm α, β and ρ – usually called vigilance parameter – the most important is ρ. α and β are almost independent of the task to solve, but the value of ρ will influence the number of states created. If it is too high the Fuzzy ART will create too many classes. If ρ is too low the state representation will be too coarse and the system will suffer from perceptual aliasing, resulting in an increase of the learning time or impossibility to achieve convergence.

Due to space restrictions we can't provide more details of the Fuzzy ART here. Nevertheless further information can be found in (Carpenter et al., 1991).

4 EXPERIMENTAL RESULTS

The two systems showed in Fig. 1 complement each other to find a solution to the learning problem. We will perform several experiments:

a) Evaluate the performance of our learning algorithm described in 2. To do this without the influence of the Fuzzy ART network we used a set of two-layered SOM networks to translate the large number of different situations that the ultrasound sensors may detect, into a finite set of 220 neurones – states (Iglesias et al., 1998).

b) Evaluate the performance of the Fuzzy ART network creating a state space from scratch, using both normalised and not normalised inputs using complement coding (CC) (Carpenter et al., 1991).

We applied our proposal to teach a mobile robot two different tasks: a wall following task; and a door traversal task. The inputs for the Fuzzy ART network will be the inverted readings provided by a laser rangefinder. We reduced the dimensionality to 8 sectors of laser readings 22.5°wide, using the lowest measure as representative of each sector.

The parameters of the learning algorithms used during the learning were: $\beta = 0.288282, \gamma = 0.9, \lambda = 0.869965$. The parameters of the Fuzzy ART were: $\alpha = 0.00001, \beta = 0.0025$.

4.1 Wall Following

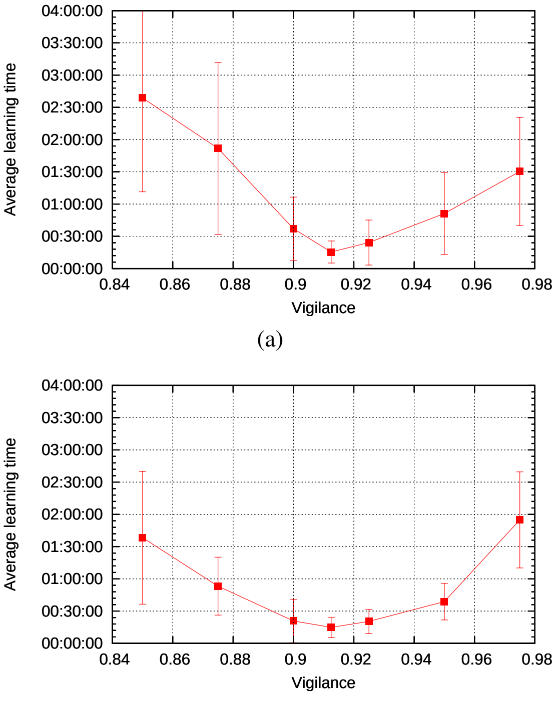
As said before we will use a static state representation to test the learning algorithms. In order to train the SOM neural networks we used a set of sensor readings collected when the robot was moved close to a wall (Iglesias et al., 1998). For comparison purposes we tested our learning approach against three classical algorithms: Q-learning and two different implementation of eligibility traces: Watkins' Q(λ) (Watkins, 1989) and what is called Naive Q(λ) (Sutton and Barto, 1998). The results obtained after the execution of 15 experiments for each algorithm can be seen in Table 1. The classical learning algorithms performed as expected. Our proposal based on learning the time before failure performed as good as Naive Q(λ). The main advantage of our learning algorithm is to have a more interpretable and simple algorithm, with almost no cost on the learning time.

The next step in the experimentation was to combine the Fuzzy ART with the learning algorithm. Considering the previous results, we chose to test the combination of the Naive Q(λ) algorithm and Fuzzy ART. In Fig. 2(a) we can see the variations in the average learning time and std. deviation with different values of the vigilance parameter – ρ . From this results we can extract that the valid range – learning times lower than 1 hour – for the vigilance is approximately [0.900, 0.950] and that the best values are around 0.9125. Fig. 2(b) shows the results of the experiments if the inputs of the Fuzzy ART are codified in complement coding. We can see that the use of complement coding does not reduce significantly the learning time achieved by the optimal vigilance value, but it does improve the learning times if the vigilance parameter is not the optimal.

The best value for the vigilance parameter found in this experiments – 0.9125 – can serve as a good starting point for the use of the Fuzzy ART in other tasks. This value will be used to learn other tasks.

Table 1: Results of the learning of a wall following task with a predefined SOM network (Iglesias et al., 1998).

Algorithm	Learning time	Std. deviation
Q-learning	00:29:37	00:13:59
Watkins's Q(λ)	00:21:35	00:12:32
Naive Q(λ)	00:17:21	00:08:21
Our proposal	00:16:39	00:08:14



(a)

(b)

Figure 2: Results of the wall following task with Naive Q(λ) and Fuzzy ART network without (a) and with (b) complement coding.

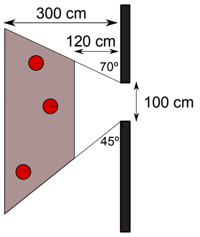


Figure 3: Experimental scenario for the door traversal task. The initial positions of the robot were within the shaded area.

Table 2: Results of the learning of a door traversal task with Naive Q(λ) and Fuzzy ART.

Vigilance	Beams	CC	Average learning time	Average deviation	Average states
SOM	8		01:36:53	00:57:55	221
0.9125	8	No	NA	NA	63.22
		Yes	01:37:12	01:00:55	94.87
	181	No	01:57:48	01:05:31	134.33
		Yes	00:39:56	00:23:20	82.87

4.2 Door Traversal

The door traversal task (Nehmzow et al., 2006) was learnt in the experimental scenario shown in Fig. 3. First we tested the system using the same SOM neural network that was used to learn the wall following task. The robot achieves a good control policy after a learning time of 01:36:53 with high variability – 00:57:55. Using our system the results are equivalent if complement coded is used, without complement coding the system is unable to learn in reasonable time (Table 2).

With the first experiments we found out that if we use the same input as in the wall following task the door was not visible from several positions. To have a better perception of the door we decided to use all 181 laser readings. This improves the times significantly, the average learning time is reduced to 00:39:56 and the std. deviation lowers to 00:23:20. As can be seen in Table 2, the dynamic representation scales very well with the increase in the dimensionality. Complement coding is the appropriate choice for the Fuzzy ART inputs.

5 CONCLUSIONS

Through reinforcement learning the robot is able to learn on its own – through trial an error interactions with the environment – using only the feedback provided by a very simple reinforcement function. The learning algorithm developed in this paper represents a simpler and more interpretable solution to the learning problem. The algorithm requires less parameters and its meaning is more straightforward – the expected time before the robot commits an error.

But one of the main problems of applying reinforcement learning in robotics is the state space definition. In this paper we showed how we can use a Fuzzy ART neural network to dynamically create the state space while the reinforcement learning algorithm learns the actions to execute on each state.

The use of a dynamic representation of states does not suppose an increase in the learning time, in fact it reduces the learning time in comparison to the use of a predefined and static state representation if a good vigilance value is chosen. We also proved that this dynamic state representation scales well with size of inputs. But the main advantage of this approach is that there is no need to create an ad-hoc state representation for the task. Creating a predefined state representation requires gathering a training and test data set, training the network and validating the network. This must be repeated until we obtain a good network for our purpose.

Our proposal was used to solve two different and common tasks in mobile robotics: wall following and door traversal. The experimental results confirm that our proposal is valid.

ACKNOWLEDGEMENTS

This work has been funded by the research grants TIN2009-07737, INCITE08PXIB262202PR, and TIN2008-04008/TSI.

REFERENCES

Carpenter, G. A., Grossberg, S., and Rosen, D. B. (1991). Fuzzy art: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks*, 4(6):759–771.

Iglesias, R., Regueiro, C. V., Correa, J., Sánchez, E., and Barro, S. (1998). Improving wall following behaviour in a mobile robot using reinforcement learning. In *ICSC International symposium on engineering of intelligent systems (EIS'98)*, Tenerife (España).

Nehmzow, U., Iglesias, R., Kyriacou, T., and Billings, S. (2006). Robot learning through task identification. *Robotics and Autonomous Systems*, 54:766–778.

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement learning: An introduction.* MIT Press.

Watkins, C. (1989). *Learning from Delayed Rewards.* PhD thesis, University of Cambridge, England.

GEOMETRIC FORMATIONS FOR A TEAM OF MOBILE ROBOTS
Odometric-based Maintenance Method for Heterogeneous Teams of Robots

Patricio Nebot and Enric Cervera
Robotic Intelligence Lab, Universitat Jaume I, E-12071 Castelló de la Plana, Spain
{*pnebot, ecervera*}*@uji.es*

Keywords: Maintenance of formations, Heterogeneous team of mobile robots, Cooperation among robots.

Abstract: One of the most important topics in multirobot formations is how to maintain the initial formations while the robots are moving or navigating through the environment. This paper presents a new approach based on the cooperation among a team of heterogeneous robots for the maintenance of multirobot formations. The robots must cooperate among them in order to get that all the robots, despite their sensory power, can maintain the formation while they are moving. To get that, the robots only have available the communication among them and their own odometry information. One of the robots, the conductor, is in charge to drive the formation and the rest of robots must follow it maintaining the formation. To do that, the use of “virtual points” and Bezier curves are introduced.

1 INTRODUCTION

A considerable amount of research has been focused on the problem of coordinated motion and cooperative control of multiple autonomous robots. Researchers have been trying to understand how a group of moving robots can perform collective tasks. In fact, this is one of the most popular fields of study in multirobot systems in the last years, with some applications that are well understood including multirobot path planning (Yamashita et al., 2000), formation generation (Arai et al., 1989), and formation keeping (Balch and Arkin, 1998).

Coordination among a group of robots can be very useful for many applications. One of the most important tasks is, in motion coordination, how to move a team of robots in an ordered way, such as maintaining a predefined formation. One of the first approximations to multirobot formations is the leader-follower approach (Das et al., 2002; Liu et al., 2005) where one robot is selected as the leader and must be followed by the rest of robots.

In this field, two works (Chiem and Cervera, 2004; Renaud et al., 2004) were developed at the Robotic Intelligence Lab in Castellón. These works were useful as the basis for the work explained in this paper.

The application described in this paper is a continuation of the previous ones but extended to multiple types of formations by using virtual points. The main idea is to make it feasible for a heterogeneous team of four robots to navigate through an environment in such a way that the robots with sensory power help the robots without it, that is, cooperating among them.

The control for the maintenance of the formation is performed using a decentralized process, where each follower robot decides which movements must be performed in order to follow the movements of the conductor robot, which is specialized in navigation because it is using a laser range-finder and is ahead in the formation. Its actual position while it is moving is sent to every follower robot. When receiving the position, each follower robot calculates the virtual point that it must follow. This virtual point consists of a displacement in the position of the conductor robot that allows it to create different formations, not only the leader-follower line formation.

Once the virtual point is calculated, the follower robot computes the trajectory it must follow in order to arrive from the current position to the estimated conductor robot relative position. For the calculation of this trajectory, Bezier curves are considered, as in (Chiem and Cervera, 2004; Renaud et al., 2004).

2 MAINTENANCE OF THE FORMATION

In this section it is described how a formation can be maintained while the robots are moving in the environment. The formation is composed of a maximum of four robots, where one of them, the conductor, is equipped with a laser range-finder that allows it to navigate in buildings and follow a path determined from the map of the building, the goal point, and the obstacles. The rest of robots have limited visibility, that is, they only know their actual position.

Each robot determines its position in order to maintain the formation with respect to the leader or a displacement of the position of the leader, with a “virtual” point. Also, they use a relative frame of reference, that is, they do not use an absolute positioning system, only the robot with the laser uses an absolute system in relation with the map that it uses to navigate. And in relation with the communication capabilities, the robots can share local and global information to compensate for their limited visibility, in fact, the global information shared among the robots is the actual position of the robot with the laser, and the local information is the actual position of each robot.

In the formation, the position of the robot followers can be controlled by the position of the conductor if all the robots are arranged in a line formation. In other cases, when the followers are positioned to the right or left of the conductor, virtual points are added to the system, as explained in (Chiem and Cervera, 2004). These virtual points are calculated by applying a displacement in the conductor position to the left or to the right, depending on the desired formation. The followers, in this case, instead of following the conductor must follow these virtual points. This arrangement is shown in Figure 1.

In the approach here proposed to achieve the maintenance of the formation, it is desired that robots could perform the task using only the odometry in conjunction with communication to estimate the position of the conductor with respect to each follower. The conductor, by means of the localization task is always correctly localized in the map, and it is assumed that it will follow a predefined trajectory, with a constant, known linear velocity. In order for another robot to follow the conductor, the linear and angular velocities need to be computed at each time step. It must be noted that the linear velocity of the follower robots is not constant, due to the different radius of their respective trajectories or because their position may be relatively further back or further forward in the formation.

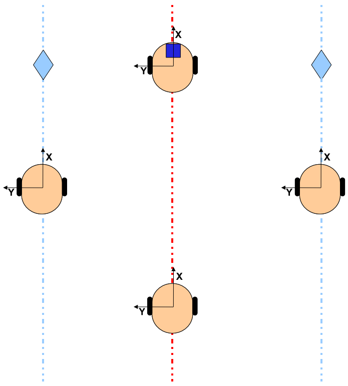


Figure 1: Formation using virtual points as reference for followers.

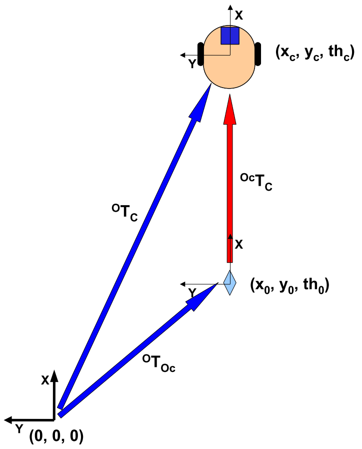


Figure 2: Calculation of the local position of the conductor.

In order to calculate the linear and angular velocities that allow the followers to move following the movement of the conductor and maintaining the formation, it is necessary to construct the corresponding Bezier curve that defines the trajectory to be followed by the robot. To compute the Bezier curve two positions are needed, the current position of the follower and the current position of the conductor in relation to the follower.

The conductor, at each step sends its own position to the followers. The conductor is always localized in the map by means of the localization task, so, when it requires information about its position, this is done with the coordinates of the map. The followers are not localized on the map, so the only information they have available is their position on their own local system which is determined by the origin that is fixed by their initial position. So, the conductor must transform their global position into a position in its local system. In Figure 2, the necessary relationships to transform the global position into a local position can be seen.

From the figure, it can be deducted that to calculate the position in the local system it is necessary to know the origin of the trajectory in the global system and the actual position of the robot in the global system.

$$^{Oc}T_C = \left(^{O}T_{Oc}\right)^{-1} \cdot {}^{O}T_C \tag{1}$$

where $^{Oc}T_C$ is the actual position of the conductor in the local system, $^{O}T_{Oc}$ is the origin of the trajectory in the global system, and $^{O}T_C$ is the actual position of the conductor in the global system. This last position is updated continuously by the localization task so that the robot have always its current position on the map.

The conductor sends its current position calculated in this way to each follower. When the followers receive the position of the conductor, they need to calculate the position of the conductor in relation to themselves in order to use that information to generate the Bezier curve necessary to calculate the velocities which allow them to move maintaining the formation.

Each follower knows its original displacement in the formation. This information in conjunction with the current position of the follower and the current position of the conductor allows the robots to calculate the relative position of the conductor in relation with the follower. In Figure 3, the necessary relationships to compute the conductor position are shown.

In this case, from the figure, the following equation to calculate the conductor position in relation with the follower $^{F}T_C$ can be obtained,

$$^{F}T_C = \left(^{Of}T_F\right)^{-1} \cdot {}^{Of}T_{Oc} \cdot {}^{Oc}T_C \tag{2}$$

where $^{Of}T_F$ is the position of the follower in its local system, $^{Of}T_{Oc}$ is the position of the origin of the conductor's trajectory in relation to the origin of the follower's trajectory. This value is indicated by the displacements assigned to the position of each robot in the creation of the formation. And finally, $^{Oc}T_C$ is the position of the conductor in its local system. This is the position that the conductor sends every step to the followers.

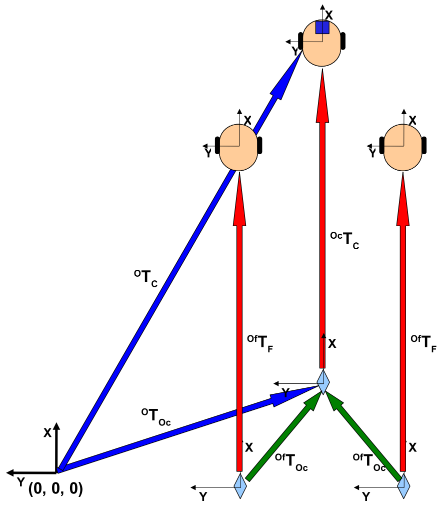


Figure 3: Calculation of the conductor's position in relation to the followers.

Once this position is calculated, it is possible to compute the Bezier curve between the current position of the follower and the position calculated for the conductor. It is at this moment when the virtual points are calculated, if applicable, applying the following formulas,

$$\begin{aligned} x &= x - \sin(\delta) \cdot dy \\ y &= y + \cos(\delta) \cdot dy \end{aligned} \tag{3}$$

where (x, y, δ) is the position of the conductor in relation to the follower, and dy is the displacement in the y axis to be applied. This generates a new point, the virtual point, which will be used to compute the Bezier curve that drives the movement of the follower.

From the Bezier curve computed, the linear and angular velocities can be obtained. The angular velocity can be computed from the curvature of the curve, and the linear velocity is computed in order to maintain the distance from the conductor, applying a gain factor proportional to the current distance from the conductor robot. If the robots is nearer than a predefined distance in the formation, it will move slower, and if the robot is further than the predefined distance, it will move faster.

3 TESTING AND RESULTS

The approach presented in this paper has been tested simulating a team of robots consisting of four Pioneer-2 mobile robots with different features mounted on them, constituting therefore a heterogeneous group. In particular, only one robot is equipped with a laser range-finder sensor and the rest of robots do not have any type of exteroceptive sensors.

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 Figure 4, some examples executed for different formations shapes are presented together with the formation in which the robots are organized. Formations from two robots up to four robots can be used.

As it can be seen, the odometry-based approach is suitable for the development of applications using multirobot formations. From the examples, it can be infered that the behaviour of the robots is as desired,

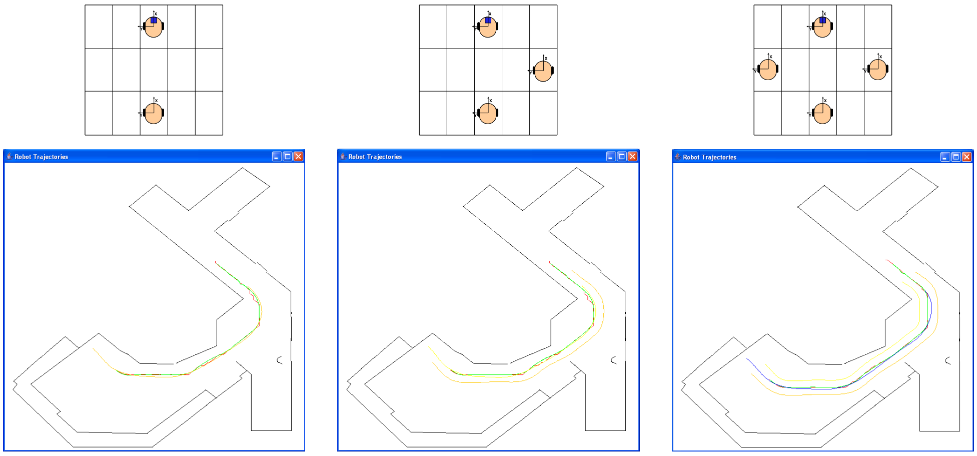


Figure 4: Results in simulation for formations with different number of robots.

and all the follower can follow the movements of the conductor by the utilization of virtual points.

4 CONCLUSIONS

The paper describes a new method for a team of heterogenous robots to navigate maintaining any type of formation by the introduction of "virtual points". In this application, the robots without sensory power get support for the navigation from the robot with navigation and localization facilities. In this way, all the robots are able to follow a predefined path getting help from others if necessary, that is, cooperating among them.

Results in simulation have demonstrated that the presented approach, where robots maintain the formation only using communication and odometry, is suitable for getting the maintenance of formations with any number of robots.

ACKNOWLEDGEMENTS

Support for this research is provided by the Generalitat Valenciana under project GV/2010/087, and the Fundació Caixa Castelló - Bancaixa under project P1-1A2008-12.

REFERENCES

Arai, T., Ogata, H., and Suzuki, T. (1989). Collision avoidance among multiple robots using virtual impedance. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 479–485.

Balch, T. and Arkin, R. (1998). Behaviour-based formation control for multiagent robot teams. *IEEE Transaction on Robotics and Automation*, 14(6):926–993.

Chiem, S. and Cervera, E. (2004). Vision-based robot formations with bzier trajectories. In *Proceeedings of the 8th Conference on Intelligent Autonomous System*.

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.

Liu, S., Tan, D., and Liu, G. (2005). Distributed formation control of robots with directive visual measurement. In *Proceedings of the IEEE International Conference on Mechatronics & Automation*, pages 1760–1765.

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.

Yamashita, A., Fukuchi, M., and Ota, J. (2000). Motion planning for cooperative transportation of a large object by multiple mobile robots in a 3d environment. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3144–3151.

PRIORITY SELECTION FOR MULTI-ROBOTS

S. H. Ji, S. M. Lee and W. H. Shon
Korea Institute of Industrial Technology, Sa-1-dong, Sangrok-gu, Ansan-si, KyungKi-do, South Korea
{robot91, lsm, shon}@kitech.re.kr

Keywords: Collision Model, Multi-robots, Mobile-robot Motion Planning, Priority Selection.

Abstract: The priority selecting problem for multi-robots deals with the determination of the relative importance of multi-robots with the limited capability in speed and acceleration in order that the robots should arrive in the minimum time without any collision. Unlike the case of a single robot, the arrival time of multi-robots depends on the delayed action for collision-avoidance. The delayed action time to avoid collision varies according to the priority order of the robots. This means that faster motion completion can be achieved by altering the distribution of the priority. However, the priority decision which provides the collision-free optimum operation of multi-robots cannot be solved mathematically. It is because the collision avoidance process among robots is closely linked mutually. Therefore in this paper, based on (M,D) network model, in considering the priority decision, how to reduce the complexity of priority decision is suggested by selecting the robots which have influence on operating performance of total robots. Conclusively, the effectiveness of the proposed approach is confirmed through simulation.

1 INTRODUCTION

Multi-robot motion planning is one of the essential research fields in robotics and has been studied for the last several decades. Multi-robot motion planning, however, is still a challenging field of research, having some technical difficulties in resolving conflict among agents. Especially, the centralized approaches to multi-robot motion planning have been faced with problems such as the curse of dimensionality, complexity, computational difficulty, and NP-hard problem (Canny, 1988) (Akella and Hutchinson, 2002). These problems are due to the fact that one system alone takes up the whole burden for planning motions of all agents interactively.

To overcome these problems in the approach, some researcher presented computational approach based on the systematic tools. The multi-robot motion planning problem is converted to a job scheduling problem or a constraint satisfaction problem (Barberl, et. al., 2001) (Chen, et. al., 2001). The converted problem is solved by project management tools or priority based tools. We proposed the extended collision map method which enables more than three robots 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 robot (Ji, et. al., 2007).

Unlike the case of a single robot, the arrival time of multi-robots depends on the delayed action for collision-avoidance. The delayed action time to avoid collision varies according to the priority order of the robots. This means that faster motion completion can be achieved by altering the distribution of the priority. Therefore, this priority order is a very important design factor of multi-robot system. But a few researches have attempted to the problem. Moreover, the researches were based on the static path information of robots without considering their mobility (Bennewiz, et. al., 2001). So, the type of approaches failed to give a reasonable solution to the priority selection problem when they were applied to a number of robots.

Therefore, in this paper, we suggest the way how to select the priority order for collision-free multi-robot operation considering static path information and dynamic mobility of multi-robots. For the purpose of our aim, analysis on characteristics of collision among agents is needed. Thus, in this paper, we use (M,D) network model based upon collision features which can express not only the complicated mutual interference among more than two robots but also help us design the collision-free operation of multi-robots and figure out the operating completion time of robots (Ji, et. al., 2009). And in this paper, based on (M,D) network model, in considering the priority decision, how to reduce the complexity of priority decision is suggested by selecting the robots which have influence on operating performance of total robots.

The remainder of the paper is organized as follows: Section 2 briefly describes multi-robots. Section 3 defines our priority selection problem in a mathematical form and Section 4 presents the concept of the key technique of this paper. Section 5 provides an implementation for a number of robots in order to verify the effectiveness of the proposed approach and finally this paper is concluded in Section 6.

2 MULTI-ROBOT SYSTEM

2.1 Assumptions

To reduce complexity of multi-robot motion planning, the extended collision map method applies several concepts as follows:

[Global Off-Line Path Planner]
Global off-line path planner (Central planner) can give the safe paths to all robots. In this paper, 'safe path' is the meaning that no robot will not crossover any other robot's starting point or destination if it keeping on its own safe path. Therefore there can be intersection points among robots' paths.

[Agent Model]
Robots in robotics can have either car-like or human-like shapes. Computation load can be reduced by using a simple model of a robot, so we model an agent as a circle. It is expressed by radius r and a center point in Eq.(1).

$$p(t) = [p_x(t), p_y(t)] \tag{1}$$

A radius of a robot is defined as an extended value of real shape considering the robot path's radius of curve and maximum path deviation error for stability. In real applications, there are sensor noise and jittering in sampling control interval. And there are effect on motion accuracy of slip and slope.

[Motion Characteristics of Agent]
An agent has physical constraints-velocity and acceleration limitation denoted by Eq. (2).

$$|\dot{p}(t)| \le v_{\max} \ , |\ddot{p}(t)| \le a_{\max} \tag{2}$$

A robot was assumed to always move in full speed within physical constraints and have a trapezoid model of the velocity profile. This assumption simplifies motion planning, because the central planner only has to consider the speed down for collision avoidance. All of the collision-free strategies in extended collision map are based on this assumption.

2.2 Collision-model

We suggested the collision model which express collision relations and predict possibility of collisions among the robots (Ji, et. al., 2009). And all of the robot's minimum delayed departure time for collision-free navigation can be extracted from the model. The elements of collision model are defined in Table 1.

Table 1: 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 i^{th} highest priority} 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

Now, we express the collision model from the case in Fig. 1 as the network model shown in Fig. 2. There are three robots (agent 1, 2, and 3) with path shapes as shown in Fig.1. 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.

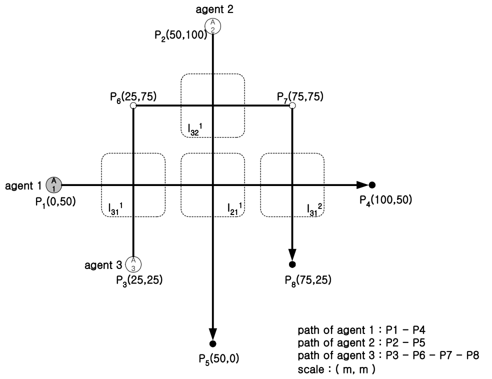


Figure 1: Three agents with intersection points.

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)}. C and T are shown in Fig. 2.

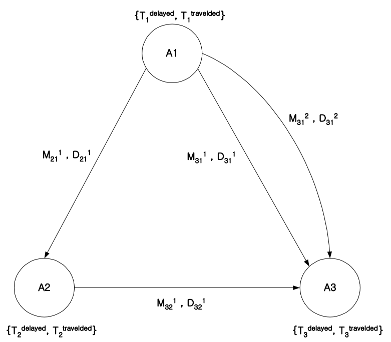

$\{T_1^{delayed}, T_1^{travelded}\}$
A1
M_{31}^2 , D_{31}^2
M_{21}^1 , D_{21}^1
M_{31}^1 , D_{31}^1
A2
A3
M_{32}^1 , D_{32}^1
$\{T_2^{delayed}, T_2^{travelded}\}$
$\{T_3^{delayed}, T_3^{travelded}\}$

Figure 2: Collision model for three agents in Figure1.

2.3 Collision-free Motion Planner based on Collision Model

2.3.1 Collision-free Motion Planner for a Robot on Collision Model

When a robot (A_i) is delayed by T_i^d, the collision characteristics related to the robot in the model are affected. For inlet links from the higher priority robots, M's increase and D's decrease by delayed departure time (T_i^d). In the other, for outlet links to lower priority robots, M's decrease and D's increase by the same amount. And as a result of the time delay, the safe inlet link may be dangerous. So we proposed an iterative approach to find the minimum delayed departure time for collision avoidance as followed:

Step1: Extract the links on which the robot is expected to collide with higher priority robots (Inlet Links) by use of collision characteristics.

Step2: Define an instantaneous delayed departure time (T_i^d) as the maximum of the Ds' in the selected links.

$$T_i^d = \max\left(\{D_{ij}^k \mid j \in P+(i), (i, j, k) \in E \text{ s.t. } M_{ij}^k > 0 \text{ and } D_{ij}^k > 0\}\right) \tag{3}$$

Step3: Modify node and link parameters by T_i^d.

Step4: Without dangerous inlet links to the robot, the robot can go to its destination safely. Otherwise, Execute above actions from the first stage.

2.3.2 Collision-free Motion Planner for Multi- robot on Collision Model

Step1: Select a robot from the priority order space (P) by use of priority index.

Step2: if the robot has the highest priority, go to first stage. Otherwise, apply the collision-free motion planner on collision model to the robot so that the robot can navigate safely.

Step3: if the selected robot has the lowest priority, all of the robots can navigate safely, and finish up this algorithm. Otherwise, increase priority index by 1 and go to first stage.

Te procedure of this algorithm for the three robots in Fig. 1 is shown in Fig. 3. Because all robots' links is in a safe state in Fig. 3(d), we can predict that the robots can navigate without collision among them.

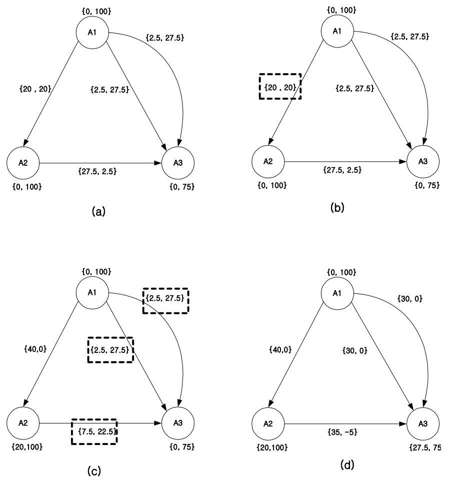


Figure 3: Procedure of collision-free motion planner on collision model for the agents in Figure 1.

3 PRIORITY SELECTION PROBLEM

The priority selecting problem deals with the determination of the priorities of multi-robots with the limited capability in speed and acceleration in order that the robots should arrive in the minimum time without any collision. In this paper, we express the problem as shown in Eq.(4) using (M,D) network model.

$$\begin{aligned}&\text{Objective : Minimize } J = \max(T_1^{\text{completed}},\\&\qquad \ldots, T_N^{\text{completed}}) \\&\text{Constraints : } M_{i,j}^{k} * D_{i,j}^{k} \leq 0,\ \forall (i,j,k) \in E\end{aligned} \tag{4}$$

Where, $T_i^{\text{completed}}$ is the arriving time of robot i which is the sum of T_i^{delayed} and T_i^{traveled}. $M_{i,j}^{k}$*and $D_{i,j}^{k}$ are the collision characteristics of the k_{th} collision region between robot i and robot j.

T_i^{traveled} is affected by path information and mobility of only robot i. But T_i^{delayed} is determined according to priority order of the robots and relations with other robots which is expressed with P, E, and C on the (M,D) network model.

In this paper, we redefine the priority selection problem for N robots based on (M,D) network as a Traveling Salesman Problem(TSP). We define robots and connections between neighbors in a priority order list as arcs as nodes and arcs on the network model. And we add a zero node to the model in order to convert our priority selection problem to a TSP. We define costs of arcs on the network model as shown in Eq(5).

$$\begin{aligned}&C_{0i} = T_i^{\text{traveled}}\ ,\ C_{i0} = 0 \text{ for all agent i}\\&C_{ij} = \max[\,0, T_j^{\text{completed}} - T_i^{\text{completed}}\,]\end{aligned} \tag{5}$$

Once robots' motion profiles are planned and robots' priorities are determined, we can get a primitive (M,D) network model, G = {V,E,C,T,P} and apply our collision-free motion planning algorithm to the robots . As a result of this motion adaption, we obtain the values of C and T defined on the table 1. These actions proceeds according to our predefined procedures suggested in section 2.3 and its calculation time is defined as a polynomial equation of the number of robots.

Because $T_i^{\text{completed}}$ has T_i^{delayed} which is affected by the position of robot i in a priority list, the cost between robot i and robot j, C_{ij}, varies according to the positions in the lists. So, our priority selection problem is a dynamic asymmetric problem.

4 SOLUTION TO THE PROBLEM

A TSP is known as a NP-hard problem. So, it is difficult to solve the problem mathematically. Thus, in this paper, we suggest how to cut out less important robots in the meaning that they will not affect the completion time of whole robots and reduce the volume of search space. In order to cut down search space, we use a method, BL-EDF (Bottom level Earliest Deadline First) scheduler, which is a task planner for multi-tasks with common resources.

4.1 BL-EDF Scheduler

The objective function of multi-tasks with priorities operating on M CPU's is defined in Eq. (6). To overcome the drawbacks of the centralized approach, the extended collision map method applies several concepts as follows:

$$\begin{aligned}&\text{Objective : Maximize } J = \min(S_1, \ldots, S_N)\\&\qquad S_i = E_i - T_i^{\text{completed}}\\&\qquad T_i^{\text{completed}} = T_i^{\text{worked}} + T_i^{\text{delayed}}\\&\text{Constraints : More two tasks should not}\\&\qquad \text{operate on a CPU at the same}\\&\qquad \text{time}\end{aligned} \tag{6}$$

Where, N is the number of robots and E_i and S_i are the deadline time and slack time of robot i.

The BL-EDF scheduler applied to our priority selection problem is as shown in table 2.

For example, there are 4 tasks in urgent group as shown in Fig. 4. For all tasks, delay times which are caused by tasks in urgent group and slack time are calculated. And if all slack times of a task are positive, the task is moved to less important task group. The procedure continues until there is no task in urgent group of which slack times are positive.

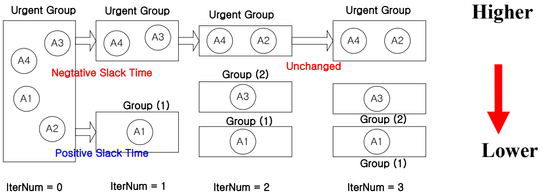


Figure 4: Procedure of BL-EDF scheduler.

4.2 Reduction of Search Space

The priority selection problem for multi-robot motion is converted to the priority selection problem for multi-task execution by defining deadline time of tasks(E_i), and tasks execution times (T_i^{worked}) in

Eq.(6) as the maximum of robot traveled times(max[$T_1^{traveled}$, …, $T_N^{traveled}$]) and traveling times of robots($T_i^{traveled}$) in Eq.(4). And a robot's maximum delayed time is equal to a sum of all M's and D's of collision regions related to the robot. Therefore, task's delayed time in redefined multi-task priority selection problem is determined with a sum of all M's and D's of inlet link to the robots as shown in Eq. (7).

Table 2: BL-EDF scheduler.

Procedure

```
Step 1 <Initialization>

  1.1
  Assign all Tasks to Urgent Group(UG), IterNum ← 1,
  UGN ← N, k ← 0
  Calculate $T_i^{worked}$ of all Tasks (i=1 to UGN )

Step 2 <Detecting Not Urgent Tasks >

  For all tasks in UG (i=1 to UGN) Do
    2.1 Determine $T_i^{delayed}$ by sum of possible delayed
    time for tasks in UG
    2.2 Calculate $T_i^{completed}$ and $S_i$
    2.3 Classify Tasks by signs of $S_i$
      If $S_i > 0$, Then move task i to N(I) and increase k by
      1
    End of Loop – i
  2.4 If k=0 or UGN = 1, Then Go to Step 3,
  Else UGN ← UGN – k , k←0, increase IterNum by 1
  and Go to Step 2

Step 3 <Assign priority of All Tasks>

  For k=1 to IterNum Do
    3.1 Assign lowest priority to Tasks in N(k), Task
    order in an N(k) is not important
    End of Loop – k
  3.2 Examine all the priority order for Tasks in UG
  which maximize J
  If J is negative, Then return FALSE
  Else Assign this priority order upper tasks in
  N(IterNum)

Complete scheduling
```

$$T_i^{delayed} = \sum_j \sum_k (M_{ij}^k + D_{ij}^k) \text{ where } j \in P_i^+ \quad (7)$$

The procedure of BL-EDF scheduler for 5 robots is shown in Fig. 5. The (M,D) network of the robots with priority order {1-2-3-4-5} is shown in Fig. 5(a). In Fig. 5(b), links has no direction and its value is equal to a sum of M and D in order to apply BL-EDF scheduler to the robots. In Fig. 5(c), robots' maximum completed time and slack time are calculated. In Fig. 5(c), because robot 4 (A4) has a positive value, robot 4 can be removed from urgent group. The (M,D) network model of remain robots without robot 4 is shown in Fig. 5(d) where robot 3 has a positive slack time. Therefore, robot 5 moves from urgent group to a less important group. This procedure iterates until there is no change in urgent group or there is one robot in the group as shown in Fig. 5(j). Finally, we get priority order {2-1-5-3-4} and its (M,D) network model is as shown in Fig. 5(k).

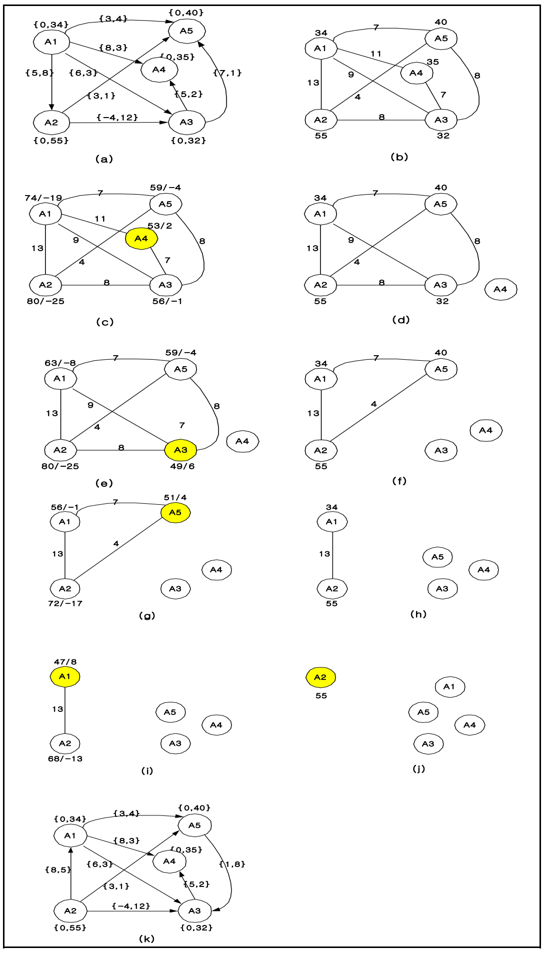


Figure 5: Application of BL-EDF scheduler for multi-robots.

5 SIMULATION RESULTS

In our simulation, robots were modeled circles of which radius, speed is 0.5 [m] and 1[m/sec] each. Their speed model was assumed to be instantaneous such that it took no time for them to accelerate from stationary status to full speed. And the robots were located uniformly in a square with 100[m] side length. We assumed that robots' paths were Manhattan city typed paths. And we assumed that their traveled times and sums of collision characteristics, M and D, were distributed uniformly in [40, 80] and [4, 20] each. Finally, we assumed delayed times (D's) of collision regions were distributed uniformly in [-8, 8].

We did 100 simulations for the 20, 30, 40, and 50 robots with 3 times intersections as many as the numbers of robots. In the Fig. 6, the x-axis shows the ratios of the numbers of intersections to the numbers of robots. This variable expresses the mean of the number of intersections which each robot has with other robots and is related to of environments. And the y-axis shows the ratios of the numbers of robots in final urgent group to the numbers of robots. This variable expresses effectiveness of BL-EDF.

Regardless of the number of robots, when normalized numbers of intersections are around 2, normalized numbers of robots in final urgent group are about 0.5. Therefore, we expect that our BL-EDF may reduce 50% in the number of robots in urgent group. In some applications including social security field, it is reasonable to assume 12 – 14 robots of which normalized number of intersections is 2. Therefore we expect that our algorithm suggested in this paper cut down calculation time needed to determine priority orders of multi-robots to 0.01 % of original expected one.

6 CONCLUSIONS

In this paper, we converted a priority selection problem for multi-robots with collision-model based motion planner to a priority selection problem for multi-tasks with common resources. And we showed that this problem is a TSP. Thus, we applied BL-EDF for multi-tasks to our priority selection problem in order to cut down search space. And effectiveness of our algorithm in this paper was proved with simulation results. In future, advance in information technologies and communications is expected to help the proposed approach be more practical in social security applications.

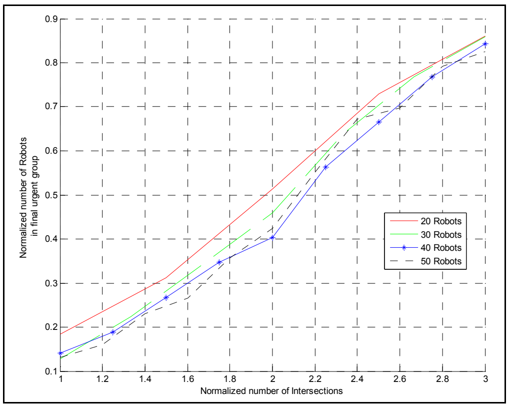


Figure 6: Results of BL-EDF scheduler for multi-robots.

ACKNOWLEDGEMENTS

This work was supported by the Program from the Ministry of Knowledge Economy (MKE).

REFERENCES

J. F. Canny, 1988. *The Complexity of Robot Motion Planning*, MIT Press.

S. Akella and S. Hutchinson, 2002. *Coordinating the Motions of Multiple Robots with Specified Trajectories*,

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.

Haoxum Chen, Luh P.B., and Lei Fang, 2001. *A Time Window Based Approach for Job Scheduling*, Proc. of IEEE Int. Conf. on Robotics and Automation.

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.

M. Bennewiz, W. Bufgard, and S. Thurn, 2001, *Optimizing Schedules for Prioritized Path Planning of Multi-Robot Systems*, Proc. of IEEE Int. Conf. On Robotics and Automation.

S. H. Ji, et. Al., 2009, *Collision-Model Based Motion Planner for Multi-Agents in a Factory*, Proc. of 6 th Int. Conf. on Informatics in Control, Automation and Robotics, pp.378-386.

J. Park, M. Ryu, and S. Hong, 2003, *Fair Real-Time Resource Allocation for Internet End System's Qos Support*, Lecture Notes in Computer Science, vol. 2713, pp. 764-789.

DEVELOPMENT OF LIGHTWEIGHT DUAL ARM ROBOT BY USING HOLLOW SHAFT SERVO ASSEMBLY

Min-Kyu Park
Division of Mechanical Engineering, Yeungnam College of Science & Technology
274 Hyeungchungro, Deagu, South Korea
mk_park@ync.ac.kr

Seok-Jo Go
Division of Mechanical Engineering, Dongeui Institute of Technology, 152 Yangjiro, Pusan, South Korea
sjgo@dit.ac.kr

Young-Jin Lee
Autopower Co. LTD., 1048-9 Obang-dong, Kimhae, Kyungnam, South Korea
ceo@autopower.co.kr

Keywords: AUTOMAN, Dual Arm Robot, Hollow Shaft Servo Assembly, Lightweight Robot Arm.

Abstract: Studies on dual arm robots have been continually carried out because the robots provide human-like flexibility of movement. In previous study, the dual arm robot with solid shafts had been designed. However, the robot has some defects in its heavy weight and big shape. In this study, a hollow shaft servo assembly for lightweight robot arm has been developed. And then, the dual arm robot (AUTOMAN) has been designed using a hollow shaft servo assembly.

1 INTRODUCTION

Single arm industrial robots are applied in the process where the simple repetitive operation is much like the automotive industry and the semiconductor industry. There are many tasks that are difficult or impossible to do with the single arm robots. The dual arm robot is able to solve these problems (University of Texas at Austin, DLR, Yaskawa Co.). In the previous study, the dual arm robot with solid shafts had been developed as shown in Figure 1 (Park, 2008). The robot with solid shafts is easily implemented. However, as the degree of freedom of a robot rises, when implementing the complicated motion, some problems are appeared such as kink and abrasion of cables. As to the dual arm robot with hollow shafts, the inside of an arm and a motor shaft are empty. It is free from a constraint by the electrical wire in the implementation of complicated motion in comparison with the robot with solid shafts (Han, 2008). Also, the load to weight ratio of a robot is improved. Accordingly, basic researches on the hollow shaft servo assembly had been carried out in the previous studies (Han, 2008; Lee, 2008; Go. 2009).

The goal of this study is to develop a dual arm robot with the hollow shaft servo assembly. Firstly, the hollow shaft servo assembly for applying to the dual arm robot is developed. Secondly, the performance of the completed servo assembly is evaluated. Finally, the industrial hollow shaft articulated dual arm robot (called as AUTOMAN) tries to be developed.

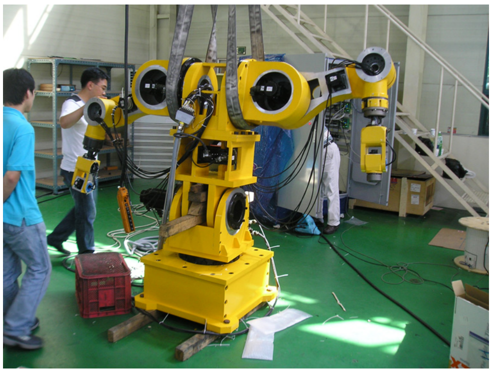


Figure 1: Dual arm robot developed in previous study.

2 HOLLOW SHAFT SERVO ASSEMBLY

In order to apply hollow shaft to industrial dual arm, the specifications of each axis have to be determined. The payload of the robot is set as 20kg. The specifications of the each hollow shaft servo assembly are determined by analyzing the maximum torque and the steady state torque under tracking trajectories at full reach within given time like Figure 2. The capacity, torque and maximum velocity of each axis are finally obtained by applying a safety factor to analyzing value as shown in Table 1 (Lee, 2008).

The assembly drawing of the hollow shaft servo assembly for the dual arm robot is shown in Figure 3. The key components comprising the servo assembly are the hollow shaft servo motor, the hollow shaft multi-turn absolute encoder, the hollow shaft brake, and, the hollow shaft harmonic drive. Table 2 shows the axis specification of the developed hollow shaft servo assembly.

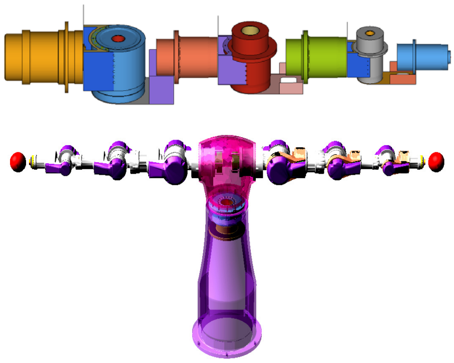


Figure 2: Dynamic critical conditions of dual arm robot.

Table 1: Specifications of servo motors.

Axis	Capacity	Torque	Velocity
1, 2	2.0 kW	6.8 Nm	3,000 RPM
3, 4	0.9 kW	2.9 Nm	3,000 RPM
5	0.6 kW	1.9 Nm	3,000 RPM
6, 7	0.4 kW	1.3 Nm	3,000 RPM

Table 2: Specifications of hollow shaft servo assembly.

Axis	Capacity [kW]	Torque [Nm]	Velocity [rpm]	Reduction gear ratio	Weight [kg]	Length [mm]
1	2.0	6.8	3,000	120	23.5	270
2				120	23.9	270
3	0.9	2.9		120	11.6	236
4				120	12.5	236
5	0.6	1.9		80	5.1	237
6	0.4	1.3		80	4.9	213
7				80	4.3	213

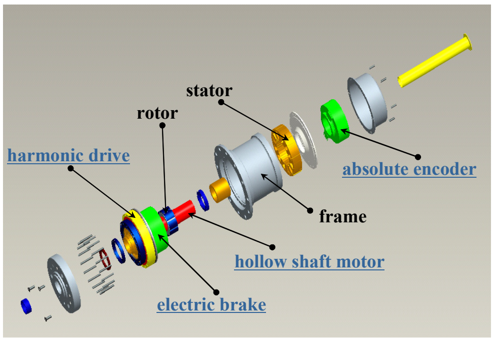


Figure 3: Hollow shaft servo assembly.

2.1 Hollow Shaft Servo Motor

As to the development of the hollow shaft servo motor, because all cables enter into the hollow shaft, the inner diameter of hollow shaft is as big as possible and it should not have a problem in an assembly between the hollow shaft encoder and the hollow shaft brake. Moreover, because the hollow shaft rotates, the protecting duct in which it protects the cable has to be installed and the axis of motor has to be designed by considering the connection with the axis of a reduction gear.

Accordingly, the hollow shaft servo motor was developed in consideration of the hollow size and the axis capacity of the motor. Figure 4 shows the developed hollow shaft servo motor. The cores of stator and rotor used the Si-steel S18 and S30 respectively. And the magnets of these used Nd-Fe-B and N-35SH.

Since being the servo system, sensors such as a resolver or a hollow shaft absolute encoder is essential. Accordingly, the hollow shaft absolute encoder is selected as shown in Table 3 (Tamagawa Seiki Co.).

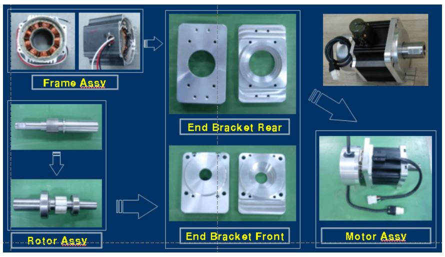


Figure 4: Hollow shaft servo motor.

Table 3: Specifications of applied encoder.

Axis	Resolution [bit]	Revolution [RPM]	Weight [kg]
1,2	33	1,500	2.1
3~7		3,000	1.2

2.2 Hollow Shaft Brake

The development of the hub shape type brake is needed, because the hollow shaft of the servo motor can be penetrated. The excitation functional type brake stops a rotator when a current is electrified in a coil. Therefore, this brake is suitable to the machine with horizontal motion. On the other hand, as to the electro-magnetic operable brake, when a current is electrified in a coil, the braking power loses and a rotator is free. This type brake is usually applied by the machines with vertical motion. In this research, since the vertical motion had to be possible in the small and narrow place, the hollow shaft brake of the non-excited operation type is developed by considering the required torque and dimension of each axis. Figure 5 shows the developed hollow shaft brake and the specifications are shown in Table 4.

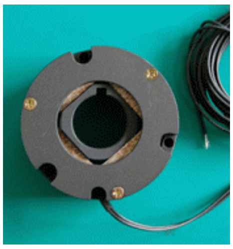


Figure 5: Hollow shaft brake.

Table 4: Specifications of developed hollow brake.

Axis	Frictional Torque [Nm]	Weight [kg]	Outer Dia. / Inner Dia. / Thickness
1,2	6.86	2.77	147 / 51 / 38.3
3,4	2.94	1.18	104 / 41 / 36/2
5,6,7	2.45	0.54	74.8 / 26 / 29/8

2.3 Hollow Shaft Harmonic Drive Unit

The reduction gear applied to the dual arm robot has to be fitted the hole size of the hollow. And not only the high-torque and but also the high precision are needed. The hollow shaft harmonic drive made by Harmonic Driver Systems in Japan is applied (Samick HDS). The motor shaft and the axis of reduction gear are connected as shown in Figure 6. The test results of eccentricity and torsion stress are satisfied the required specifications as shown in Table 5. The specifications of developed hollow shaft harmonic drive are presented by Table 6.

2.4 Performance Test of Hollow Shaft Servo Assembly

After making one side of the dual arm robot as Figure 7, the test of repeatability carried out. Table 7 shows the specifications of the developed hollow shaft robot arm. In the performance evaluation about the robot arm, the repetition precision about the location tracking was verified according to the KS 7082 standard (KS B 7082,) After from P1 to P5 designated the coordinate in which a robot has to follow according to the three-dimensional space-phase like Figure 8, the repetition tracking was performed by 30 time series. The laser tracker was used to measure the positions. Results of repeatability test are as shown in Figure 9. The average repeatability about each site is ± 0.05 mm within and it is satisfactory.

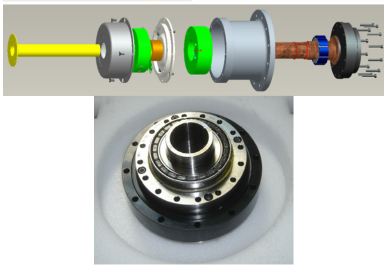


Figure 6: Hollow shaft harmonic drive.

Table 5: Performance test results of the unit.

	Spec.	Test 1	Test 2
Concentricity1	0.02 and less	0.0103	0.0122
Concentricity2		0.0047	0.0185
Diameter[mm]	50 + 0.015	50.0144	49.9985
Length[mm]	300±0.3	300.0625	300.1394
Torque[Nm]	More than 4.7	274.5	246.37

Table 6: Specifications of the hollow harmonic drive.

Axis	Ratio	Torque [Nm]	Speed [RPM]	Weight [kg]	Outer/Inner [mm]
1,2	120	523	3,000	6.9	190 / 52
3,4	120	178	3,000	3.1	142 / 36
5,6,7	80	44	3,000	0.89	90 / 21

Table 7: Specifications of developed hollow Shaft robot arm.

	Weight	Length	Payload
Specification	94kg	1,574mm	20kg

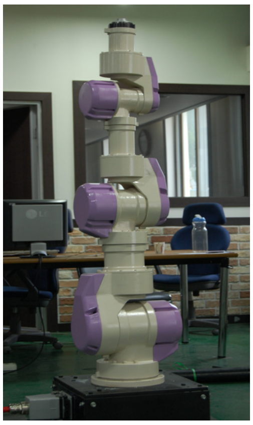


Figure 7: Developed hollow-type robot arm.

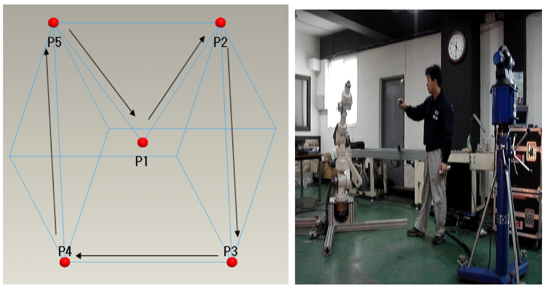


Figure 8: Reference trajectory.

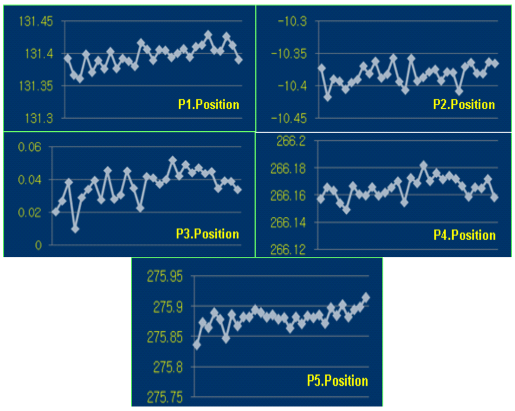


Figure 9: Results of repeatability test. [unit : mm].

3 DESIGN OF DUAL ARM ROBOT

In the previous research, the solid type dual arm robot like Figure 1 had been developed (Park, 2008). The dual arm robot developed in the previous research has the both arms of 6 degree of freedoms with the payload 20 kg, and the body (torso) of 2 degree of freedom. However, as to the developed dual arm robot, some problems have occurred such as the overweight compared with payload, the increase of the servo motor capacity due to a volume and weight enlargement, a kick and an abrasion of cables. In order to solve these problems, the hollow shaft servo assembly is developed. In the development of the robot, it tried to lighten through the stress analysis and optimal design. The servo control systems like servo drivers and controller are used by commercial controller.

3.1 Design and Fabrication of Robot

The dual arm robot makes to put an emphasis on the improvement of the weight and movement. Housing, frame, cover and the key element parts of robot are developed by considering the design specifications as shown in Figure 10. Especially, in the case of the housing, in order to satisfy the maximum allowable deflection 3 mm at full reach of the horizontal direction, the target rigidity is obtained by the CAE analysis (Lee, 2008). Figure 11 shows the dual arm robot (AUTOMAN) in which it makes in this research. It can show the reduction of weight and the improvement of design. AUTOMAN is composed of two arms and one twist joint. Each arm has 7 DOF. Therefore, AUTOMAN totally has 15 DOF. The robot can be designed so that cables can pass through the central part of a robot. The length of the arm is about 1,500 mm. Table 8 shows the specification comparison with the solid type dual arm robot.

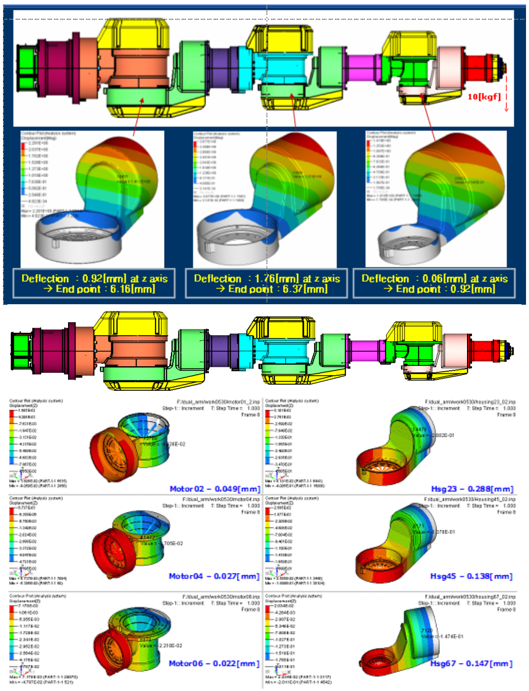


Figure 10: Design of motor frame and housing.

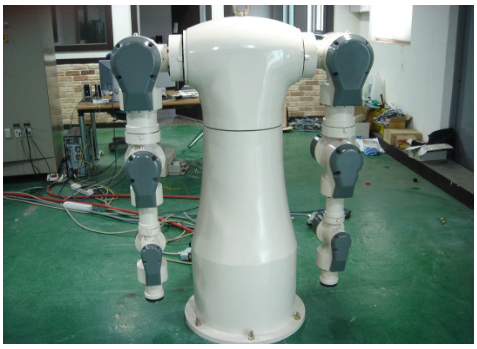


Figure 11: AUTOMAN.

Table 8: Specifications of developed dual arm robot.

Item	Solid type dual arm robot (Fig. 1)	AUTOMAN
Controlled axes	14(Right 6, left 6, rotate 2)	15(Right 7, left 7, rotate 1)
Payload	20 kg	20 kg
Mass	500 kg	300 kg
Repeatability	±0.7 mm	±0.1 mm
Maximum Speed	100 °/s	100 °/s

3.2 Development of Control System

The robot control system is designed for considering convenience of developing environment, powerful computing, real-time operation ability. Figure 12 shows the control system of the robot. The MEI ZMP controller (The Motion Engineering) and PMAC controller (DELTA TAU) were examined in order to select the controller in which it is suitable for the dual arm robot (MEL, DELTA TAU). It is altogether able to use with these controllers in case the moving track of a robot is predetermined.

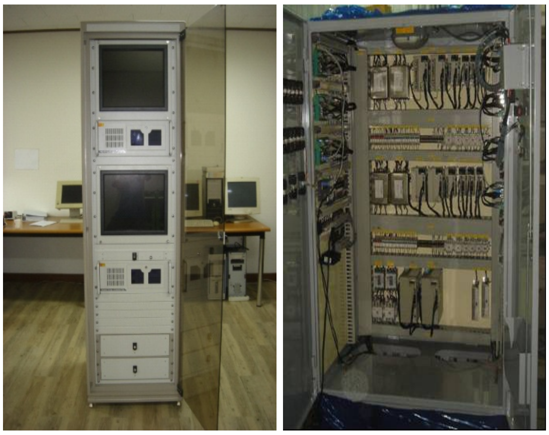


Figure 12: Control system of AUTOMAN.

However PMAC controller has evaluated that it could not be guaranteed a real-time ability in case the moving track of a robot is determined in real time such as a visual servoing control and a force control. Accordingly, in order that the target platform of the control system was determined, the ZMP board (DELTA TAU), that is the commercial controller in which a performance is verified, was selected as the controller of the dual arm robot.

4 CONCLUSIONS

This research tried to be comprised the light weight of the dual arm robot by the hollow shaft servo assembly, and a miniaturization and highly-rigid in order to solve a problems reflected in the previous developed dual arm robot. The hollow shaft servo assembly is developed by designing and fabricating the hollow shaft servo motor, the hollow shaft brake, and so on. And the dual arm robot (AUTOMAN) for the flexible package operation is designed and made. The control system is made in consideration of a convenience and operation processing speed of the development environment. The performance of the arm is confirmed that the average repeatability is measured by ± 0.05 mm within and it is satisfactory. Moreover, by installing in the cable, kink and friction problems were solved. It is finally remarked that the comparison between the solid type robot and the developed hollow type robot will be undertaken as a second phase of this work.

ACKNOWLEDGEMENTS

This work was supported by the Ministry of Commerce, Industry and Energy under Grant No. 10025003.

REFERENCES

University of Texas at Austin, http://www.robotics.utexas.edu/rrg/.

DLR, http://www.dlr.de/en/.

Yaskawa Inc., http://www.motoman.com/.

Park, C. H., Park K. T., and Kim, D. H., "Design of dual arm robot manipulator for precision assembly of mechanical parts," *2008 International Conference on Smart Manufacturing Application*, pp. 424-427, April 2008.

Han, H. G., Go, H. Y., Lee, Y. C., Jung, C. G., Shin, Y. S., Go, S. J., and Lee, Y. J., "Development of Hallow

Shaft Servo Assembly for Light Weight Robot," *Control, Automation, and Systems Symposium 2008*, pp. 403-406, October 2008.

Lee, J. W., Han, H. G., Jung, S. W., Yoo, W. S., and Lee, Y. J., "Design of Dual-Arm Robot Using CAE Techniques," *Control, Automation, and Systems Symposium 2008*, pp. 407-410, October 2008.

Go, S. J., Han, H. G., Go, H. Y., Lee, Y. C., Jung, C. G., Shin, Y. S., and Lee, Y. J., "Development of Hallow Shaft Servo Assembly for Dual Arm Robot and Its Performance Evaluation," *KACC 2009*, pp. 58-61, September 2009.

Samick HDS, http://www.samickhds.co.kr.

Tamagawa Seiki, http://tamagawa-seiki.com.

KS B 7082, ISO9283, Manipulating industrial robots: Performance criteria and related test methods.

MEI, Inc., Product Manual.

DELTA TAU Data Systems, Inc., *Product Manual*.

STATIC BALANCE FOR RESCUE ROBOT NAVIGATION
Translation Motion Discretization Issue within Random Step Environment

Evgeni Magid and Takashi Tsubouchi
ROBOKEN - Intelligent Robot Laboratory, University of Tsukuba, Tsukuba, Japan
$\{$*evgeni, tsubo*$\}$*@roboken.esys.tsukuba.ac.jp*

Keywords: Random step environment, Rescue robot navigation, Teleoperation, Static balance.

Abstract: The goal of rescue robotics is to extend the capabilities of human rescuers while also increasing their safety. During the rescue mission mobile robot is deployed on the site and is operated from a safe place by a human. A decision on the robot's path selection is very complicated, since the operator cannot see the robot and the environment. Our goal is to provide a kind of automatic "pilot system" to propose the operator a good direction or several options to traverse the environment, taking into account the robot's static and dynamic properties. To find a good path we need a special path search algorithm on debris. The real state space of the search is extremely huge and to decrease the number of search directions we discretize robot's motion and the state space before the search. Search algorithm needs a proper definition of node's neighborhood, which will ensure smooth exploration of the search tree. In this paper we present our results in estimation of the transition possibilities between two consecutive states, connected with a translation step, and discuss the problems arising from the discretization of the state space. Exhaustive simulations were used to structure, analyze and solve the discretization issue problems and help to remove unsuitable directions of the search from the search tree.

1 INTRODUCTION

A long standing goal of mobile robotics is to allow robots to work in environments unreachable or too hazardous to risk human lives. Urban search and rescue is one of the most hazardous environments imaginable with victims often buried in unreachable locations. Rescue robotics is the application of robotics to the search and rescue domain. The goal of rescue robotics is to extend the capabilities of human rescuers while increasing their safety. In particular, the inside of severe earthquake stricken buildings or underground area should be investigated in advance of manned rescue operation in order to avoid risk of suffering from secondary disaster. During the rescue mission the mobile robot is deployed on the rescue site, while the human tele-operator is monitoring the robot's activities and giving the orders from a safe place outside of the site (Figure1). The system consists of a robot control subsystem and a remote operation station, connected with a wireless LAN.

In this paper we present our current results in estimation of the transition possibilities of a crawler type vehicle between two consecutive states, connected with a translation step within Random Step Environment, which is a simulated debris environment, proposed by NIST (Jacoff et al., 2001). To find a good path we implement a special path search algorithm on debris. Because the debris site is dangerous and unstable, the main goal of the algorithm is to keep the robot maximally stable at every step of its path. The real state space of the search is extremely huge and to decrease the number of search directions we discretize robot's motion and the state space before the actual search. A search algorithm utilizes a search tree (Cormen et al., 2001); for our problem dynamically created search tree cannot be explicitly presented as a skeleton. To present it as a function $F(Args) = Res$, where arguments $Args$ are the robot's current configuration and the environment and output Res is a set of accessible within one step configurations, we need a proper definition of function F which will guide the tree search. We created a theoretical basis for this function F and confirmed it with exhaustive simulations, removing all unsuitable directions of the search from the search tree. A new feature of our research is a path evaluation process, which includes robot's postures quality together with the inter posture transitions quality.

Currently rescue robots are operated manually by human operators. The remote operator uses only visual information about the environment, which is usually not sufficient to carry out complex tasks because of the limited visual fields of cameras. In the case of an on-site operator, which stays inside a crawler-type rescue vehicle, the human can feel the inclination of the vehicle and the decision on the traversability of the path becomes more easy. Unfortunately, the off-site operator can not use any of those natural biological sensors and have to judge on the next move on the base of the partial available information, taking subjective and time consuming decisions. Many optional paths from the start to the target position exist and it is very hard for the operator to choose a good and safe path. Transferring the function of taking such decisions to a computer will decrease the burden on the operator. Our final goal is to provide a ”pilot system” to propose an operator a good direction or several options to traverse the environment. The operator will receive a proposal on a good path from the ”pilot system” on the computer display by means of GUI and apply it in a real scenario driving KENAF robot.

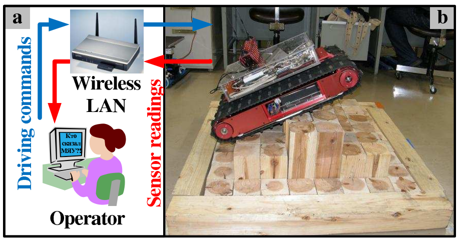


Figure 1: (a) Operator. (b) KENAF traversing RSE.

2 THE SYSTEM FRAMEWORK

The National Institute of Standards and Technology (NIST) has created a set of reference test arenas for evaluating the performance of mobile autonomous robots performing urban search and rescue tasks (Jacoff et al., 2001). One of the examples, simulating cluttered environment with debris is a so-called Random Step Environment(RSE) or Stepfield, which is widely used in the RoboCup Rescue competitions and rescue related research (Sheh et al., 2007). RSEs are designed to be easily reproduced, and yet behave in a similar way to real rubble. RSE consists of a final number of random steps of some minimal size simulating a heavily damaged environment of the buildings after the earthquake(Figure1b).

Our RSE is constructed from wooden block cells of 85x85mm size and 0, 90, 180 or 270 mm height each, where 0mm corresponds to the ground level around the RSE-patch. We assume a simple tractor-like crawler non-reconfigurable robot, corresponding to the main body of ”KENAF” robot(Figure2b). The main body of ”KENAF” consists of two large tracks with a small gap in between; the main specifications of ”KENAF” without sensors, front and the back pairs of arms, used in experiments and by the simulation ”pilot system”, are given in table 1.

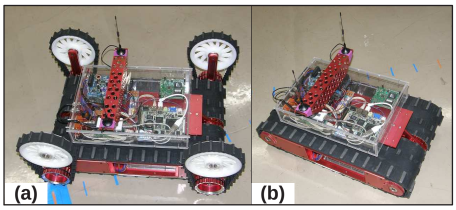


Figure 2: (a)Full KENAF configuration without sensors. (b) Main body without service arms and sensors.

Table 1: Specifications of ”KENAF” in basic configuration.

Parameter	Measurement
Maximal inclination	
dynamic	60 deg
static	80 deg
Main body length	584 mm
Main body width	336 mm
Track width	150 mm
Hight	270 mm
Weight	17.8 kg

3 STATIC STABILITY AND BALANCE ESTIMATION

The most important question which the path search algorithm should be able to answer is if a specific robot configuration is possible or not. This includes not only collisions with the obstacles, but also the capability of the robot to keep the current configuration. The robot should be able to stay in the specific configuration without slippering or turning upside-down. In other words, a safe and reliable motion of an autonomous vehicle requires continuous satisfaction of static and dynamic constraints (Shoval, 2004). Static stability is a minimal necessary condition for the general vehicle stability. For the static stability satisfaction the robot’s center of mass (CM) must lie above the **support polygon** - a polygon with vertices at the contact points of the robot’s crawlers with RSE.

In (Magid et al., 2008) we presented an algorithm for static balance posture estimation of the robot’s posture in a specified configuration. In this section we give a brief description of the static balance posture types where we give color names, and add one new posture type - cyan. From the point of static balance estimation, we distinguish six posture types:

(R) Red State. Presents a statically unstable posture; it results in robot's turning upside down while trying to climb to an impossible steepness.

(G) Green State. Stands for a statically stable posture (Figure3a).

(Y) Yellow State. Is a private case of green state. Normalized Energy Stability Margin (NESM) (Hirose et al., 1998) is applied for estimating the quality of the green state posture. Green state turns into yellow if the S_{NESM} criterion does not exceed a predefined minimum established in the process of experiment trials with KENAF robot.

(O) Orange State. Is something between red and green states. This posture is possible, but not stable. It does not result in robot's turning upside down, but do not guarantee a single stable posture since there exist two options and the real one depends on the preceding posture and moving direction. Figure3b demonstrates a side view of an orange state with two possible postures. The orange state is very important, since it affords the robot to lose the balance on purpose, when for example the robot traverses the barrier. Traversing the barrier includes climbing up and going down with loosing balance twice on top of the barrier. We distinguish O_1 and O_2 cases. O_1 is the first part of the O-posture, before the robot looses its balance. O_2 is the second part, which occurs after the robot have lost its initial balance; robot changes its posture discontinuously at that point and obtains a balance again in a different body orientation.

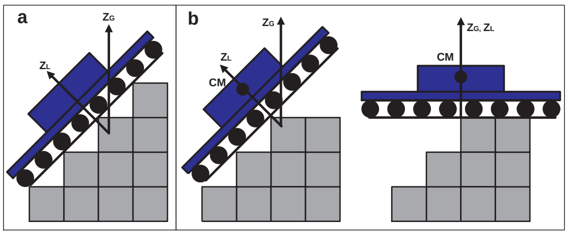


Figure 3: a) Green state b) Orange state: O_1 (left) and O_2 (right).

(M) Magenta State. Appears when the robot has to climb up or to slide down the vertical slope of the environment; it is legal only for translation motion.

(C) Cyan State. The cyan posture is detected between two successive G/Y/O postures if CM position change in Z-coordinate exceeds the predefined threshold. This jump is detected only after the second posture is explored and compared to the first posture. Cyan posture is legal only for a rotation motion.[1] Further we denote by C-posture a posture which static balance corresponds to a Cyan type, R-posture - red type etc.

4 SEARCH SPACE AND SEARCH TREE

Next important task of the search algorithm is to decide on possible next steps of the robot from a given current location and orientation. In standard 2D navigation there are 2 types of cells in the state space "free" and "obstacle" and all transitions between free adjacent cells are legal (Latombe, 1991). Our important improvement of the existing approach is as follows. We have "possible" (stable) and "impossible" (unstable) postures with regard to robot's static stability; but even in a case of two adjacent "possible" postures we check if the transition between them is legal. To decrease the number of search directions we discretize robot's motion and the state space before the search. Next exhaustive simulations and experiments will remove all unsuitable search directions.

Initially we had chosen 3 levels of search space discretization for XY-coordinates of the environment:

- DISC2 - each 85x85mm cell of RSE turns into 2x2 cells of the internal robot map with the cell size of 42.5x42.5mm
- DISC5 - 5x5 cells of 17x17mm size per RSE cell
- DISC10 - 10x10 cells of 8.5x8.5mm size per cell

KENAF supports two types of motion: translation and rotation. Thus at each node of the search tree the search algorithm has to open the 3-neighborhood of the node - go straight or turn left/right - and to proceed the search in the most promising direction. Immediately we must cut off from the search tree all impossible search directions, which are different for rotation and translation steps. In this paper we present our results in estimation of the transition possibilities between two consecutive states, connected with a translation step.

Translation step is defined as a one cell length step forward in the direction of robot local frame's axis X_L. Transition between two stable postures is

[1] Exhaustive simulations showed that cyan posture exists for the translational motion only due to the discretization issue: it exists for a big scale of the internal map of DISC2 (as defined in section 4), rarely appears at DISC5 and almost disappears at DISC10.

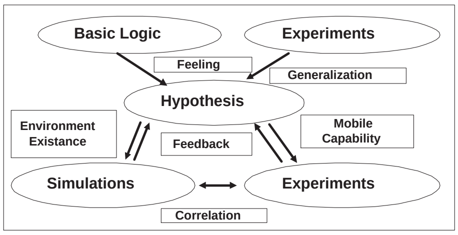


Figure 4: Theory, simulations and experiments.

not always possible and to define this condition, we created a set of theoretical hypotheses, based on our experimental experience. Exhaustive simulations for environments existence in MATLAB and experiments with KENAF robot in RSE gave a valuable feedback for our theory and finally produced a branch cutting condition for the path search algorithm(Figure4). Successive transition patterns will be integrated in the search algorithm as a part of neighbor opening and branch cutting function $F(Args) = Res$.

5 DESCRIBING A POSTURE

To characterize robot's posture qualitatively we use the coloring of the states. To decide possible transitions between two successive states, we have chosen 6 variables, whose combinations help us to define legal transitions between the states.

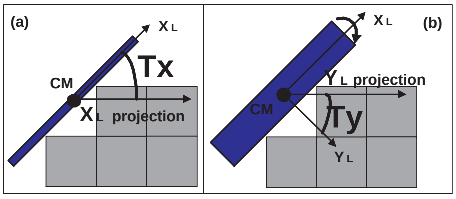


Figure 5: (a) Steepness θ_X. (b) Moment θ_Y.

Steepness θ_X. The angle, showing the steepness of the environment at a given robot configuration. Angle between the local axis X_L and its projection on the plane of the global axes X_G, Y_G(Figure5a).

Moment θ_Y. The angle, showing the dangerous rotational moment around robot's X_L-axis at a given configuration. Angle between the local axis Y_L and its projection on the plane of the global axes X_G, Y_G(Figure5b).

Contact Points Quality (CPQ). Depends on the angle θ_{CPQ} between the robot's crawlers and the edges of the RSE cells and affects the robot's ability of climbing the obstacles and sliding down safely.

Inclination. Is the sign of the *steepness* angle θ_X. We distinguish three groups of posture sets with respect to this parameter: $G_{U_{inc}}$ is a climbing up the steps of the environment posture, $G_{D_{inc}}$ is a going down and $G_{Z_{inc}}$ is a neutral inclination posture.

M-sign. Is the sign of the *moment* angle θ_Y. Similar to inclination, group $G_{P_{MS}}$ contains all postures with positive M-sign, $G_{N_{MS}}$ with negative and $G_{Z_{MS}}$ with neutral[2].

NESM-stability. Shows the probability of the robot's turning upside down because of the situation, when the CM is too close to one of the edges of the support polygon in the sense of NESM.

Inclination and *M-sign* are the most important variables. They signal about discretization problems, pointing on the missed posture between two successive postures due to the discretization issue. 4 other variables are emphasized for the experimental work and particulary for creating input, which will satisfactorily span all possible translation step cases.

6 HYPOTHESES

We conducted a large set of experiments with KENAF robot in several Random Step Environments. Based on operation experience and basic logic, we created a set of rules on the translations (TR) between two successive postures. It includes trivial statements, definitions and assumptions:

(TR1) Translation preserves orientation θ.

(TR2) Starting at O_1 posture, we immediately translate to O_2 as a result of inertia; there is no way to obtain O_2 posture without previously obtaining O_1.

(TR3) Z-posture is defined if robot's body is parallel to the ground level: $G_{Z_{inc}} \bigcap G_{Z_{MS}}$

(TR4) The only way to climb up or slide down a vertical slope of RSE is to apply a sequence of M postures between two stable postures (start and end of the *M-sequence*). *M-sequence* cannot contain any non-M posture.

(TR5) C-posture does not exist.

(TR6) M-posture has no real θ_X, θ_Y and NESM parameter data, but only an approximation. Also it has no own inclination and M-sign data.

(TR7) NESM-stability coefficient is zero for O-postures.

(TR8) Change of inclination between two postures can occur only through O-posture and its O_1 or O_2 is a Z-posture.

[2] $G_{Z_{inc}} : |\theta_X| \leq \varepsilon$; $G_{Z_{MS}} : |\theta_Y| \leq \varepsilon$; ε=1 degree

(TR9) Change of M-sign between two postures can occur only through O-posture.

(TR10) Significant change of steepness θ_X so that $\Delta\theta_X \doteq |\theta_X(P_{prev}) - \theta_X(P_{curr})| \geq TX_{MAX}$ without change in inclination or M-sign between two postures can occur only through M-posture.

(TR11) If the change between two successive postures in θ_Y is $\Delta\theta_Y \geq TY_{MAX}$ degrees, it can occur only through O-posture.

(TR12) The possible change between two successive postures for a uniform translation motion case is $\Delta\theta_X \leq TX_{MIN}$ and $\Delta\theta_Y \leq TY_{MIN}$ degrees.

(TR13) Inevitable O-posture (IOP) is obtained while passing the edge of the RSE-cell. If the posture (x,y,θ), preceding IOP, becomes ($x\pm\delta_x$,$y\pm\delta_y$,θ) with $\delta_{x,y} \to 0$, we will again and again obtain this IOP at the next translation step, i.e. IOP is preserved in the case of small shift δ in at least one direction.

(TR14) Accidental O-posture (AOP) is obtained while passing through the corner of the RSE-cell. If the robot posture (x,y,θ), preceding AOP, would become ($x\pm\delta_x$,$y\pm\delta_y$,θ) with $\delta_{x,y} \to 0$, this AOP will not be obtained at the next translation step, i.e. any small shift will result in a differently colored posture.

We calculated a theoretical value of TX_{MIN} for DISC5 by maximizing the difference $\Delta\theta_X$ between two successive G-postures with optimization method according table 1. Obtained theoretical value was updated through a set of simulations and finally set to $TX_{MIN} \doteq 3.5$ degree. In a similar way we defined $TX_{MAX} \doteq 8$, $TY_{MAX} \doteq 8$ and $TY_{MIN} \doteq 3.5$ degrees. Unfortunately, the intermediate postures of (TR8)-(TR11) cannot always be obtained explicitly due to the discretization issue problem of the search space.

7 SUCCESSIVE POSTURE GROUPS

We divided all possible pairs of postures, connected with a translational step, into groups. Each group contains theoretically possible or impossible sequence with regard to section 6. For the translation case a legal set of colors for each posture is $\{G, Y, O, M\}$ and we are supposed to obtain 16 groups of pairs at most.

7.1 Excluded and Forbidden Sequences

To decrease the number of groups we treat at the simulation level G and Y as a same G color. Since there is no real data for the M-cases by (TR6), we can treat it only as a color and a detailed study of G→M, M→M and M→G groups was done through experiments.

Pairs M→O, O→O and O→M are dangerous sequences, which should be forbidden and treated as R-posture. Such sequences are theoretically possible in very rare specific cases. Yet when the simulated path, containing such sequences, is to be repeated in the real world scenario by the operator, any small deviation may result into drastic path change and even into robot's turning up side down. As an example, consider a pair O→O: just within one step the robot have to loose the balance twice; this means climbing and going down through a corner of the RSE-cell with a very small contact square between one of the crawlers and a cell, being close to the AOP case (TR14).

7.2 Possible and Impossible Sequences

Next we present a detailed description of G→G and G→O pairs. In this section we denote each posture as C(I) where C is the color, I is the *inclination*. For example, G(Z) means a green Z-posture (TR3) and O(U) means an orange posture with U_{inc}.

G → G groups :

(GG1): P_1=G(Z)→P_2=G(D) due to the discretization issue by (TR8) there is a missed intermediate O-posture between P_1 and P_2(Figure6).

(GG2): G(D)→G(Z); by (TR4) there is a missed intermediate M-posture(Figure7).

(GG3): G(Z)→G(Z) corresponds to a trivial case when the robot is moving uniformly through the flat pattern of RSE.

(GG4): G(Z)→G(U); by (TR4) there is a missed intermediate M-posture.

(GG5): G(U)→ G(Z); by (TR8) there is a missed intermediate O-posture.

(GG6): a sign change for θ_X or θ_Y between 2 postures signals about a missed AOP(Figure8).

(GG7): $\varepsilon \leq |\Delta\theta_X| \leq TX_{MIN}$ and $\varepsilon \leq |\Delta\theta_Y| \leq TY_{MIN}$ - the robot is uniformly climbing up or going down with small 3D orientation changes in θ_X and θ_Y.

(GG8): G(U)→G(U), $\Delta\theta_X > TX_{MAX}$ and $|\Delta\theta_Y| < \varepsilon$. The robot is climbing up the vertical slope of the RSE through a missed M-posture(Figure9).

(GG9): G(D)→G(D), $|\Delta\theta_X| > TX_{MAX}$ and $|\Delta\theta_Y| < \varepsilon$. This is a rare situation, when the robot is sliding down the vertical slope of the RSE with a missed M-posture, followed immediately by loosing the balance on the edge and finally gets into P_2. Such translation is still possible but dangerous and may be chosen only when no other path option exists.

(GG10): $|\Delta\theta_X| > TX_{MAX}$ and $|\Delta\theta_Y| > \varepsilon$, in most cases corresponds to a dangerous AOP and should be forbidden.

(GG11): $TX_{MIN} \leq |\Delta\theta_X| \leq TX_{MAX}$ and $TY_{MIN} \leq |\Delta\theta_Y| \leq TY_{MAX}$, in most cases corresponds to a dangerous AOP and should be forbidden.

Pairs of type GG1 and GG5 are missing O-posture between two G-postures. Since O-posture is more important and has a higher cost in the path planning, we recolor the second posture P_2 of the sequence into O-color. Similarly, P_2 of GG2, GG4, GG7 and GG8 are recolored into M; and P_2 of GG6, GG10 and GG11 - into R.

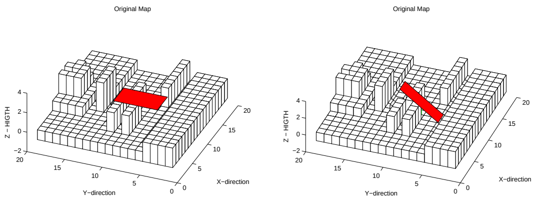


Figure 6: Translation GG1 from (left) to (right), missing intermediate O-posture.

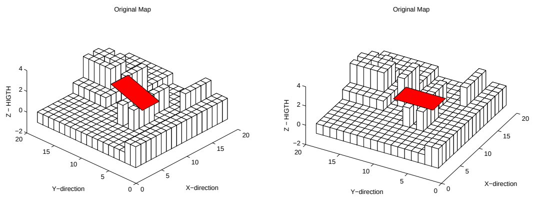


Figure 7: Translation GG2 from (left) to (right), missing intermediate M-posture.

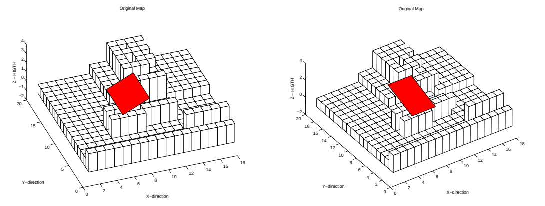


Figure 8: Translation GG6 from (left) to (right), missing intermediate AOP type O-posture.

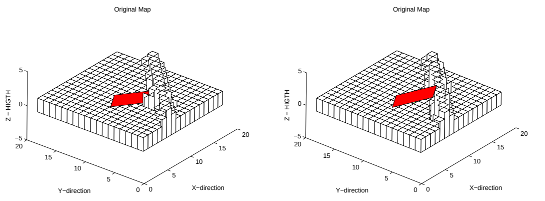


Figure 9: Translation GG8 from (left) to (right), climbing up through a missed M-posture.

Green → Orange: note that here P_2=O_1 and corresponds to the first part of the O-state *before* loosing the balance.

(GO1): G(U)→O(D) is forbidden; it occurs only if we missed another intermediate O-posture (TR8), which corresponds to a forbidden sequence O→O (section 7.1).

(GO2): G(D)→O(U) is similarly forbidden.

(GO3): G(D)→O(D) is a rare case, meaning loosing balance on the edge of the cell while going down and having a small change in θ_X or/and θ_Y.

(GO4): G(D)→O(Z) is an extremely rare case, which occurs when there is a missed intermediate M-posture. This mistake results into wrong calculations and the swap between O_1 and O_2; without this swap a jump between P_1 and P_2 would exceed the permitted threshold and P_2 would obtain R-color (Figure10: while the real posture sequence is a-b-c, the simulator understands it as a-c-b).

(GO5): G(U)→O so that P_2 has Z_{inc} and $|\theta_Y| > \varepsilon$. This situation occurs due to AOP and is forbidden.

(GO6): G(Z)→O(Z) is the most common case when the robot passes through the edge of the RSE cell while going down from the flat top of the barrier(Figure11).

(GO7): G(U)→O(U) is the most common case when the robot passes through the edge of the RSE cell while climbing up to the flat top of the barrier.

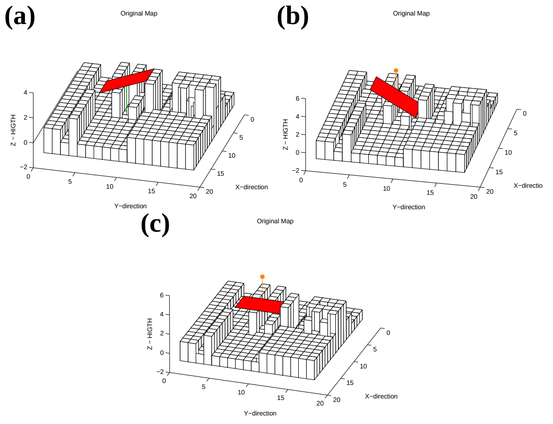


Figure 10: Translation GO4 from G (a) to O_1 (b), followed by O_2 (c), is missing an intermediate M-posture between (a) and (b).

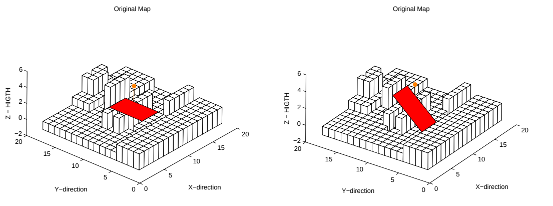


Figure 11: Translation GO6 - going down from the flat top of the barrier from O_1 (left) to O_2 (right).

While cases GO1, GO2, GO4 and GO5 are recolored into R-posture, cases GO3 should be accepted only if no other better choice of the path exists.

Orange → Green : Pairs O→G are exactly the same as G→G. The only difference is that the second posture for all cases different from GG3 and GG7 are recolored as R with regard to section 7.1.

8 SIMULATION

The only real proof of any theoretical hypothesis is an experimental proof. Thousands of different situations can occur in a completely random RSE and it is physically impossible to execute such huge number of experiments. The exhaustive simulations help us to conclude which situations can not occur due to the physical rules of RSE. Pairs of postures, impossible in the real world, are also impossible within the simulation. Since the reverse statement is not true, the simulator can not substitute the experiments, but assists to structure the data and remove the impossible types of sequences, saving time and efforts.

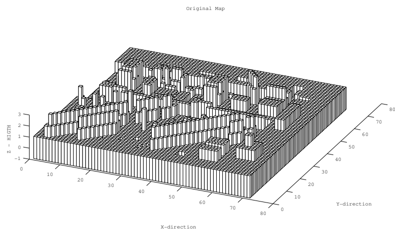


Figure 12: A complicated environment of size 61x61 cells, covering all main types of the environment obstacles.

We created a huge environment of 61x61 cells (Figure12) which includes all typical obstacles, usually appearing in the random step environment: horizontal and diagonal barriers, pairs of parallel barriers, traversable and non-traversable pikes and holes. For the simulation we have chosen discretization DISC5. An exhaustive check of all possible pairs of neighboring postures connected with a translation step were proceeded with voting for each group. As a first robot's CM posture of the pair we took every node of the grid[3]; a second posture of the pair was calculated as a 1-unit length change of CM's location in the direction of the robot's heading direction θ. The total number of posture pairs was more then 6 millions.

The simulation included 91 robot orientations $\theta \in \{0, \frac{\pi}{180}, \frac{2\pi}{180}, ..., \frac{89\pi}{180}, \frac{\pi}{2}\}$. In addition to pointing at the impossible (empty) cases the simulation reveals the rare cases and the most often cases.

The results are summarized in the following tables. Table 2 presents the distribution of the pair appearances before recoloring the second posture of the pair at discretizations DISC2 and DISC5. We listed as *legal* pairs GG3, GG7, GO6 and GO7; pairs, containing at least one R-posture are legal as well, but we present them as a separate case. *Undesirable* pairs include fixable cases GG1, GG2, GG4, GG5, GG8, GG9, GO3, which appearance we still would prefer to avoid. *Forbidden* pairs are GG6, GG10, GG11, GO1, GO2, GO4, GO5, OO, OM, MO and C (second posture of the pair is cyan). We excluded from the statistics all G→M, M→M and M→G cases, which otherwise would contribute 3.96%. Table 3 presents the distribution of the forbidden pair appearances in percents from the total pairs at DISC5; pairs GO1, GO2 and GO5 had zero appearance so they are excluded from the table. C-posture appearance signals on the missed M-posture. However, such M-posture should be forbidden, because the height difference between two G-postures exceeds KENAF's climbing abilities threshold: fast vertical change and is too dangerous when sliding down and impossible when climbing up. Table 4 presents the distribution of the undesirable pair appearances in percents from the total pairs at DISC5. Table 5 presents the distribution after the recoloring. For DISC5 legal pair percent varies from 93.1% to 95.84% for different θ choice, which means we have a wide enough range of options for choosing a good path at DISC5.

Table 2: Pair distribution in percents for DISC2 and DISC5.

DISC	Legal	Red	Undesirable	Forbidden
2	86.74	4.4	4.42	4.43
5	93.21	2.06	1.55	3.18

Table 3: Forbidden pair distribution in percents for DISC5.

GG6	GG10	GG11	GO4
0.33	0.57	1.38	0.009

MO	OO	OM	C
0.011	0.009	0.004	0.862

Table 4: Undesirable pair distribution in percents for DISC5.

GG1	GG2	GG4	GG5	GG8	GG9	GO3
0.08	0.55	0.66	0.05	0.11	0.1	0.006

Table 5: Pair distribution in percents for DISC2 and DISC5 after the recoloring.

DISC	Legal	Red	Undesirable
2	91.33	8.66	0.03
5	94.61	5.38	0.01

[3] The nodes too close to the borders of the map were excluded

9 DISCRETISATION ISSUE

The choice of a proper discretization of the search space is a very complicated task. We started our research with DISC5, which we considered to be good enough to notice main changes while traversing RSE. It turned that many of our initial theoretical expectations of robot's behavior in RSE did not fulfil. We called those cases *undesirable* and started a deeper exploration of what is happening in such cases.

We discovered, that those unexpected cases appear only due to the level of the discretization: discretizing the environment into 17x17mm cells turned to be too coarse to note all the changes and capture all the intermediate postures, obtained by the robot within one translational step. Increasing discretization to DISC10 decreases percentage of *undesirable* cases, but unfortunately can not solve the problem completely. If the level of the discretization would be infinitely high, we would definitely obtain a properly colored posture between *undesirable* pair of postures in every case. Here is a simple example of the discretization influence: a small analytical step from one G-posture to another G-posture of 10^{-28}cm length, approximated by MATLAB as $2.8422 \cdot 10^{-14}$, resulted into robot orientation change of $\{25, -15.5, -17\}$degrees with regard to global frame of RSE, signaling about a missed intermediate posture. This example shows that for any finite level of the discretization we still will have undesirable pairs appearances. Since we can not increase the discretization infinitely, we concluded that applying the results of section 7 for recoloring of the states at DISC5 is a good trade-off between execution time and precision. Of course, forbidding dangerous and suspicious transition, which still may be theoretically possible, limits our path choice, but increases the security of the practical use.

10 CONCLUSIONS AND FUTURE WORK

The final target of our research is to provide an assistant ”pilot system” for an operator of a rescue robot, decreasing the burden on the human operator. As soon as the robot obtains data from the environment and creates an internal world model, a selection on the path within the internal model should be done, followed by applying this path in the real world scenario. Since usually there exist more then just a single path, the path search algorithm needs a good instrument to evaluate the quality of each path. The search algorithm within the graph requires a proper definition of neighboring states to ensure smooth exploration of the search tree. In this paper we presented our results in estimation of the transition possibilities between two consecutive states, connected with a translation step. It is an important step toward a proper definition of a search tree neighborhood function $F(Args) = Res$, where arguments *Args* are the robot's current configuration and the environment and output *Res* is a set of accessible within one step configurations. We created a theoretical basis for function F and confirmed it with exhaustive simulations; the later were used to structure, analyze and solve the discretization of the RSE state space issue problems and help to remove unsuitable search directions. Next we plan to confirm our results with experiments and to complete function F with the theory for the rotation step neighbor node.

ACKNOWLEDGEMENTS

This research has been partially supported by NEDO Project for Strategic Development of Advanced Robotics Elemental Technologies, High-Speed Search Robot System in Confined Space.

REFERENCES

Cormen, T., Leiserson, C., Rivest, R., and Stein, C. (2001). Introduction to algorithms. In *Second Edition*. The MIT Press and McGraw-Hill.

Hirose, S., Tsukagoshi, H., and Yoneda, K. (1998). Normalized energy stability margin: generalized stability criterion for walking vehicles. In *1st Int.Conf. On Climbing and Walking Robots*.

Jacoff, A., Messina, E., and Evans, J. (2001). Experiences in deploying test arenas for autonomous mobile robots. In *Proc. of the 2001 PerMIS Workshop*.

Latombe, J. C. (1991). Robot motion planning. In *Proc. of the 2001 PerMIS Workshop*. The MIT Press and McGraw-Hill.

Magid, E., Ozawa, K., Tsubouchi, T., Koyanagi, E., and Yoshida, T. (2008). Rescue robot navigation: Static stability estimation in random step environment. In *Proc. of Int.Conf. on SIMPAR*.

Sheh, R., Kadous, M., Sammut, C., and Hengst, B. (2007). Extracting terrain features from range images for autonomous random stepfield traversal. In *IEEE Int. Workshop on Safety, Security and Rescue Robotics*.

Shoval, S. (2004). Stability of a multi tracked robot traveling over steep slopes. In *IEEE ICRA*.

VISUAL MAP BUILDING AND LOCALIZATION WITH AN APPEARANCE-BASED APPROACH

Comparisons of Techniques to Extract Information of Panoramic Images

Francisco Amorós, Luis Payá, Óscar Reinoso, Lorenzo Fernández and Jose M^{a} Marín
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: Appearance-based techniques have proved to constitute a robust approach to build a topological map of an environment using just visual information. In this paper, we describe a complete methodology to build appearance-based maps from a set of omnidirectional images captured by a robot along an environment. To extract the most relevant information from the images, we use and compare the performance of several compressing methods. In this analysis we include their invariance against rotations of the robot on the ground plane and small changes in the environment. The main objective consists in building a map that the robot can use in any application where it needs to know its position and orientation within the environment, with minimum memory requirements and computational cost but with a reasonable accuracy. This way, we present both a method to build the map and a method to test its performance in future applications.

1 INTRODUCTION

The applications that require the navigation of a robot through an environment need the use of an internal representation of it. Thanks to it, the robot can estimate its position and orientation regarding the map it has with the information captured by the sensors the robot is equipped with. Omnidirectional visual systems can be stood out due to the richness of the information they provide and their relatively low cost. Classical researches into mobile robots provided with vision systems have focused on the extraction of natural or artificial landmarks from the image to build the map and carry out the localization of the robot (Thrun, 2003). Nevertheless, it is not necessary to extract such kind of landmarks to recognize where the robot is. Instead of this, we can process the image as a whole. These appearance-based approaches are an interesting option when dealing with unstructured environments where it may be hard to find patterns to recognize the scene. With these approaches, the comparisons are made using the whole information of the scenes. As a disadvantage, we have to work with a huge amount of information, thus having a high computational cost, so we need to study compression techniques.

There are several researches that show compression techniques that can be used. For example, PCA (Principal Components Analysis) is a widely used method that has demonstrated being robust applied to image processing, (Krose et al., 2007). Due to the fact that conventional PCA is not a rotational invariant method, other authors introduced a PCA approach that, although being computationally heavier, takes into account the images with diverse orientations (Jogan and Leonardis, 2000). There are authors that use the Fourier Transform as a generic method to extract the most relevant information of an image. In this field, (Menegatti et al., 2004) defines the Fourier Signature, which is based on the 1D Discrete Fourier Transform of the image rows and gets more robustness dealing with different orientation images. On the other hand, (Dalal and Triggs, 2005) used a method based on the Histogram of Oriented Gradients (HOG) to the pedestrian detection, proving that it could be a useful descriptor for computer vision and image processing using the objects' appearance.

(Paya et al., 2009) present a comparative study of appearance-based techniques. We extend this study, taking into account three different methods: Fourier Signature, PCA over Fourier Signature and HOG.

2 REVIEW OF COMPRESSION TECHNIQUES

In this section we summarize some techniques to extract the most relevant information from a database made up of panoramic images trying to keep the amount of memory to a minimum.

2.1 Fourier-based Techniques

As shown in (Paya et al., 2009) it is possible to represent an image using the Discrete Fourier Transform of each row. Taking profit of the Fourier Transform properties, we just keep the first coefficients to represent each row since the most relevant information concentrates in the low frequency components of the sequence. Moreover, as we are working with omnidirectional images, when the Fourier Transform of each row is computed, another very interesting property appears: rotational invariance. Due to the fact that the rotation of a panoramic image is represented as a shift of its columns, the Fourier Transform component's module will be the same. So, the amplitude of the transforms is the same as the original, and just the phase changes. Therefore, we can find out the relative rotation of two images by comparing its Fourier coefficient phases.

2.2 PCA over Fourier Signature

PCA-based techniques have proved to be a very useful compressing methods. They make possible that, having a set of N images with M pixels each, $\vec{x}^j \in \Re^{Mx1}, j = 1 \ldots N$, we could transform each image 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, $k \leq N$. However, if we apply PCA directly over the matrix that contains the images, we obtain a database with information just with the orientation of the robot when capturing those images but not for other possible orientations. What we propose in this point is to transform the Fourier Signature components instead of the image, obtaining the compression of rotational invariant information, joining the advantages of PCA and Fourier techniques.

2.3 Histogram of Oriented Gradient

The Histogram of Oriented Gradient descriptors (HOG) (Dalal and Triggs, 2005) are based on the orientation of the gradient in local areas of an image. Basically it consist in computing the orientation binning of the image by dividing it in cells, and creating the histogram of each cell, obtaining module and orientation of each pixel. The histogram is computed based on the gradient orientation of the pixels within the cell, weighted with the corresponding module value. An omnidirectional image contains the same pixels in a row although the image is rotated, but in a different order. So, if we calculate the histogram of cells with the same width as the image, we obtain an array of rotational invariant characteristics.

However, to know the relative orientation between two rotated images vertical windows are used, with the same height of the window, being able to vary its width and application distance. Ordering the histograms of these windows in a different way, we obtain the same results as calculating the histogram of a rotated image with an angle proportional to the distance between windows. That also will determine the accuracy in orientation computation.

3 LOCALIZATION AND ORIENTATION RECOVERING

In this section, we measure the goodness of each algorithm by assessing the results of calculating the pose of the robot with a new image compared to a map created previously. All the functions and simulations have been made using Matlab R2008b under Max OS X. The maps have been made up of images belonging to a database got from Technique Faculty of Bielefeld University (Moeller et al., 2007). They were collected in three living spaces under realistic illumination conditions. All of them are structured in a 10x10 cm rectangular grid. The images were captured with an omnidirectional camera, and later converted into panoramic ones with 41x256 pixel size. The number of images that compose the database varies depending on the experiment, since, in order to assess the robustness of the algorithms, the distance between the images of the grid we take will be expanded. In the results shown in this paper, the grid used is 20x20cm, with 204 images.

The test images used to carry out the experiments is made up of all the available images in the database, with 15 artificial rotations of each one (every 22.5°). 11,936 images altogether . Because the pose includes the position and orientation of the robot, both are studied separately. Position is studied with recall and precision measurement (Gil et al., 2009). Each chart shows the information about if a correct location is in the Nearest Neighbour (N.N.), i. e., if it is the first result selected, or between Second or Third Nearest Neighbours (S.N.N or T.N.N). Regarding the rotation, we represent the results accuracy in bar graphs depending on how much they differ from the correct ones. If the experiment error is bigger than ±10 degrees, it is considered as a fail and not taken into account.

3.1 Fourier Signature Technique

The map obtained with Fourier Signature is represented with two matrices: the module and the phase of the Fourier Coefficients. With the module matrix we can estimate the position of the robot by calculating the Euclidean distance of the power spectrum of that image with the spectra of the map stored, whereas the phase vector associated to the most similar image retrieved is used to compute the orientation of the robot regarding the map created previously.

Figure 1 (a),(b),(c) show recall and precision measures. We can see that when we take more coefficients, the location is better, but there is a limit where it is not interesting to raise the number of elements we take because the results do not improve. The phase accuracy (Figure 1(d)) also improves when more coefficients are used to compute the angle, although is quite constant when we take 8 or more components. It can be stressed that with just 2 components (Figure 1(a)) we have 96 percent accuracy when we study the Nearest Neighbour, and almost 100 percent when we keep the three Nearest Neighbours.

3.2 PCA over Fourier Signature

After applying PCA over Fourier Signature module matrix, we obtain another matrix containing the main eigenvectors selected, and the projection of the map images onto the space made up with that vectors. These are used to calculate the position of the robot. On the other hand, we keep the phase matrix of Fourier Signature directly to estimate the orientation. To know where the robot is, first the Fourier Signature of the current position image must be computed. After selecting the corresponding coefficients of each row, we project the vector of modules onto the eigenspace, and find the most similar image through Euclidean distance. When the position is known, the phase is calculated the same way than when we do not apply PCA since the phase matrix is not modified.

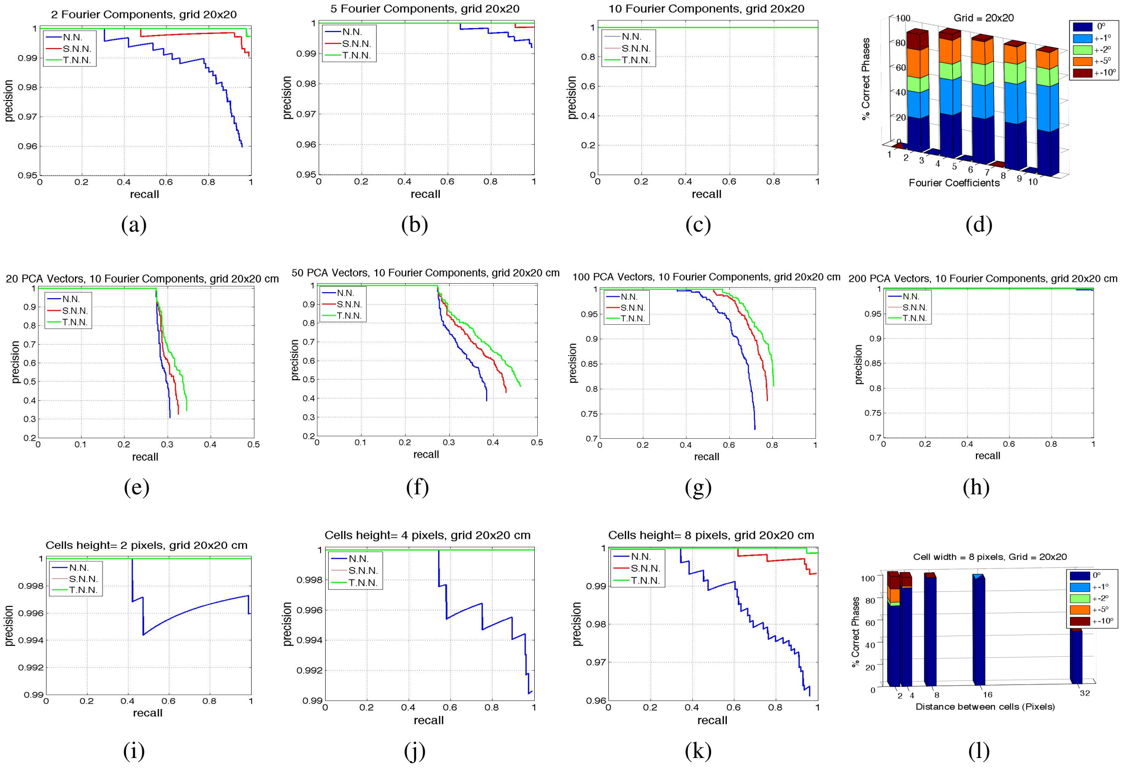
As we can see in Figure 1(e),(f),(g),(h), if we are looking for a high accuracy in the localization task, it is required a high number of PCA eigenvectors, what means loosing the advantages of applying this method. Moreover, in the majority of the experiments, the number of Fourier coefficients we need is bigger than when we do not use PCA, incrementing the memory used. Phase results are not included because the results are exactly the same as showed in Figure 1(d) since its calculation method does not vary.

3.3 Histogram of Oriented Gradient

When a new image arrives, we need to calculate its histogram of oriented gradient using cells with the same size of those we used to build the map. So, the time needed to find the pose of the robot varies depending on both vertical and horizontal cells we use. To find the location of the robot the horizontal cell information is used, whereas to compute the phase we need the vertical cells. In both cases, the information is found by calculating the Euclidean distance between the histogram of the new image and the stored ones in the map. The recall-precision charts (Figure 1(i),(j),(k)) shows that the more windows to divide the image, the better accuracy we obtain. However, it is not a notably difference between the cases. Regarding the orientation (fig 1(l)), although the results are good, it can be stressed that, when the window application distance is greater than 2 pixels, the results are like binary variables, appearing just cases with zero gap, or failures, which is to say that the error is zero or greater than 10 degrees.

4 CONCLUSIONS

This work has focused on the comparison of different appearance-based algorithms applied to the creation of a dense map of a real environment, using omnidirectional images. We have presented three different methods to compress the information in the map. All of them have demonstrated to be valid to carry out the estimation of the pose of a robot inside the map. Fourier Signature has proved to be the most efficient method since taking few components per row we obtain good results. No advantages have been found in applying PCA to the Fourier signature, since in order to have good results it is needed to keep the great majority of the eigenvectors obtained and more Fourier coefficients. In both cases the orientation accuracy depends just on the number of Fourier components, and the error in its estimation is less than or equal to 5 degrees is the great majority of simulations. Regarding HOG, results demonstrate it is a robust method in localization task, having slightly worse results than Fourier algorithm ones. However the orientations computing is less effective due to fact that the degrees are sampled depending the number of windows we use, determining that way its accuracy. This paper shows again the wide range of possibilities of appearance-based methods applied to mo-



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

Figure 1: (a), (b), (c) Recall-Precision charts with N.N., S.N.N. and T.N.N and (d) phase accuracy using Fourier Signature varying number of components. (e), (f), (g), (h) Recall-Precision charts with N.N., S.N.N. and T.N.N using PCA over Fourier Signature varying number of PCA vectors. (i), (j), (k) Recall-Precision charts with N.N., S.N.N. and T.N.N and (l) phase accuracy using HOG varying horizontal window's height.

bile robotics, and its promising results encourage us to continue studying them in deep, looking for new available techniques or improving the robustness to illumination changes for them.

ACKNOWLEDGEMENTS

This work has been supported by the Spanish government through the project DPI2007-61197.

REFERENCES

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients fot human detection. In *Proc of the IEEE Conf on Computer Vision and Pattern Recognition, San Diego, USA. Vol. II, pp. 886-893.*

Gil, A., Martinez, O., Ballesta, M., and Reinoso, O. (2009). A comparative evaluation of interest point detectors and local descriptors for visual slam. *Machine Vision and Applications (MVA).*

Jogan, M. and 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.

Krose, B., Bunschoten, R., Hagen, S., Terwijn, B., and Vlassis, N. (2007). Visual homing in enviroments with anisotropic landmark distrubution. In *Autonomous Robots, 23(3), 2007, pp. 231-245.*

Menegatti, E., Maeda, T., and 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.*

Moeller, R., Vardy, A., Kreft, S., and Ruwisch, S. (2007). Visual homing in enviroments with anisotropic landmark distrubution. In *Autonomous Robots, 23(3), 2007, pp. 231-245.*

Paya, L., Fernandez, L., Reinoso, O., Gil, A., and Ubeda, D. (2009). Appearance-based dense maps creation. In *6th Int Conf on Informatics in Control, Automation and Robotics ICINCO 2009*. Ed. INTICC PRESS - pp. 250-255.

Thrun, S. (2003). Robotic mapping: A survey, in exploring artificial intelligence. In *The New Milenium, pp. 1-35.* Morgan Kaufmann Publishers, San Francisco, USA.

SELF DEPLOYED ROBOTIC NETWORK FOR LONG RANGE SEMIAUTOMATIC OPERATION

Robotics Network for Distance Data Connection, Areal Signal Connection Coverage or Areal Data Acquisition

Tomas Solarski, David Vala and Jiri Koziorek
Department of Measurement and Control, VŠB-Technical University of Ostrava, 17. Listopadu 15, Ostrava, Czech Republic
{tomas.solarski, david.vala, jiri.koziorek}@vsb.cz

Keywords: Robot, Network, Communication, Data, Acquisition, Sensor, Control, Motion.

Abstract: This paper covers questions about long distance communication in a difficult hazardous environment. Distant communication is presented via number of robotics carriers determined to link communication between a centre of control and a remote controlled deployed robot in hostile area or gather data from certain area. The robotic network is autonomous cooperative system of vehicles to provide data connection for the tele-operated deployed robot for example in rescue mission. This robotic network acts autonomously and reacts with surroundings background to guarantee data connection. Two main issues are presented: the data communication and the robotic carrier. Communication can be created by own wireless system carried on robots or use accessible communication such Wi-Fi/Ethernet in urban areas to use the installed networks in buildings to increase a capability of network. The robotics carriers will be realized as modular system with capability to modify each main part of carrier to fit specific environment. Basic construction of the robotic carrier is traction unit with basic frame with motor(s) and lithium based batteries, control unit based on MCU/DSP controller and internal sensor unit with capability to install another set of the external sensors. To provide positions of the carriers to on operator a visualisation of their position is planned by Goole Earth like application.

1 INTRODUCTION

Nowadays advances in embedded systems computations and communication technologies provide support for the cooperative multivehicle systems – mobile robots.

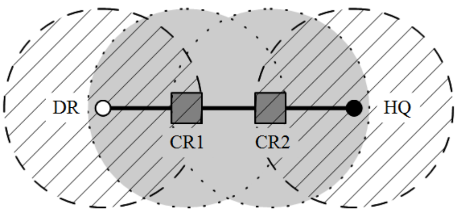


Figure 1: Extended communication by two repeaters.

Mobile robotics are part of our nowadays life, there is an amount of usage of them in explicit areas. Part of robots is designed to work in difficult areas dangerous to human. Environments like areas after or under natural (flood, wildfire) or industrial (chemical, mine) disasters could cause a human injury or even lost of life. They are available only for specific robotic unit equipped by specific tools, sensors or both to perform a search and rescue mission or gather data of the situation.

Employment of such robot can be complicated due to limitations of signal connection, to obtain an extended range. The communication problem can be caused by density of urban areas, difficult mountain landscape, complicated shaft net in mines, radio power limitations etc. Solving extended communication range from control centre to deployed robot can be via system of mobile signal repeaters carried on auxiliary robots (robot carrier).

Robot carriers can act autonomously and perform several nets topology to secure connection to deployed robot or perform data acquisition from area depended on sensor equipment on the robot carrier and its count. Figure 1 shows simple line topology to extend radio communication distance between control centre (HQ) and deployed robot (DR) by two robots carrying wireless repeaters (CR).

Robot carrier himself is completed like multipurpose kit containing several independent units to rebuild carrier and fit it to specific area. Carrier kit contains a wireless communication unit, a control unit and a traction unit (chassis), optionally a sensor unit to gather interest data. Each robotic carrier's unit will act like a distributed system with control subsystem secondary to control unit.

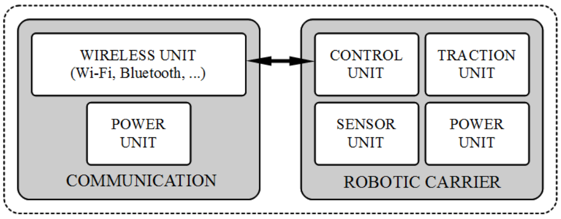


Figure 2: Block scheme of robot.

2 COMMUNICATION

There are many important parts and tasks to develop in this project. One of the most important parts is to develop open radio network for communication between service robot(s) and base station. Because usual teleoperator or a service robot use several communication channel to data exchange separated by purpose or origin of data (camera / video, control, telemetry, payload, etc).

To avoid collisions with this amount of data interface with different, frequency, bandwidth, data throughput rate, range and so on, we establish common interface to all robot internal system as well as transferred data to Ethernet. It causes that you can connect any existing or future device directly or through simple bridge whereas most of them are commercial accessible and also has not any influence to existing communication inside robots system as e.g. the control system and its peripheries. There is possible to transfer control command to the communication unit, telemetry data as well as multiple real time still or motion picture in this solution.

This kind of plug and play system for communication subsystem allows to mount the best suitable communication unit directly before robots mission and to use its all features during operation. As main communication standard we use cheep IEEE802.11b/g system which has sufficient data throughput and range for most of the robot applications. Also the using of modules with GSM/GPRS/EDGE is possible when they are available or robots operate on wide area.

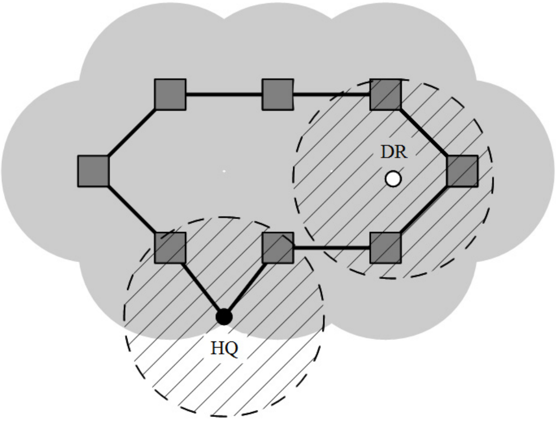


Figure 3: Areal signal covering.

Data exchange with operators or between service robots behind direct radio visibility during rescue and security mission is necessary to be all the time. We design semiautonomous robotics retranslation unit to solve this problem. This robotic mobile repeater can be carried on main robots and deployed in situation when direct communication is unstable or not reliable. There is a new issue to solve – how to organize the radio network to have the most optimal area coverage.

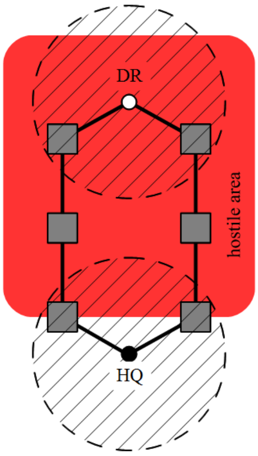


Figure 4: Doubled connection to deployed robot.

We have designed several basic scenarios for testing different type of radio signal inaccessible conditions. This model could be described as line, area, path, and circle coverage of hostile area. And in this model we are looking for best points to deploy the mobile repeater carriers. Advantage of repeater mobility could be exploited when surrounding condition are changed or is necessary adapted network topology. Higher mobility and climb ability in e.g. "Leg's" version of repeater allows finding better position to retransmitting data from source to destination. Some variation of scenario describes backup communication line with dual line of repeater. This is done for safety reason when noise, lost signal or failure causes a termination of one repeater carrier.

Second task is long term operation of service robots in wide area in cooperation with several mobiles robots and fixed station. This mode describes coverage of area with size over units or tens of square kilometres equipped of partial working infrastructure like Wi-Fi, GSM or private radio network. Robots communicate trough real dynamic reconfigurable heterogeneous network over TCP/IP protocol in this task. However it seems that small mobile repeater have no advantage and place in this scenario, relative small and cheap mobile robot carrier with cooperation with main services robots can keep well coverage of wide area radio network. Mobility of repeater carrier can help them to survive difficult conditions. Sensor equipped mobile repeater also provides support to environmental analyses not only useful for routing option.

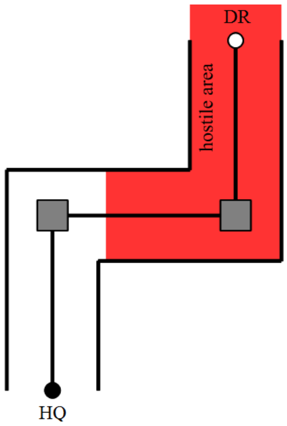


Figure 5: Using of two carried repeaters in urban or mine area.

3 ROBOTIC CARRIER

To provide a positional layout of signal repeaters, autonomous carriers are needed. In our case we will work with solution based on service robotics modular system. Modular means that key parts of the robotic carrier are interchangeable to fit specific area of interest.

Main idea is to split solution of robotic carrier into several units and solve them separately. Basic robot can be divided into units: traction, power supply, control and sensor. The traction is mostly entire mechanical problem due to containing frame (chassis) of robot. The chassis is depending on the traction system. Several traction solutions can be used mainly for indoor or outdoor purpose.

3.1 Traction Unit

By the traction unit we understand a subsystem of robot that provides movements and creates a frame for other parts of the robot. The traction unit contains: frame, gearing, motor(s) and driver (electric), eventually battery.

Simplified block scheme on Figure 6 shows a main part of the traction unit including a battery. Due to a lot of robot's construction possibilities the selection of battery is depended on traction. Nowadays we have to use battery pack based on Lithium due to very good power to weigh ratio and also a maximum current capability, like Li-Pol and others. Size of traction is related to battery capacity and nominal voltage (number of cells).

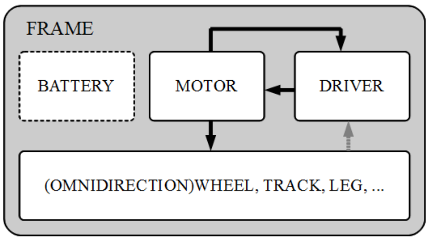


Figure 6: Block scheme of traction unit.

The frame of the robotic carrier is basic construction set for entire robotic system. It has to be light and solid and also suitable for areas of interest. Most used traction on robots nowadays is wheels and tracks (tank under cart). Wheels are very simple to control and robot can move in high speed. On the other hand tracks are slow but robust and can pass terrain with more obstacles.

There are several main environments which influence selection of suitable gears for robot. Generally there are indoor and outdoor. By outdoor we mean only land terrain.

Indoor conditions are represented by short distances and flat surfaces where the main obstacle is stairs. Capability to pass stairs can split indoors robots into two areas. The stairs passing robots have more complicated construction and they are similar as the outdoors robots. Indoor environment provide ideal surface to movement of robot with special traction based on omni directional wheel or walking robots. The omni-wheel is a segmented wheel with capability of two axis movement. The robots based on the omni-wheel can contain common wheels (two) and be completed with omnidirectional (third) to improve turning of the robot. Also a fully omnidirectional equipped robot can be possible, for example a three wheel robot. Movement of this kind of traction is by a various speed of omni-wheels, so robot can change its direction based on the actual speed of omni-wheels.

The indoor environment can be also very good for walking robots without special abilities to control balance on inclined or another complicated surfaces. Segment of walking robots are humanoids - but there are problems with balance and speed of them and so there are not very suitable. More useful robotic traction is based on spider like chassis – 6 or 8 legs. This traction unit provides stability without a complicated balancing system but still has limited speed due to complicated leg's movement. Also a 6 or 8 legs robots have to be equipped with a number of servomechanisms regardless to number of joints on each leg.

Pure walking robots are designated to indoors applications but there can be hybrid solution based on combination of wheel and leg. This solution provides a speed from wheels and agility from legs.

The outdoor conditions are more complicated and for the purpose of robotic network are more probable. Outdoor applications could be majority for robot's movement. Suitable traction for outdoor application is wheels (like in car) and tracks (tank). This kind of traction is very suitable for terrain, roads even in buildings and stairs. Outdoor designed robots can easily be used also in buildings even to climb stairs.

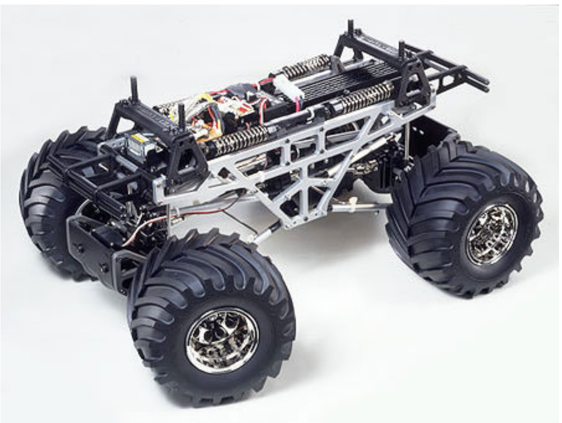


Figure 7: Tamiya TXT-1.

Regarding to high efficiency and speed of wheel based traction this kind of robot is very suited for outdoor application and to carry a wireless connection unit. Figure 7 presents a system from RC car which is very good for outdoor robot. This traction provides two DC motors for driving all four wheels (4x4) on entirely suspended base. Also both axletrees are steered. Tires are adjusted to work in complicated terrain.

Movement of the traction is based on electric motor of any kind. There is also gas-engine where endurance is enlarged, but this has to be more serviced and control of gas-engine is more complicated, so we will work only with DC or EC motor.

Part of traction unit is a driver for electric motor. Construction of driver is very similar even we have DC or BLDC motor. The driver is based on circuits called half-bridges and due to the DC or BLDC there are two or three half-bridges in driver. Half-bride is junction of two power switches which can amplify logic signal to provide power supply battery level and can provide required current. In mobile robotics are used almost only MOSFET based constructions. Advantages of MOSFET are very low power losses and high frequencies of work with minimum of driving energy. N-MOSFET work like ideal power switch and is widely used in power electronics like motor drivers with low voltage (<100V). The driver will be equipped with his own control to provide close loop control of motor. Based on traction system there can be feedback from sensor on shaft of motor or on a wheel, also sensors on legs etc. Control system in traction, based on MCU will communicate with his superior the Control unit of the carrier.

3.2 Power Supply Unit

Power supply or battery for the robotic carrier is very depended on the traction unit. The indoor and outdoor application may be very separate of each other in range and online time and also in a size/weight of robot. The battery pack will be then different. So we can think about including power supply to traction unit and have it together in kit. In this paper we will be talking about power supply unit as separate part of the system. But in the robotic carrier kit will be part of traction.

Batteries which can be used in mobile robotics are basically based on Lithium. Thinking about NiMH, SLA or even NiCd is obsolete. Advantage of Lithium based cells is power to weigh ratio and high current capability (Li-Pol). Also a minimum looses of energy stored that make robot less needed of service. Main disadvantage on Li-xxx is precise charging and watch on discharging to not outrun a minimum voltage limits. Due to this we need to precise charging with balancing/limiting system and when used (discharged) a system to count energy and estimate remaining capacity (to stop discharging in right time) and also watch limits especially minimum voltage per cell.

Power supply unit then contains a basics subsystems like: battery cells, protective circuits and fuel gauge. The protective systems are used in charging of series of cells and balance variance of each cell (capacity is not ideally similar, one cell is fully charged faster than another). Fuel gauge system based on MCU counts energy stored in cells and estimate remaining energy which can be provided. Method is based on simple current and time measurement and it can be determined a quantum of current which flows from battery to load (motor). Also current when charging can be measured (will be negative, current flows back) but there is a problem with protective systems because basic protective system change redundant charging energy into heat.

In mobile robots for outdoor application we can estimate a 7 to 10 cell needed to provide power supply up to 30-42V with continuous discharge current of 10 to 15A. This make an output power in range 300 to 600W which is ideal for mobile wheel based robot with traction based on system from Figure 7. For the indoor robot same cell can be used but only two or three will be needed.

Power supply unit is slave to the control unit (master on the bus) and contain separate control to measure capacity of power cell(s) and can estimate and report it to control unit.

3.3 Control Unit

Main unit in robotic carrier is the control unit. This unit have to decide how to ensure data connection, how to get to position, to drive traction unit and to communicate with others carriers and to establish connection from control centre to deployed robot.

Entire control of robotic carrier is distributed because all of units have their own control by MCU. The control unit act like a master on bus via which all MCUs are connected together. Preferred bus on such mobile robot is LIN, RS485 or even CAN based on required speed and data security.

The control unit depends on complexity of needed solution from algorithms. Based data are gained from sensor unit and from communication. The control unit calculates a position to move and control movement by sending required speed to closed loop control of motor in traction unit. Basically this can be covered by some newer MCU with 32 bit core to obtain a real time solved problems and results.

Power supply to control unit is from power unit, so there is a power supply shared with motor on the other hand a wireless unit have its own because primary function is to provide data connection.

3.4 Sensor Unit

In mobile robotics we can split sensors into two groups. Internal sensors are used to ensure a basic function of the robot. Typically internal sensors are incremental rotary sensors in motor (traction unit) or sense resistors in battery (power unit). These sensors are used to operate a basic system of robot. In the other hand external sensors are used to contact robot with his exterior, robot can operate self without it and this external sensors determine the basic type and purpose of the robot.

Internal sensors are then on unit which control basic function of the robot. Sensor unit contains external sensors to gather data from surrounded area of robot. Because a primary function of robot is to carry a wireless system and establish connection to deployed robot a sensor unit is basically equipped with sensors to obtain a global position of robot and to provide them to control unit to establish right network topology. Also safety ability has to be implemented to avoid contact robot with obstacle or to do not injure humans. Basically, the robot needs a set of tactile sensors to avoid obstacles. Tactile sensor is contact or contactless, where the simplest contact tactile sensor is a switch, but this is not very useful because a short range of operation. More useful sensors are contactless based on ultrasonic sensors or infrared optical sensor, both work like range finder so we can find an obstacle and also we can estimate range of it.

In indoor application of carrier a network topology can be determined basically on odometer (data from traction unit). Indoor due to short ranges and covering sky a global data cannot be provided and precise admeasurements have to be used. So there we need to know a start position of each carrier to work with this offset of position.

Additional sensors can be used to gather more data from area where carrier robot is used. This will work as additional function and area where are robotics carriers deployed can be better monitored than only with deployed robot. There can be used simplest sensors to measure temperature, pressure sound waves, chemical materials in air etc.

4 VISUALIZATION

The entire system needs a visualization system to provide human-machine interface, to simplify using and maintenance. Each system has specifics demands for visualization and for system described above is very important except of standard telemetry data also position data of each unit as well as give to operator overview and knowledge of terrain (etc.) where robots operate. For this purpose is useful include part of geographic information system to operator control panel to access terrain map data. It could be dynamically refilled and updated by e.g. aerial photography, laser scan or other robots sensor data or data from mobile repeater such as temperature, chemical sensor or camera.

"Wide are" scenario has huge requirement for GIS data source which could be very expensive. For this reason we are use public GIS source like Google Earth. This makes possible to visualize robots, repeaters and other position in 3D maps and terrain model. Big advantage of this system is possibility dynamic updating visualized data. It simplifies maintenance of whole system and allows online publishing of e.g. environmental data for community or government usage; depend on mission and devices type.

5 CONCLUSIONS

This work on mobile robotic network is basically in phase of prototype development. Presented information will be taken under test and will show us possibilities of next development and serves us as guide to how to continue on this project. Basic idea was to enlarge the range of wireless system to our robot. Nowadays we have knowledge on constructing mobile robots and this work is based on prototypes of them. We believe that our work will provide a functional prototype of robotic network to use in emergency and rescue operations.

Robotics system has also feature to join several branches of technical development. Traction is based on mechanical engineering, network is based on communication solutions, sensors and control unit need to be solved by electronics and finally visualization is based on computer programming.

ACKNOWLEDGEMENTS

This work is supported by grant of the Grant Agency of the Czech Republic GA102/08/1429 - Safety and security of networked embedded system applications.

REFERENCES

Cruz, D., McClintock, J., Perteet, B., Orqueda, O., Cao, Y., Fierro, R., 2007. Decentralized cooperative Control – A multivehicle platform for research in networked embedded systems. In *IEEE Control Systems Magazine*, [internet] June. Available at : http://www.ece.unm.edu/faculty/rfierro/papers/2007/ Marhes_CSM-Published.pdf [Accessed 12 December 2009].

Sanfeliu, A., Hagita, N., Saffiotti, A., 2008. Network robot systems. In *Robotics and Autonomous Systems*, [internet]. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.140.558&rep=rep1&type=pdf [Accessed 15 December 2009]

Kárník, L., 2002. Lokomoční ústrojí mobilních robotů pro nestrojírenské aplikace. In *Automa*, [internet] July. Available at: http://www.odbornecasopisy.cz/index.php?id_document=28494 [Accesed 18 December 2009]

Solarski, T., Musil, K., Janckulík, D., Vala, D., 2008. Robotic platform remote controled by enhanced wireless technology. In *8th International Scientific – Technical Conference PROCESS CONTROL 2008*. ISBN 978-80-7395-077-4

Stankovič, J., Vala, D., 2008. Data Acquisition by heterogeneous mesh networks in early warning systems. In *8th International Scientific – Technical Conference PROCESS CONTROL 2008*. ISBN 978-80-7395-077-4

ROBOT SOCCER STRATEGY – BIOMIMETIC APPROACH

Nicu George Bîzdoacă, Coman Daniela
University of Craiova, Craiova, Romania,
nicu@robotics.ucv.ro, amdcoman@yahoo.com

Hani Hamdan
SUPELEC, Department of Signal Processing and Electronic Systems, Gif Sur Yvette, France
hani.hamdan@supelec.fr

Khalid Al Mutib
King Saud University, College of Computer and Information Sciences, Riyadh, Kingdom of Saudi Arabia
muteb@ksu.edu.sa

Keywords: Robotics, Biomimetic Approach, Control Strategy, Cooperative Tasks.

Abstract: Biomimicry is the right pattern for a successful algorithm or control strategy. The present paper applies biomimetic approach to robotic soccer game, using as model the human soccer team's strategy. For implementation the authors use a team of MIROSOT soccer robot, with three robots. Using software Simi Scout a suitable tactics analysis can be extract from the games. Analyzing the real soccer game, a number of attributes are specified and are embedded at different levels. The specified levels are interconnected and the game analysis is processed for optimization. The authors approach offer a flexible and more complex situation: all the robots can be attacking, defending or goalkeeper player. In this case the role changing strategy is based on the distance between player and the ball, the payer and the goal area and the player and the attack area. The controller for the attacking robot is designed using Petri nets. The Petri net model is implemented in Petri Net Toolbox under MATLAB environment. Using this information the robot program is adapted and the tests/games are experimented. The results are commented and improved control architecture, based on practical results, is proposed.

1 INTRODUCTION

The nature's inventions have inspired researchers in developing effective algorithms, methods, materials, processes, structures, tools, mechanisms, and systems. Biomimetics is a new multidisciplinary domain that includes not only the uses of animal-like robots – biomimetic robots as tools for biologists studying animal behavior and as research frame for the study and evaluation of biological algorithms and applications of these algorithms in civil engineering, robotics, aeronautics. Multi-agent system has emerged as a subfield of AI and helps to understand and provide with theory and principles for constructing complex systems with multiple agents and their coordination/competition in dynamical environments. The robot soccer game is an interesting benchmark problem for the multi-agent systems.

Generally spoken in robot soccer regarding the division of labor between the components of a soccer team, namely between the host computer system and the autonomous mobile robots, three system configurations are defined: 1. Remote brainless system; 2. Robot-based system; 3. Vision-based system.

The vision-based system can be described as the step from the remote brainless to the robot based system, as some of the intelligence is transferred from the main computer to the single agents, but the control of the vision system and the strategic coordination still remain tasks of the host unit.

The Micro-Robot' World Cup Soccer Tournament (MiroSot) is the brainchild of Jong-Hwan Kim (Coman, 2008) of KAIST, Korea and was initiated in 1995. A mobile robot soccer team consists of up to three micro mobile robots. Two teams play soccer according to the rules similar in

nature to the real soccer game. The Federation of International Robot Soccer Association (FIRA) has established these rules.
The mechanical construction of the micro mobile robot is based on a duralumin frame, on which two DC motors with gear boxes and a controller board are mounted.

Robot has two parallel wheels and in addition is suspended by two slipper elements mounted at the front and back part of the frame.

2 POLLING ROBOT SOCCER STRATEGY FROM THE REAL SOCCER GAME

The soccer robot game structure has 6 steps:

Step 1. Image acquisition and primary calculation (distance, velocity, relative angle calculation etc.)

Step 2. Decide which posture is suitable for player offensive or defensive.

Step 3. Determination of team strategy and player profile assignment.

Step 4. Determination of the target position.

Step 5. Path planning.

Step 6. Calculation of the wheels' moving direction, velocity and displacement.

In order to have a successful team the secret are: good player's profile, good team strategy.

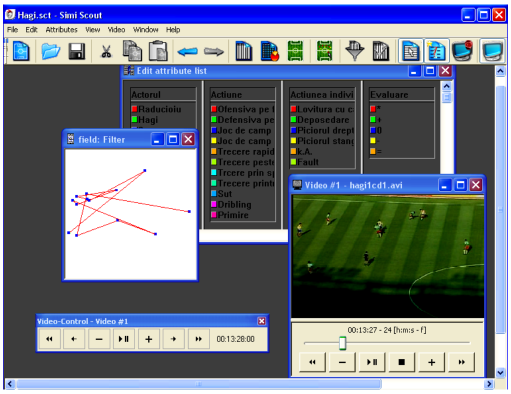
The player profiles are:

- Attack
- Midfield
- Defender
- Goalkeeper

Using Simi Scout a real game is analyzed and the player profile is extracted (for this example is used Romania Columbia, 1994, soccer game, and for players profile, is used Romanian player profiles).

Using the real game information, the player situations can be extracted. Let's analyse , in detail, the strategy for **Attack player** (Bîzdoacă et al., 2005)**.**

Assuming that the left half of the playground is the opponent side, it is reasonable that the attacking player must move to the right side of the ball as soon as possible.



Actorul	Actiune	Actiunea indivi	Evaluare
🟥Raducioiu	🟥Ofensiva pe t	🟥Lovitura cu c	🟥*
🟩Hagi	🟩Defensiva pe	🟩Deposedare	🟩+
	🟦Joc de camp	🟦Piciorul drept	🟦0
	🟨Joc de camp	🟨Piciorul stang	🟨-
	🟧Trecere rapid	🟧k.A.	🟧=
	🟩Trecere peste	🟩Fault	
	🟦Trcere prin s		
	🟩Trecere print		
	🟦Sut		
	🟪Dribling		
	🟪Primire		

Figure 1: SimiScout analise for soccer game Romania-Columbia.

3 PETRI NET MODEL FOR ATTACKING SOCCER ROBOT

Petri nets, developed by Carl Adam Petri in his Ph.D. thesis in 1962, are generally considered as a tool for studying and modeling of systems. A Petri net is foremostly a mathematical description, but it is also a visual or graphical representation of a system. A Petri net can be analysed in order to reveal important information about the structure and behaviour of the modeled system. The information may for instance suggest improvements to the system.

A Petri Net consists of a number of places and transitions with tokens distributed over places. Arcs are used to connect transitions and places. When every input place of a transition contains a token, the transition is enabled and may fire. The result of firing a transition is that a token from every input place is consumed and a token is placed into every output place. The firing of transitions represents causality and inferencing relations. Petri nets have been generalized by allowing multiple token arcs, inhibitor arcs, place capacity, colored tokens etc. (Peterson, 1981).

Formally, the structure of a Petri net, defined by its places, transitions, input function and output function (Coman, 2008), (Kim et al., 2004).

The Petri net model (Kim et al., 2004) for the attacking soccer robot modeling the real situation, has been designed so that the topology to contain the following situations in which the attacking robot can be, as follows:

a) The attacking robot is behind the ball;
b) The attacking robot kicks the ball;
c) The attacking robot is in offside position;
d) The attacking robot is in contact with ball and

the ball is situated behind the robot.

In state (a) the attacking robot is in a probable position to kick, in state (b) it is kicking the ball, in state (c) it is in front of ball, so should be careful to avoid the offside position, and in state (d) it is in contact with ball.

With these four states, the Petri-net for the attacking robot controller is formed. “Angle” is used to refer to the angle between the heading direction of the attacking robot and ball. “Distance” is used to refer to the distance between the attacking robot and ball in pixels. In state (a), the attacking robot is ordered to move to ball and kick it. In state (b), if “Angle” is above 45° , or “Distance” is more than 20 pixels, the attacking defense robot goes to state (a). In state (a), if the ball is on the right side of attacking robot (offside position), it should go to state (c). In state (b), if the attacking robot fails to kick the ball, the robot goes to state (c). In state (c), the attacking controller orders the robot to move sideways and it comes behind the ball without touching it and goes to state (a). In state (c), if “Distance” is below 10 pixels, the robot goes to state (d), so it should move away from the ball till “Distance” is above 20 pixels and then can get into state (c).

Analyzing the situations for Goalkeeper player, for Defender and for Midfield we can generalize the situations as:

p_1) The robot is behind the ball;

p_2) The robot kicks the ball in the opponent direction;

p_3) The robot is in unwanted position;

p_4) The robot is in contact with ball and the ball is situated behind the robot.

Using Simi Scout a set of attributes are assured. Statistically, after analyzing the all attributes proposed, only 7 are retained and become transition for our algorithm:

t_1. Tries to kick the ball, though it is not in a good position to kick,

t_2. In front of the ball and at the following instant it is in a good position to kick,

t_3. In front of the ball, and moving to an unwanted position,

t_4. In unwanted position and escaping from that,

t_5. Misses the ball, and is in front of the ball,

t_6. In unwanted position, and then in contact with the ball and behind, and

t_7. Away from the ball and behind, but still in an unwanted position.

The incidence matrix has rank 3, so that the Petri net model shown in Fig. 2 will have:

m - rank $(A) = 1 \Rightarrow$ at most 1 P-invariants are linearly independent,

$n - \text{rank}(A) = 4 \Rightarrow$ at most 4 T-invariants are liniarly independent.

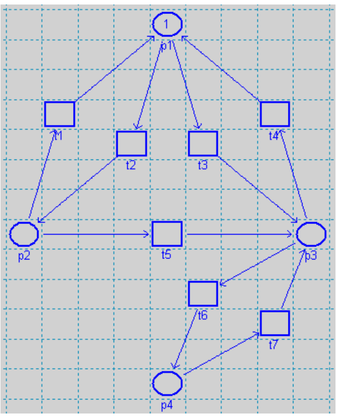


Figure 2: Petri net model for the soccer robot.

As every robot has the same strategy, the role for attack, defender, midfield or attack can be easily switched between the robots. The roll will be assigned dynamically, be image analyze.

A role assigned it is means that the intervention area of the robot is in:

- Attack Area
- Midfield Area
- Defense Area
- Goalkeeper Area

The ball position will switch the team strategy.

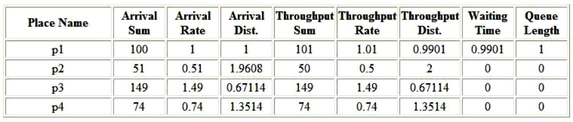
The team strategy extracted (from Romania – Columbia 1994 - soccer game - using Simi Scout software) is:

- If the ball is in attack area then 2 player are attack player, and the nearest player to midfield area is midfield
- If the ball is in the midfield area, then 2 player are midfield player, and the nearest player to defense area is defender
- If the ball is in defense area, then 2 players are defense player, and the nearest player to goal area is goalkeeper.

5 NUMERICAL SIMULATION AND EXPERIMENTS

The results from the mathematical method of checking through the invariants method associated transitions and the corresponding positions after calculating the incidence matrix of the net have been validated through the simulations using Petri Net Toolbox in Matlab environment.

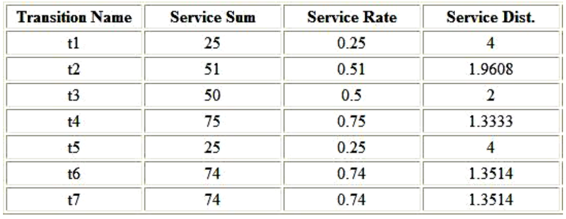
Table 1: Global Statistics Places.



Place Name	Arrival Sum	Arrival Rate	Arrival Dist.	Throughput Sum	Throughput Rate	Throughput Dist.	Waiting Time	Queue Length
p1	100	1	1	101	1.01	0.9901	0.9901	1
p2	51	0.51	1.9608	50	0.5	2	0	0
p3	149	1.49	0.67114	149	1.49	0.67114	0	0
p4	74	0.74	1.3514	74	0.74	1.3514	0	0

It was validated in this way that the net topology, the evolution of (their dynamics), as well as structural and behavioral properties. The following two tables (Table 1- Global Statistics Places and Table 2 - Global Statistics Transitions) present the complete lists of global indices associated with the places and the transitions considered in the architecture of Petri net that modeling the controller for attacking robot.

Table 2: Global Statistics Transitions.



Transition Name	Service Sum	Service Rate	Service Dist.
t1	25	0.25	4
t2	51	0.51	1.9608
t3	50	0.5	2
t4	75	0.75	1.3333
t5	25	0.25	4
t6	74	0.74	1.3514
t7	74	0.74	1.3514

Also, the special options of Petri Net Toolbox, which confers a high capacity of analysis, has made possible a synthesis of this Petri net model which allows exploring the dependences of global performance indicators associated with the net positions/transitions on two "Design Parameters".

6 CONCLUSIONS

The research proposes a biomimetic algorithm based on analyzing the real situation (Peterson, 1981). In this paper, a Petri net model is used for designing a low level controller for the soccer robots. The presented controller did not use the information of opponent team. A future development of this research will add the opponent predicted strategy and a more dynamical switching strategy. Using a single model robot controller offer advantages in terms of implementation, but the team strategy can be easily indentified by the opponent and annihilated. The Petri net model presented is implemented in MATLAB environment. The simulations studies was validated that the net topology, the evolution of (their dynamics), as well as structural and behavioral properties and was provided the global performance indices associated with the places and the transitions and also the whole set of global indices associated with all the nodes of the net. Finally, the feasibility of the proposed architecture is demonstrated by the experimental results.

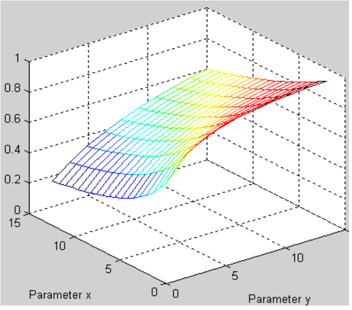


Figure 3: Dependency of the *Queue length* index associated p_1 position.

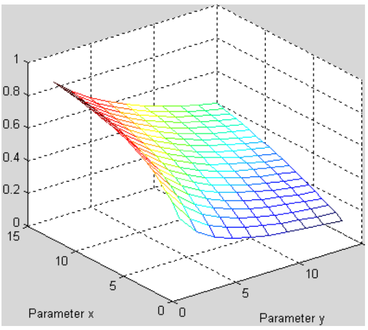


Figure 4: Dependency of the *Queue length* index associated p_3 position.

ACKNOWLEDGEMENTS

This report is part of the ROMANIAN NATIONAL UNIVERSITY RESEARCH COUNCIL (CNCSIS) contribution to the project PNCDI – II - 289/2008.

REFERENCES

Coman D..2008. Using Petri Nets in the Soccer Robot Control Architecture, Annals of DAAAM. pp. 295 – 296.

Bîzdoacă N. G., Degertu S., Diaconu I. 2005. Behavior based control for robotics demining. *International Symposium on System Theory, SINTES 12*, ISBN 973-742-148-5, 973-742-152-3, pp. 249-254.

Kim J. H., Kim D. H., KimY.J., Seow K.T.,2004. *Soccer Robotics*, Springer-Verlag.

Peterson J. L.,1981. Petri net theory and modelling of systems, Prentice-Hall Inc.

TOWARDS OBJECT-ORIENTED SOFTWARE DEVELOPMENT FOR INDUSTRIAL ROBOTS[*]

Facilitating the Use of Industrial Robots by Modern Software Engineering

Alwin Hoffmann, Andreas Angerer, Andreas Schierl, Michael Vistein and Wolfgang Reif
Institute for Software & Systems Engineering, University of Augsburg, Augsburg, Germany
{*hoffmann, angerer, schierl, vistein, reif*}*@informatik.uni-augsburg.de*

Keywords: Industrial robotics, Object-oriented software development, Robot programming, Software engineering.

Abstract: Industrial robotics is characterized by sophisticated mechanical components and highly-developed control algorithms. However, the efficient use of robotic systems – with regard to flexibility, reusability and extensibility – is very much limited by existing programming methods. As a consequence, software development for industrial robots is a complex and time-consuming task which generates considerable costs. This work performs an analysis of the current software development for robotics systems and identifies shortcomings from a software engineering point of view. Based on that, it outlines an architectural approach that addresses the identified problems and allows efficient software development for industrial robotic systems.

1 INTRODUCTION

According to ISO standard 8373, an industrial robot is defined as "an automatically controlled, reprogrammable, multipurpose manipulating machine [...] for use in industrial automation applications". Given an appropriate tool, industrial robots are able to perform a large variety of tasks ranging from assembly over welding to quality assurance. But when looking deeper, the use of robots in companies is mostly restricted to mass production with rather easy, recurring tasks. Although small and medium enterprises with their small production batches could benefit from flexible production systems, industrial robots are rarely found there. One major reason for the limited use is the way of adapting robots to perform a task. Today, industrial robots are still programmed with special robot programming languages which have robotic-specific data types, allow the specification of motions, and communicate with external devices and systems via fieldbus (e.g. tools, sensors, or PLCs). Programs written in these languages are executed in special runtime environments by stepwise interpretation of instructions and submitting them to the underlying robot controller.

Due to these low-level programming techniques, developing software for an industrial robot is a complex and tedious task requiring considerable technical expertise. Already in 1990, (Miller and Lennox, 1990) identified the fundamental problems of programming robots which are still relevant in industrial robotics even two decades later. The problems mentioned are e.g. high costs for application development, low reusability of robot-specific source code, and the complexity of extending existing applications with new tasks or devices. Despite these problems, research in software engineering for robotics is mostly targeting experimental robotics (Brugali, 2007), because industrial robotics has been considered an already solved problem for a long time (Hägele et al., 2008).

Hence, this paper analyzes the software development for robotics systems and explains why the current practice has shortcomings from a software engineering point of view (Sect. 2). Subsequently, we briefly present an object-oriented approach that can overcome those shortcoming in Sect. 3 and illustrate the possible benefits in Sect. 4. Finally, conclusions and future work are presented in Sect. 5.

2 PROBLEM ANALYSIS

Software development for commercially available in-

[*] This work presents results of the research project *Soft-Robot* funded by the European Union and the Bavarian government within the *High-Tech-Offensive Bayern*. The project is carried out together with KUKA Roboter GmbH and MRK-Systeme GmbH.

dustrial robots is usually performed using the development tools available from the robot manufacturers, where each manufacturer provides its own system. These systems are mostly based on proprietary, special programming languages which are tailored to the features of the underlying robot controller. Examples are the KUKA Robot Language or RAPID from ABB. These languages are rather limited compared to general-purpose programming languages. For example, there is no built-in support for the development of graphical user interfaces, and interaction with tools and external systems is based on low-level fieldbus communication. On the other hand, these languages focus on the basic features of automation systems and these limitations help ensuring safety and real-time capabilities, which are necessary to achieve precision and repeatability. Based on their basic languages, manufacturers provide programming extensions for special application domains like welding. These extensions contain macros for common application-specific tasks and allow to configure tools.

However, with increasing requirements to the applications, using robot programming languages alone is not sufficient any more. Especially user interaction or complex domain-specific application logic is often realized using a general-purpose programming language, while robot languages are only used for moving the robot and triggering exactly timed tool actions. The interplay between such an application and the robot controller depends on the use case. One approach is to generate robot source code and to transfer it to the robot controller. This approach is often used in off-line programming systems where a virtual model of the robotic workcell is used to program and simulate tasks.

A different approach is to use a pre-defined set of robot programs which can be remotely parameterized and executed by an application. Today, such a mixed application structure is typical for many existing solutions in industrial robotics. For example, (Ge and Yin, 2007) describe the implementation of a plastic injection molding application using the general-purpose programming language C# for the domain logic and the graphical user interface whereas RAPID was only used for commanding the robot. Another example is described by (Pires et al., 2009) where a programming-by-demonstration application is presented. The main application as well as a speech interface were again realized with C#. Although the force-controlled guiding system was directly implemented on the robot controller, it is forwarding its data to the main application for further processing. After having taught a new task by demonstration, robot code is generated and submitted to the controller. As these examples show, it is possible to implement complex domain-specific applications and customized user interfaces on top of existing robot programming environments. However, the result is always an individual solution, requiring great effort and expertise for development and, consequently, causing high costs. Evaluating these solutions from a software engineering point of view shows shortcomings in software quality. While functionality and usability is often increased using such approaches, other quality attributes (i.e. reliability, performance, maintainability, extensibility or portability) are influenced negatively. For example, robot motions that are triggered inside an application must always be backed with appropriate programs on the robot controller. Consequently, both application parts have to be maintained or extended carefully with mutual dependencies in mind. Approaches that rely on automatic generation of robot control code provide a higher level of extensibility (by modifying code generators) and increase portability (with multiple code generators). However, maintainability problems arise when generated code has to be adjusted: In most cases, these adjustments cannot be transferred back to the generating model. Further issues concern performance (especially for code generation) and reliability, as concurrent programs are difficult to predict, test and debug.

Standard business applications are nowadays developed on top of modern frameworks like Microsoft .NET or Java Enterprise Edition. These frameworks offer solutions for many standard problems like communication in distributed systems, persistency and security. Our goal is to show that with a novel approach robotic applications can be developed in the same way as business applications. By providing a robotic framework that already incorporates the required infrastructure solutions, such an approach can increase productivity, reduce costs and weaken the trade-off between functionality and software quality.

3 APPROACH

We have analyzed a broad variety of typical tasks for industrial robots (e.g. welding or palletizing). This analysis has shown that robot programs usually consist of a defined set of real-time critical control actions which are embedded into a high-level work flow. For example, welding a single seam on a workpiece is one real-time critical action consisting of operations of the welding torch during the robot's movement. The work flow of a welding application only coordinates the proper execution of these actions in order to weld

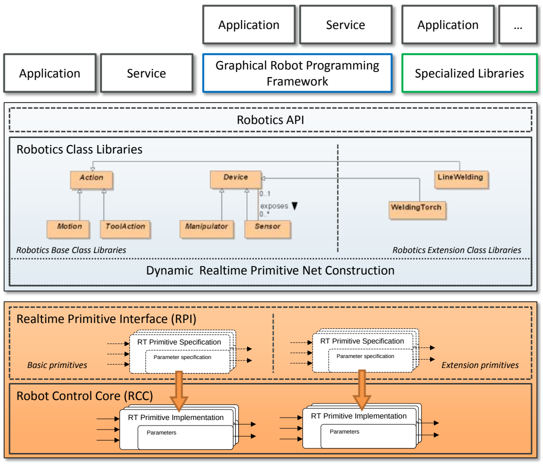


Figure 1: The components of the proposed multi-tier software architecture.

a series of seams. Hence, if it is possible to create an adequate abstraction layer for encapsulating such real-time critical actions, robotic applications can be programmed in any programming language or environment regardless of real-time issues.

Based on this observation, we have developed a multi-tiered software architecture (Hoffmann et al., 2009) which is shown in Figure 1. Core part of this architecture are the *Robotics Base Class Libraries (RBCL)*. They represent an object-oriented framework for robotic applications and provide an abstract class model for the industrial robotics domain as well as the aforementioned abstraction layer for specifying real-time critical actions. The domain model provides objects for common concepts like robots, tools, frames or instructions (e.g. motions or tool actions). They can be extended in order to encapsulate task-specific functionality (e.g. for arc welding) or to add new robots, tools and sensors. All functionality exposed by the RBCL and their extensions (*Robotics Extension Class Libraries*) forms the *Robotics Application Programming Interface (Robotics API)*. This interface is intended to be a broad platform for developers of robot programs. At the moment, there exists an implementations of the RBCL in C# which is able to control KUKA lightweight robots.

Instructions of the Robotics API can either be executed independently of each other, or be combined to larger instruction units by certain operators with defined semantics (e.g. temporal delay, superposition). When executed independently, no hard guarantees can be given concerning timing relations across multiple instructions. When executing complex instructions that have been pre-combined using one of the aforementioned operators, real-time deterministic execution is guaranteed. To achieve this, all instructions are dynamically translated into commands for the *Realtime Primitives Interface (RPI)* and executed by a *Robot Control Core (RCC)*. Such an RPI command consists of a combination of certain, predefined (yet extendable) calculation modules, and a specification of the data flow among them. Implementing only the RCC with real-time aspects in mind is sufficient to allow the atomic execution of Robotics API instructions under real-time conditions.

4 BENEFITS

Using this approach, applications for industrial robots can be mainly developed using standard technologies and environments. For example, a modern general-purpose programming language such as C# and a development environment such as Microsoft Visual Studio can be used. Common functionality of industrial robots is provided through a well-defined and extensible application programming interface (the Robotics API), and can be easily used by application developers. In consequence, they are able to focus on solving domain-specific problems and, as far as possible, do not have to deal with issues like communication and synchronization with the robot control or real-time programming.

Furthermore, the Robotics API provides a comprehensive, abstract model of the robotics domain – including physical objects, interesting points on those objects and ways to define specialized devices (e.g. a welding robot). This framework can be used to formulate certain tasks that are specific to the applications of interest (e.g. welding a line on a workpiece). Basic functionality can be reused, in particular existing mechanisms for encapsulating real-time critical work flows (e.g. moving along a welding line with exactly timed activation of a welding torch). With this abstraction, existing development processes, methods and tools of modern software engineering can be used to construct intuitive domain specific applications, regardless of real-time issues. As a consequence, complex application logic and high usability can be achieved, whereas maintainability, performance and reliability are taken care of by the automatic generation and execution of RPI commands.

Besides basic motion commands, our architecture also supports the integration of new motion types. The fine grained structure of RPI allows a flexible specification of motion control algorithms as e.g. required for sensor integration and compliant motions. The tight integration of high-level motion specification and low-level motion execution, which is achieved by the automatic generation and execution of RPI commands, leads to good maintainability of resulting applications, and real-time constraints can be satisfied by an appropriate implementation of an RPI interpreter, ensuring performance and reliability for the critical system parts.

Traditionally, the communication with external devices and systems is performed using a fieldbus that is connected to a port with analog and digital I/Os. In our approach, the physical I/Os can be mapped to logical units, e.g. the necessary I/Os to drive a gripper can be controlled using a single gripper object. This object encapsulates the properties, i.e. the configuration, as well as the behavior of the tool and can be used for programing. Besides, complex devices can be controlled with real-time performance by introducing specialized RPI modules and corresponding objects for the Robotics API. Furthermore, instructions usually incorporate a two-stage error handling. If an error occurs during the execution of the instruction, basic error handling causes the robot system to reach a stable and secure state. Subsequently, an exception is thrown and a high-level recovery strategy can be executed. Thus, it is possible to guarantee reliable error handling and to provide high-level error mechanisms that developers can use.

There are some tasks which require the use of multiple robots, e.g. mobile manipulation scenarios or lifting large and heavy workpieces. Traditionally, each robot has its own controller and must be programmed individually, using fieldbus communication or special markers to handle synchronization with the other robots. Using our approach, it is possible to develop a single program that controls multiple robots. The robots can be synchronized with real-time performance using special operators offered by the Robotics API. It is even possible to define a logical unit, that consists of multiple robots, but can be programed as a single robot (e.g. for load sharing, or for a mobile manipulator).

5 CONCLUSIONS

This paper presents a new approach for the software development for industrial robots – an approach that is rather radical. Its focus does not originate from mechanical engineering or control theory, but from software engineering. Furthermore, it is not directly compatible with current robot controllers and even requires a new generation of robot control software. However, it shows a way to overcome the intrinsic problems and limitations of current programming environments for industrial robots. The development of this approach is performed in tight cooperation with industrial partners: KUKA Roboter GmbH, a specialist for robotics internals and Europe's leading robot manufacturer, and MRK Systeme GmbH, a system integrator working with KUKA robot and future user of our approach. This cooperation and the joint development of prototypical systems helped greatly in validating the feasibility and industrial meaningfulness of our research.

The presented approach promises advantages on the feature side – as it was designed with extensibility in mind – but the main benefits can be found coping with non-functional requirements. Most notably, usability and maintainability benefit from the advances achieved in the field of software engineering during the last 15 years, which can directly be applied to the robotics domain using our approach. Although the benefits and first results are promising, the approach must demonstrate its advantages in practice. Therefore, future work focuses on creating challenging application examples on top of our approach. This includes the development of a reusable graphical robot programming framework for SMEs and the introducing of service-oriented architectures for robot-based automation processes.

REFERENCES

Brugali, D., editor (2007). *Software Engineering for Experimental Robotics*. Springer Tracts in Advanced Robotics. Springer.

Ge, J. G. and Yin, X. G. (2007). An object oriented robot programming approach in robot served plastic injection molding application. In *Robotic Welding, Intelligence & Automation*, volume 362 of *Lect. Notes in Control & Information Sciences*, pages 91–97. Springer.

Hägele, M., Nilsson, K., and Pires, J. N. (2008). Industrial robotics. In Siciliano, B. and Khatib, O., editors, *Springer Handbook of Robotics*, chapter 42, pages 963–986. Springer, Berlin, Heidelberg.

Hoffmann, A., Angerer, A., Ortmeier, F., Vistein, M., and Reif, W. (2009). Hiding real-time: A new approach for the software development of industrial robots. In *Proc. 2009 IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, St. Louis, USA.

Miller, D. J. and Lennox, R. C. (1990). An object-oriented environment for robot system architectures. In *Proc. 1990 IEEE Intl. Conf. on Robotics and Automation*, pages 352–361, Cincinnati, Ohio, USA.

Pires, J. N., Veiga, G., and Araújo, R. (2009). Programming by demonstration in the coworker scenario for SMEs. *Industrial Robot*, 36(1):73–83.

DYNAMICAL INVARIANTS FOR CPG CONTROL IN AUTONOMOUS ROBOTS

Fernando Herrero-Carrón, Francisco de Borja Rodríguez and Pablo Varona
Grupo de Neurociencia Computacional, Escuela Politécnica Superior, Universidad Autónoma de Madrid
Calle Francisco Tomás y Valiente 11, 28049, Madrid, Spain
{*fernando.herrero, f.rodriguez, pablo.varona*}*@uam.es*

Keywords: Bio-inspired robotics, Central pattern generators.

Abstract: Several studies have shown the usefulness of central pattern generator circuits to control autonomous rhythmic motion in robots. The traditional approach is building CPGs from nonlinear oscillators, adjusting a connectivity matrix and its weights to achieve the desired function. Compared to existing living CPGs, this approach seems still somewhat limited in resources. Living CPGs have a large number of available mechanisms to accomplish their task. The main function of a CPG is ensuring that some constraints regarding rhythmic activity are always kept, surmounting any disturbances from the external environment. We call this constraints the "dynamical invariant" of a CPG. Understanding the underlying biological mechanisms would take the design of robotic CPGs a step further. It would allow us to begin the design with a set of invariants to be preserved. The presence of these invariants will guarantee that, in response to unexpected conditions, an effective motor program will emerge that will perform the expected function, without the need of anticipating every possible scenario. In this paper we discuss how some bio-inspired elements contribute to building up these invariants.

1 INTRODUCTION

Biology has been a source of inspiration for robot constructors for some time now. In particular, many works have built Central Pattern Generator circuits to generate rhythmic motion in a robot (see (Ijspeert, 2008) for a review). Usually CPGs are regarded as a black box dynamical system capable of generating rhythmic patterns, in a way that is flexible enough to accommodate external modulation and/or perturbations, but robust enough to stay close to an effective pattern.

Living CPGs have developed to perform very specific tasks involving rhythmic activity: controlling swimming, chewing, breathing, heart beating, walking, etc (Grillner, 2006). But, not every individual of one same species presents the same physiological structures: due to differences in size, weight, or even history (illnesses, lesions, etc.) CPGs must be able to adapt to different environments (Bucher et al., 2005). Therefore, we can conclude that underlying CPG circuits there are mechanisms that allow for a great degree of flexibility while preserving its core function.

The core function of a CPG can be expressed as a set of *dynamical invariants* that must be preserved. Recent research (Reyes et al., 2008) has clearly shown that the pyloric CPG of the lobster presents substantial differences from animal to animal, yet robustly preserves the relationship between bursting period and phase lag between neurons. Even when one single CPG is forced by replacing one synapse with an artificially modified synapse, the CPG is shown to have some mechanism by which this relationship is preserved.

Traditional techniques to build CPG control in robots try to solve motor control tasks by mimicking known biological systems, in particular circuit topology and the presence of oscillators. This approach provides limited capability to deal with unforeseen scenarios. The concept of the dynamical invariant allows for a new design strategy in which the motor control can be achieved in a way that is not specified a priori, but emerges from the set of biologically inspired principles. Having a general set of invariant preserving mechanisms allows for more freedom in the process of design, providing more flexible and general solutions to existing problems in robot control.

2 INGREDIENTS FOR DYNAMICAL INVARIANT PRESERVING CPGS

In this section we discuss scientific results on the mechanisms known to contribute to the essential flexibility and robustness of CPGs, which may be directly related to the implementation of dynamical invariants.

2.1 Topologies

CPGs are known for their flexibility: they generate a robust rhythmic activity with a recognizable pattern, but they can be modulated by external input. The final shape of the rhythm produced by one CPG will depend on many factors: intrinsic properties of individual neurons and synapses, strength and sign of the couplings and network topology, all contribute to the function of the circuit. It is known (Huerta et al., 2001; Stiesberg et al., 2007) that non-open topologies maximize the quality of the rhythm produced in terms of flexibility and regularity. A non-open topology is that in which every neuron receives at least one connection from another CPG member, in contrast to an "open" topology, where at least one neuron does not receive synapses from any other CPG member

In the design of an artificial CPG for robot control, it is necessary to take this into account. All neurons within the circuit must receive feedback from the rest of the circuit, so that knowledge about how the CPG is performing is distributed to all units. Of course, not all neurons will process this information in the same way.

Different works, both theoretical and experimental, have studied the role of synapses in the synchronization of neural oscillators. However, this study cannot be decoupled from the properties of the units being connected. Depending on the intrinsic characteristics of neurons, synaptic activity will have different effects. In fact, different CPGs may adopt different combinations of neural and synaptic properties to achieve the same goal (Prinz et al., 2004). The result is that the CPG designer has a wider variety of mechanisms to choose from. For instance, it is widely accepted that in-phase synchronization can be achieved through excitatory interaction, but it can also be achieved by inhibitory interaction with the appropriate conditions (Wang and Rinzel, 1992). And conversely, anti-phase synchronization is usually considered to happen under inhibition (Wang and Rinzel, 1992; Rowat and Selverston, 1997) but it can be found to happen under excitatory connections as well (Kopell and Somers, 1995).

Usually phase relationship between modules of a CPG is an important dynamical invariant that must be preserved. There exist some important theoretical studies on the mechanisms for phase locking between oscillators. A general study based on a phase model can be found in (Kopell and Ermentrout, 2000). However, it poses strong restrictions upon the coupling between oscillators. More realistic approaches that take into account synaptic dynamics for predicting the type of synchronization among bursting neurons can be found in (Oprisan et al., 2004) and (Elson et al., 2002).

When building complex artificial CPGs, it is difficult to predict how the coupling of neurons within a given topology will affect synchronization. Being able to predict stable phase-locking regimes as in the above mentioned works could help us design complex CPGs, with rich dynamics and guarantee that the resulting phase and frequency relationships will be the desired ones.

2.2 Starting, Stopping and Maintenance of Rhythm

Few studies in robotics are concerned with how CPGs are started and stopped. There exist various mechanisms underlying the starting, stopping and sustaining of rhythm in living CPGs.

In cases like breathing or heart beating, the group of neurons responsible for the maintenance of rhythm cannot afford to stop. There are other cases, however, in which a set of neurons needs to be activated and, after some time of activity, deactivated.

In the first case, special neurons called "pacemaker" neurons are responsible for setting the pace at which the rest of the group will oscillate (Selverston et al., 2009). Thanks to biophysical sub cellular mechanisms, they can produce oscillations in isolation, without external input. Through synaptic connections, the activity of pacemakers is "distributed" to other members of the groups.

Pacemaker neurons can usually adapt their burst frequency over a large range. By means of external input and feedback from other neurons in the network, the pacemaker group will be able to decide if the circuit is working properly and, if not, set the appropriate frequency. Some CPGs show redundancy mechanisms in order to maintain the correct rhythm in case of failure. Several neuron models are capable of endogenous oscillation as seen in living CPG neurons: many variations of the bio physically detailed model by Hodgkin and Huxley (Hodgkin and Huxley, 1952); one classical model for bursting activity (Hindmarsh and Rose, 1984) and other recent models with less computational requirements (Rulkov, 2002; Aguirre et al., 2005).

In the second case, initiation and termination are usually caused by external forces. A silent group of neurons may be "recruited" by an external excitatory force. Such is the case in (Arshavsky et al., 1998), where neuron group IN 12 is recruited by neuron group IN 8 and inhibited by neuron group IN 7. This a very good example in which neurons that do not generate a rhythm autonomously contribute to the proper functioning of the CPG.

Interestingly, this CPG presents another important feature. Depending on the intensity of the swimming, an *early* group of neurons is recruited for weak swimming and if more powerful strokes are needed, a neuron group with higher activation threshold will also be recruited. This is the *delayed* group of neurons.

Another mechanism found in some neurons is known as "post-inhibitory rebound". This feature is widely observed, in particular again in (Arshavsky et al., 1998). A neuron with this feature will remain silent while inhibited. If inhibition is suddenly released, the neuron will respond with a spike or burst of activity.

It is believed that the selection of motor programs may be subject to an inhibitory mechanism (Grillner et al., 2005). Each possible motor program is kept under inhibition, until upper control centers decide to activate it. At this point, inhibition is released. By the mechanism of post-inhibitory rebound, a motor program could immediately start upon lifting of inhibition. Note that this is not equivalent to activation by excitation. The difference is that in a normal state, the circuit responsible of the motor program will not spontaneously activate because of noise, since inhibition will keep it forcibly silent.

The mechanism for termination of activity will depend on the goal of the CPG. If neurons in the CPG are not capable of endogenous oscillation, they may just go silent by themselves after activity has been elicited by one of the mechanisms mentioned earlier. If the activity needs to be sustained, then mutual excitatory connections within a group of neurons will keep them firing. This is known as a "pool" of neurons.

So the repertoire for how activity is elicited, maintained and stopped is really ample. However, each mechanism has different subtleties, depending on the underlying cellular characteristics, on whether inhibition or excitation should be preferred, etc.

2.3 Motor Command Coding

Individual CPG neurons display a mainly bursting activity. Motor commands are encoded in some aspect of the neuron's activity, for instance in the frequency of the spikes, or on the precise timing between them. How information is extracted from this activity is not trivial (Brezina et al., 2000), but simple mechanisms can be used in robots.

The fact that rhythmic motor commands are encoded using bursting neurons has an advantage over the classical view of CPGs in robotics: the mechanisms that encode motor commands and those used for synchronization are decoupled. In the traditional view, one non-linear oscillator represents a whole CPG. Then, one variable of the oscillator is used as output to the controlled joint and to other oscillators that need to be synchronized. Bursting neurons have the ability to flexibly adapt the timing between bursts and still produce robustly reproducible bursting patterns. With this decoupling, the system gains simultaneously in robustness and flexibility. In addition, some bursting neurons have the ability to encode information relative to their identity and their context and send messages to neighbouring neurons, as we discuss in the following section.

2.4 Neural Signatures

Recent experiments have shown that CPG individual cells have neural signatures that consist of neuron specific spike timings in their bursting activity (Szucs et al., 2003). Model simulations indicate that neural signatures that identify each cell can play a functional role in the activity of CPG circuits (Latorre et al., 2006). These signatures coexist with the information encoded in the slow wave rhythm of the CPG which results in a neural signal with multiple simultaneous codes. Readers of this signal (muscles and other neurons) can take advantage of the multiple simultaneous codes and process them one by one, or simultaneously in order to perform different tasks. The sender and the content of the signals can also be used separately to discriminate the information received by a neuron by distinctly processing the input as a function of these codes. These mechanisms can contribute to build dynamical invariants through a self-organizing strategy that includes non supervised learning as a function of local discrimination. Artificial CPG networks built with neurons that display neural signatures allow for a new set of learning rules that include not only the modification of the connections, but also the parameters that affect the local discrimination.

2.5 Homeostasis: Self-regulation Mechanisms

CPG research has also shown the presence of many homeostatic mechanisms to self-regulate and to deal with unexpected circumstances at the cellular and neural network levels and in multiple time scales (Marder and Goaillard, 2006). Dynamical invariants that work in short time scales can be built with the above mentioned mechanisms that involve neuron and synapse dynamics and specific topologies. However other types of invariants working in longer time scales can use mechanisms of adaptation and learning (including sub cellular plasticity) to generate rhythms even in the absence of the input that sustains this rhythm under normal circumstances (Thoby-Brisson and Simmers, 1998). Long scale dynamical invariants arise from self-regulatory mechanism that involve tightly regulated synaptic and intrinsic properties. Models could implement this self regulation through specific synaptic and sub cellular learning paradigms.

Implementing already known sub cellular and network learning mechanisms (Marder and Prinz, 2002), CPGs may easily accommodate the degradation process of a working robot. If a joint loses torque, if pieces begin to wear off, a CPG implementing these mechanisms could adjust its function to reflect the natural evolution of the mechanical parts. Even if some part suddenly stops working or is disconnected from the main body of the robot, the CPG could still keep its original function.

3 CONCLUSIONS

Central Pattern Generator neural circuits are well known for their ability to generate and sustain rhythmic activity. And they do so in a robust manner, tolerant to perturbations, but with flexibility, being able to adjust their working regime to the requirements of the environment.

We have presented recent results that extend the idea of a CPG. Results from which the robotics field will surely benefit. To begin with, we have introduced a new design idea, termed *dynamical invariant*. The key aspect of designing an artificial CPG should be specifying what restrictions must be kept under any condition. So, the presence of one or several dynamical invariants will guarantee the effectiveness of motor control in unknown or changing environments. Living CPGs use different strategies to implement their invariants, we have reviewed some of them in this paper.

First, we have reported on studies that conclude that those topologies that maximize rhythm quality are non-open topologies. That is, every neuron in the circuit receives at least one input from the rest of the circuit. Following this, we review those mechanisms that are responsible for rhythm start, stop and sustenance. Few works known to the authors have concerned themselves with the issue of initiation and termination of CPGs in robotics. We believe that the works mentioned here will be of great inspiration to robot designers. Then, we discuss the mechanism by which living motor neurons are believed to code motor commands. Studying it, robotic CPGs can gain greater flexibility and robustness by decoupling synchronization and coding mechanisms. Next, a recently discovered property of bursting neurons is presented: neural signatures. This mechanism increases the computing capabilities of a CPG. It allows neurons to code context specific information so that receivers of different neural *messages* may discriminate their input according to the identity of the sender, circumstances on which a message occurs, etc. We believe that this result will widen the view of CPGs from simple pattern generators to information processing capable circuits. And finally, we present reports on the ability of long term adaptation of CPGs. Several results indicate that CPGs can satisfy their dynamical invariants despite differences from animal to animal, or development of the animal from young to adult or changes in function and/or size due to illnesses and injuries. Understanding the mechanisms underlying this ability will allow designers to construct CPGs that will adapt over a range of "robot families" and to the consequences of long time operation like loss of performance and degradation of the parts.

To conclude, we believe that biology has still many secrets to reveal. We have pointed out key issues that robotics has not yet explored, but with promising potential. We are beginning to implement some of these ideas in our own robots (Herrero-Carrón et al., 2010), and look forward to the next generation of CPGs.

ACKNOWLEDGEMENTS

Work supported by MICINN BFU2009-08473, CAM S-SEM-0255-2006 and TIN 2007-65989. Fernando Herrero-Carrón with an FPU-UAM grant.

REFERENCES

Aguirre, C., Campos, D., Pascual, P., and Serrano, E. (2005). A model of spiking-bursting neuronal behavior using a piecewise linear two-dimensional map. *Computational Intelligence and Bioinspired Systems*, pages 130–135.

Arshavsky, Y. I., Deliagina, T. G., Orlovsky, G. N., Panchin, Y. V., Popova, L. B., and Sadreyev, R. I. (1998). Analysis of the central pattern generator for swimming in the mollusk clione. *Annals of the New York Academy of Sciences*, 860(1):51–69.

Brezina, V., Orekhova, I. V., and Weiss, K. R. (2000). The neuromuscular transform: The dynamic, nonlinear link between motor neuron firing patterns and muscle contraction in rhythmic behaviors. *J Neurophysiol*, 83(1):207–231.

Bucher, D., Prinz, A. A., and Marder, E. (2005). Animal-to-animal variability in motor pattern production in adults and during growth. *J. Neurosci.*, 25(7):1611–1619.

Elson, R. C., Selverston, A. I., Abarbanel, H. D., and Rabinovich, M. I. (2002). Inhibitory synchronization of bursting in biological neurons: Dependence on synaptic time constant. *J Neurophysiol*, 88(3):1166–1176.

Grillner, S. (2006). Biological pattern generation: The cellular and computational logic of networks in motion. *Neuron*, 52(5).

Grillner, S., Hellgren, J., Ménard, A., Saitoh, K., and Wikström, M. A. (2005). Mechanisms for selection of basic motor programs–roles for the striatum and pallidum. *Trends Neurosci*, 28(7):364–370.

Herrero-Carrón, F., Rodríguez, F. d. B., and Varona, P. (2010). Bio-inspired design strategies for cpg control in modular robotics. *Submitted.*

Hindmarsh, J. L. and Rose, R. M. (1984). A model of neuronal bursting using three coupled first order differential equations. *Proceedings Of The Royal Society Of London. Series B, Containing Papers Of a Biological Character. Royal Society (Great Britain)*, 221(1222):87–102.

Hodgkin, A. L. and Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, 117(4):500–544.

Huerta, R., Varona, P., Rabinovich, M. I., and Abarbanel, H. D. (2001). Topology selection by chaotic neurons of a pyloric central pattern generator. *Biological Cybernetics*, 84(1):L1–L8.

Ijspeert, A. (2008). Central pattern generators for locomotion control in animals and robots: A review. *Neural Networks*, 21(4):642–653.

Kopell, N. and Ermentrout, G. (2000). Mechanisms of phase-locking and frequency control in pairs of coupled neural oscillators.

Kopell, N. and Somers, D. (1995). Anti-phase solutions in relaxation oscillators coupled through excitatory interactions. *Journal of Mathematical Biology*, 33(3):261–280.

Latorre, R., Rodríguez, F., and Varona, P. (2006). Neural signatures: Multiple coding in spiking–bursting cells. *Biological Cybernetics*, 95(2):169–183.

Marder, E. and Goaillard, J.-M. (2006). Variability, compensation and homeostasis in neuron and network function. *Nature Reviews Neuroscience*, 7(7):563–574.

Marder, E. and Prinz, A. A. (2002). Modeling stability in neuron and network function: the role of activity in homeostasis. *BioEssays : news and reviews in molecular, cellular and developmental biology.*, 24(12):1145–1154.

Oprisan, S. A., Prinz, A. A., and Canavier, C. C. (2004). Phase resetting and phase locking in hybrid circuits of one model and one biological neuron. *Biophys J*, 87(4):2283–2298.

Prinz, A. A., Bucher, D., and Marder, E. (2004). Similar network activity from disparate circuit parameters. *Nature Neuroscience*, 7(12):1345–1352.

Reyes, M., Huerta, R., Rabinovich, M., and Selverston, A. (2008). Artificial synaptic modification reveals a dynamical invariant in the pyloric cpg. *European Journal of Applied Physiology*, 102(6):667–675.

Rowat, P. F. and Selverston, A. I. (1997). Oscillatory mechanisms in pairs of neurons connected with fast inhibitory synapses. *Journal of Computational Neuroscience*, 4(2):103–127.

Rulkov, N. F. (2002). Modeling of spiking-bursting neural behavior using two-dimensional map. *Physical Review E*, 65(4):041922+.

Selverston, A. B., Szücs, A., Huerta, R., Pinto, R. D., and Reyes, M. (2009). Neural mechanisms underlying the generation of the lobster gastric mill motor pattern. *Frontiers in Neural Circuits.*

Stiesberg, G. R., Reyes, M. B., Varona, P., Pinto, R. D., and Huerta, R. (2007). Connection topology selection in central pattern generators by maximizing the gain of information. *Neural Computation*, 19(4):974–993.

Szucs, A., Pinto, R. D., Rabinovich, M. I., Abarbanel, H. D., and Selverston, A. I. (2003). Synaptic modulation of the interspike interval signatures of bursting pyloric neurons. *Journal of neurophysiology*, 89(3):1363–1377.

Thoby-Brisson, M. and Simmers, J. (1998). Neuromodulatory inputs maintain expression of a lobster motor pattern-generating network in a modulation-dependent state: Evidence from long-term decentralization in vitro. *J. Neurosci.*, 18(6):2212–2225.

Wang, X.-J. and Rinzel, J. (1992). Alternating and synchronous rhythms in reciprocally inhibitory model neurons. *Neural Comput.*, 4(1):84–97.

LASER BASED TELEROBOTIC CONTROL FOR ASSISTING PERSONS WITH DISABILITIES PERFORM ACTIVITIES OF DAILY LIVING

Karan Khokar, Redwan Alqasemi and Rajiv Dubey
Department of Mechanical Engineering, University of South Florida, 4202 E. Fowler Ave., Tampa, Florida, U.S.A.
kkhokar@mail.usf.edu, alqasemi@usf.edu, dubey@eng.usf.edu

Keywords: Telerobotics, Rehabilitation Robotics, Human-Machine Interaction.

Abstract: In this paper we demonstrate an innovative use of range information from a laser sensor mounted on the end- effector of a remote robot manipulator to assist persons with disabilities carry out ADL (Activities of Daily Living) tasks in unstructured environments. Laser range data is used to determine goals, identify targets and via points to enable autonomous execution of trajectories. The human operator performs minimal teleoperation and is primarily involved in higher level decision making. We hypothesize that laser based assistance improves task performance in terms of time and accuracy and also reduces the cognitive load on the user executing the task. Tests on ten healthy human subjects in executing a remote manipulation task conform the hypothesis.

1 INTRODUCTION

Robotic devices have thus been used to enable physically disabled individuals to execute ADL tasks (Bolmsjo, Neveryd and Eftring, 1995). However, teleoperation of a remote manipulator puts a lot of cognitive load on the operator (Bolmsjo et al., 1995). Our previous work at the Rehabilitation Robotics Laboratory at University of South Florida has focused on augmenting the performance of motion-impaired users in job-related tasks using scaled teleoperation and haptics (Pernalete, Yu, Dubey and Moreno, 2002) and assistance based on real-time environmental information and user intention (Yu, Alqasemi ,Dubey and Pernalete, 2005). In this work the laser data is used to determine goal points, identify targets and via points in unstructured environments that enables autonomous execution of certain sub tasks under human supervisory control thus providing assistance to the human. The human is still in the loop to make high level decisions like pointing the laser to critical points in the remote environment by teleoperating the arm.

Hasegawa, Suehiro and Takase (1991) made use of the laser range information to compute and record 3D co-ordinates of points on objects in the environment, to enable autonomous task execution. Takahashi and Yashige (2000) presented a simple and easy to use laser based robot positioning system to assist the elderly in doing daily pick-and-place activities. Nguyen, Anderson, Trevor, Jain, Xu and Kemp (2008) made use of an arrangement consisting of a laser pointer, a monochrome camera, a color filter and a stereo camera pair to estimate the 3D co-ordinates of a point in the environment and thereby fetch objects in the environment designated with the laser pointer.

2 LASER ASSISTED CONTROL

We have used PUMA560 as the remote manipulator and Phantom Omni haptic device as the master. The human user by means of teleoperation points the laser to critical points in the environment. These critical points could be goal points, objects or planar surfaces of interest. The laser sensor is mounted on the end-effector of PUMA and has the same orientation as the end-effector. The laser beam direction thus will always be parallel to the z axis of the end-effector.

2.1 Laser Assisted Target Position Determination and Autonomous Trajectory Execution

The user points the laser to a specific target and locks the target by ceasing to teleoperate. Then as the user commands by pressing a keyboard key, the machine generates a linear trajectory from the current PUMA end effector location to the target point determined by the laser and also executes the trajectory.

For generating a linear trajectory, the initial and final transformation matrices are required. The initial point transformation matrix is determined from PUMA forward kinematics and that for the final point is determined using laser range data and (1) (ref Fig. 1).

$$T_O^B = T_E^B * T_L^E * T_O^L \tag{1}$$

2.2 Laser Assisted Autonomous Surface Alignment

For certain ADL tasks determination of orientation of the planar surface associated with the target is necessary to orient the end-effector in such a way that it is aligned with the perpendicular to the surface. This alignment has been implemented as an autonomous function using laser data and it helps the user to manipulate a target from a convenient angle in teleoperation.

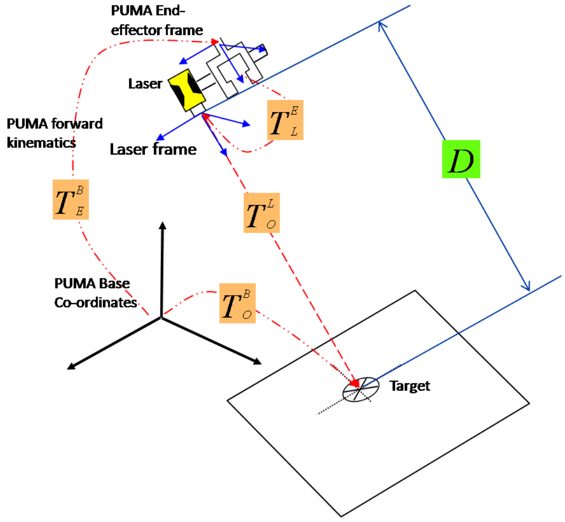


Figure 1: Autonomous trajectory generation concept.

The user points the laser to three distinct points on the planar surface to record their co-ordinates. When the user activates autonomous alignment algorithm, the transformation matrix required for end-effector alignment is computed online and the end-effector aligns with the surface. The z-component of the rotation component of the transformation matrix is the same as the unit vector along normal to the surface which is computed by the cross product of the vectors connecting the three points P1, P2 and P3. The x and y-components are then obtained using the right-hand rule.

3 APPLICATION OF LASER BASED CONCEPT IN TASK EXECUTION

Fig. 2 shows the sequence of steps in executing a pick and place task using laser assistance. The user starts with locating the destination point by pointing the laser in teleoperation and commanding the system to record the co-ordinates of that point (Fig. 2(a)). Next the user identifies three random points on the planar surface with the laser (Fig. 2(b)). After this the user locates the target object with the laser and commands the execution of the autonomous trajectory (Fig. 2(c)). When the arm stops near the target (Fig. 2(d)) the user commands autonomous alignment with the planar surface (Fig. 2(e)). Now as the arm is in a convenient configuration for grasping the user makes fine adjustment in teleoperation to locate the gripper exactly over the target to grasp it.

Next the user commands the system to autonomously execute a path from the current location to the destination point while avoiding the obstacle. The path is executed in three linear segments. , the first being from the current PUMA location to a point vertically above it so that the arm is clear of the tallest obstacle (Fig. 2(e) and 2(f)), the second being in the horizontal plane from this point to a point vertically above the destination (Fig. 2(f) and 2(g)) and the third is to a point just above the destination (Fig. 2(g) and 2(h)). The initial and final transformation matrices for each segment are computed by the system from forward kinematics, the heeight of the tallest obstacle and the destination point coordinates. The height of the tallest obstacle is determined by pointing the laser to the top of the obstacle and recording the co-ordinates.

After the arm has traversed the path, the user makes fine movements to precisely locate the target over the destination by teleoperation and places the target. Thus we see that the human is only a supervisor and is involved in minimum teleoperation. The trajectories and generated online and executed autonomously. Moreover the interface consisting of pointing with laser and pressing

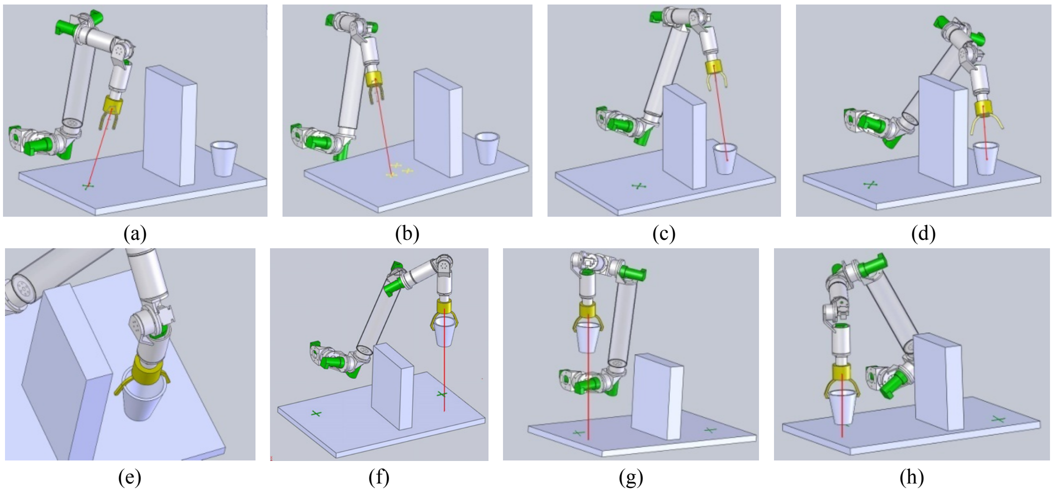


Figure 2: Laser based features in pick and place task execution: (a) Pointing to destination point for path planning (b) Pointing to platform points for planar surface orientation determination (c) Pointing to target (cup) for autonomous trajectory generation (d) End of autonomous trajectory (e) Autonomous end-effector alignment (f) Autonomous path – end of first segment (g) Autonomous path – end of second segment (h) Autonomous path – end of third segment.

keyboard keys is simple to use. This way we aim at relieving the human from much cognitive load in task execution. The destination point, target and surface could be located anywhere in the environment and thus the method can be used for manipulating in unstructured environments.

4 TESTS AND RESULTS

The test bed consists of PUMA and Omni manipulators (Fig. 3). A SICK DT60 laser range finder and a Logitech MT Orbit CCD Camera were mounted on the PUMA end effector.

For testing, 10 healthy subjects were asked to perform a pick and place task thrice in each of the two modes, one is the unassisted teleoperation mode and the other is the laser assisted mode. For each run the time taken to complete the task, the end effector Cartesian co-ordinates and the user experience were recorded. Before starting the tests the subjects were given sufficient time to acclimatize with the system. In general each subject was given 5 to 6 trials before testing.

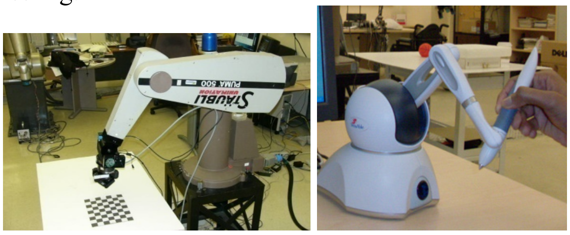


Figure 3: Test bed of PUMA and Omni manipulators.

The experimental set up is shown in Fig. 4. The cup is the target which is to be picked up from the location shown, and placed on the orange sticky shown to the left. The folder simulates an obstacle. The task here is to command the arm from a ‘ready’ position towards the target, grasp the target and place it over the destination point while avoiding the obstacle.

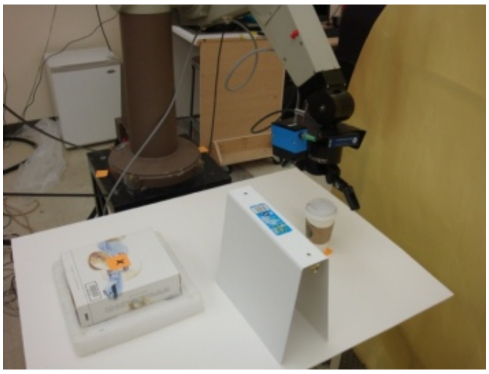


Figure 4: Experimental set-up for pick and place task.

The task performance is compared between the two modes in terms of accuracy and time. Accuracy is presented in the form of the path traversed by the end effector in the two modes.. Plot for only one subject is shown in Fig. 5. We observe that the motion generated by laser assisted mode was, continuous and accurate. On the other hand the motion in the unassisted mode was discontinuous and the arm often deviated from the path. The subjects had tremendous difficulties in teleoperating the PUMA without any assistance. The loops around the pick-up point are due to subjects repeatedly trying to orient the gripper properly so that it is in a convenient configuration for grasping. Orienting the arm properly was one of the most challenging activities the subjects faced.

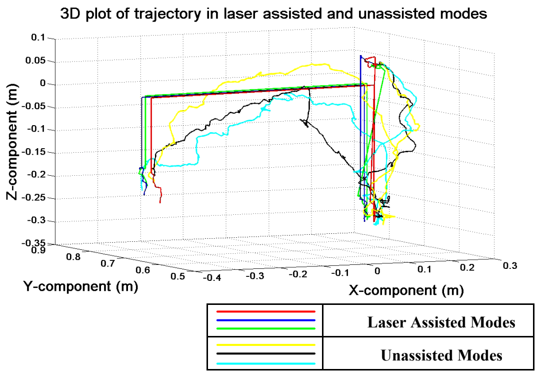


Figure 5: Accuracy in path traversal for laser assisted and unassisted modes.

The average time per subject per mode was computed and is presented in the form of plots in Fig. 6. We observe that each subject spent more time to complete the task in the unassisted mode than in the laser assisted mode.

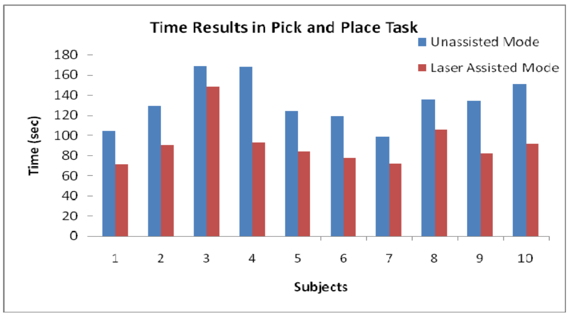


Figure 6: Time plots in executing pick and place task.

The user experience in executing the task in the two modes was also noted. The users preferred the laser assisted mode as generating task plans by picking up points with the laser was easier for them and the robot executed the task. In teleoperating without assistance they experienced a lot of mental load.

5 CONCLUSIONS

Our hypothesis that the laser assisted telerobotic control methodology of task execution would improve the task performance and reduce the mental load on the users has been validated by tests on healthy subjects on a pick and place task. The task performance in terms of time and accuracy improved and the subjects were overwhelmingly in favour of using the laser assisted mode in executing the task as they experienced less cognitive load. With the results obtained we believe that this telerobotic system would make it possible for persons with disabilities to execute ADL tasks with much greater ease. Next we intend to test the method on persons with disabilities and for a variety of ADLs.

REFERENCES

Bolmsjo, G., Neveryd, H., Eftring, H. (1995). Robotics in Rehabilitation. IEEE Transactions on Rehabilitation Engineering. 3(1), 77 -83.Hasegawa, T., Suehiro, T. and Takase, K. (1991). A Robot System for Unstructured Environments Based on an Environment Model and Manipulation Skills. *IEEE International Conference on Robotics and Automation* (Vol. 1, pp. 916-923).

Nguyen, H., Anderson, C., Trevor, A., Jain, A., Xu, Z. and Kemp, C. (2008). El-E: An Assistive Robot that Fetches Objects from Flat Surfaces. *The Robotic Helpers Workshop at HRI.*

Pernalete, N., Yu, W., Dubey, R. and Moreno, W. (2002). Development of a Robotic Haptic Interface to Assist the Performance of Vocational Tasks by People with Disability. *IEEE International Conference on Robotics and Automation*. (Vol. 2, pp. 1269-1274).

Takahashi, Y. and Yashige, M. (2000). Robotic manipulator operated by human interface with positioning control using laser pointer. *IEEE 26th Annual Conference of the Industrial Electronics Society* (Vol. 1, pp. 608-613).

Tarn, T., Xi, N. and Guo, C. (1996). Task-Oriented Human and Machine Co-Operation in Telerobotic Systems. *Annual Reviews in Control*. (Vol. 20, pp. 173-178).

Yu, W., Alqasemi , R., Dubey, R. and Pernalete, N. (2005). Telemanipulation Assistance based on Motion Intention Recognition. *IEEE International Conference on Robotics and Automation*. (pp. 1121-1126).

AUTHOR INDEX

AUTHOR INDEX (CONT.)

AUTHOR INDEX (CONT.)

Proceedings Indexed by:

THOMSON REUTERS
CONFERENCE PROCEEDINGS
CITATIONS INDEX

Inspec

dblp.uni-trier.de
Computer Science
Bibliography

(Awaiting confirmation
of indexation)

Proceedings of ICINCO 2010
International Conference on Informatics in Control, Automation and Robotics
ISBN: 978-989-8425-01-0
http://www.icinco.org