ICINCO 2007

FOURTH INTERNATIONAL CONFERENCE ON INFORMATICS IN CONTROL, AUTOMATION AND ROBOTICS

Proceedings

Robotics and Automation - Vol. 1

ANGERS, FRANCE · MAY 9-12, 2007

ORGANIZED BY

INSTICC

IN COOPERATION WITH

aaai

CO-SPONSORED BY

IFAC

GdR MACS

CNRS

CEA

INSTICC PRESS

ICINCO 2007

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

Volume RA-1

Angers, France

May 9 – 12, 2007

Co-organized by
INSTICC – Institute for Systems and Technologies of Information, Control and Communication
and
University of Angers

Co-sponsored by
IFAC - International Federation of Automatic Control
GDR MACS - Groupe de Recherche "Modélisation, Analyse et Conduite des Systèmes dynamiques
CNRS - Centre National de la Recherche Scientifique
and
EEA – Club des Enseignants en Electronique, Electrotechnique et Automatique

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

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

Printed in Portugal

ISBN: 978-972-8865-83-2

Depósito Legal: 257879/07

http://www.icinco.org

secretariat@icinco.org

BRIEF CONTENTS

INVITED SPEAKERS

Dimitar Filev

The Ford Motor Company

U.S.A.

Mark W. Spong

University of Illinois at Urbana-Champaign

U.S.A.

Patrick Millot

Université de Valenciennes

France

SPECIAL SESSION CHAIRS

Samir Ladaci

IRCCyN

France

Jean-Louis Boimond

LISA

France

Jean Jacques Loiseau

IRCCyN Nantes

France

Oleg Gusikhin

Ford Research & Adv. Engineering

U.S.A.

ORGANIZING AND STEERING COMMITTEES

Conference Co-chairs

Jean-Louis Ferrier, University of Angers, France

Joaquim Filipe, Polytechnic Institute of Setúbal / INSTICC, Portugal

Program Co-chairs

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

Janan Zaytoon, CReSTIC, URCA, France

Proceedings Production

Andreia Costa, INSTICC, Portugal

Bruno Encarnação, INSTICC, Portugal

Vitor Pedrosa, INSTICC, Portugal

CD-ROM Production

Paulo Brito, INSTICC, Portugal

Webdesigner and Graphics Production

Marina Carvalho, INSTICC, Portugal

Secretariat and Webmaster

Marina Carvalho, INSTICC, Portugal

PROGRAM COMMITTEE

Eugenio Aguirre, University of Granada, Spain

Arturo Hernandez Aguirre, Centre for Research in Mathematics, Mexico

Frank Allgower, University of Stuttgart, Germany

Fouad AL-Sunni, KFUPM, Saudi Arabia

Bala Amavasai, Sheffield Hallam University, U.K.

Francesco Amigoni, Politecnico di Milano, Italy

Yacine Amirat, University Paris 12, France

Nicolas Andreff, LASMEA, France

Stefan Andrei, National University of Singapore, Singapore

Plamen Angelov, Lancaster University, U.K.

Luis Antunes, GUESS/Universidade de Lisboa, Portugal

Peter Arato, Budapest University of Technology and Economics, Hungary

Helder Araújo, University of Coimbra, Portugal

Gustavo Arroyo-Figueroa, Instituto de Investigaciones Electricas, Mexico

Marco Antonio Arteaga, Universidad Nacional Autonoma de Mexico, Mexico

Vijanth Sagayan Asirvadam, University Technology Petronas, Malaysia

Nikos Aspragathos, University of Patras, Greece

Robert Babuska, TU Delft, The Netherlands

Ruth Bars, Budapest University of Technology and Economics, Hungary

Karsten Berns, University Kaiserslautern, Germany

Robert Bicker, University of Newcastle upon Tyne, U.K.

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

Patrick Boucher, SUPELEC, France

Alan Bowling, University of Notre Dame, U.S.A.

Edmund Burke, University of Nottingham, U.K.

Kevin Burn, University of Sunderland, U.K.

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

Luis M. Camarinha-Matos, New University of Lisbon, Portugal

Marco Campi, University of Brescia, Italy

Marc Carreras, University of Girona, Spain

Jorge Martins de Carvalho, FEUP, Portugal

Alicia Casals, Technical University of Catalonia, Spain

Alessandro Casavola, University of Calabria, Italy

Christos Cassandras, Boston University, U.S.A.

Riccardo Cassinis, University of Brescia, Italy

Raja Chatila, LAAS-CNRS, France

Tongwen Chen, University of Alberta, Canada

YangQuan Chen, Utah State University, U.S.A.

Albert M. K. Cheng, University of Houston, U.S.A.

Graziano Chesi, University of Hong Kong, China

Sung-Bae Cho, Yonsei University, Korea

Ryszard S. Choras, University of Technology & Agriculture, Poland

Carlos Coello Coello, CINVESTAV-IPN, Mexico

Patrizio Colaneri, Politecnico di Milano, Italy

António Dourado Correia, University of Coimbra, Portugal

Yechiel Crispin, Embry-Riddle University, U.S.A.

Keshav Dahal, University of Bradford, U.K.

Mariolino De Cecco, DIMS - University of Trento, Italy

Bart De Schutter, Delft University of Technology, The Netherlands

Angel P. del Pobil, Universitat Jaume I, Spain

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

Rüdiger Dillmann, University of Karlsruhe, Germany

Feng Ding, Southern Yangtze University, China

Denis Dochain, Université Catholique de Louvain, Belgium

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

Alexandre Dolgui, Ecole des Mines de Saint Etienne, France

PROGRAM COMMITTEE (CONT.)

Marco Dorigo, Université Libre de Bruxelles, Belgium

Petr Ekel, Pontifical Catholic University of Minas Gerais, Brazil

Heinz-Hermann Erbe, TU Berlin, Germany

Gerardo Espinosa-Perez, Universidad Nacional Autonoma de Mexico, Mexico

Simon Fabri, University of Malta, Malta

Sergej Fatikow, University of Oldenburg, Germany

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

Jean-Louis Ferrier, Université d'Angers, France

Florin Gheorghe Filip, The Romanian Academy & The National Institute for R&D in Informatics (ICI), Romania

Georg Frey, University of Kaiserslautern, Germany

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

Colin Fyfe, University of paisley, U.K.

Dragan Gamberger, Rudjer Boskovic Institute, Croatia

Leonardo Garrido, Tecnológico de Monterrey, Mexico

Ryszard Gessing, Silesian University of Technology, Poland

Lazea Gheorghe, Technical University of Cluj-Napoca, Romania

Maria Gini, University of Minnesota, U.S.A.

Alessandro Giua, University of Cagliari, Italy

Luis Gomes, Universidade Nova de Lisboa, Portugal

John Gray, University of Salford, U.K.

Dongbing Gu, University of Essex, U.K.

Jason Gu, Dalhousie University, Canada

José J. Guerrero, Universidad de Zaragoza, Spain

Jatinder (Jeet) Gupta, University of Alabama in Huntsville, U.S.A.

Thomas Gustafsson, Luleå University of Technology, Sweden

Maki K. Habib, Saga University, Japan

Hani Hagras, University of Essex, U.K.

Wolfgang Halang, Fernuniversitaet, Germany

J. Hallam, University of Southern Denmark, Denmark

Riad Hammoud, Delphi Electronics & Safety, U.S.A.

Uwe D. Hanebeck, Institute of Computer Science and Engineering, Germany

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

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

Vincent Hayward, McGill Univ., Canada

Dominik Henrich, University of Bayreuth, Germany

Francisco Herrera, University of Granada, Spain

Victor Hinostroza, University of Ciudad Juarez, Mexico

Weng Ho, National University of Singapore, Singapore

Wladyslaw Homenda, Warsaw University of Technology, Poland

Alamgir Hossain, Bradford University, U.K.

Dean Hougen, University of Oklahoma, U.S.A.

Amir Hussain, University of Stirling, U.K.

Seth Hutchinson, University of Illinois, U.S.A.

Atsushi Imiya, IMIT Chiba Uni, Japan

Sirkka-Liisa Jämsä-Jounela, Helsinki University of Technology, Finland

Ray Jarvis, Monash University, Australia

Odest Jenkins, Brown University, U.S.A.

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

Ivan Kalaykov, Örebro University, Sweden

Dimitrios Karras, Chalkis Institute of Technology, Greece

Dusko Katic, Mihailo Pupin Institute, Serbia

Graham Kendall, The University of Nottingham, U.K.

Uwe Kiencke, University of Karlsruhe (TH), Germany

Jozef Korbicz, University of Zielona Gora, Poland

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

PROGRAM COMMITTEE (CONT.)

Bart Kosko, University of Southern California, U.S.A.

George L. Kovács, Hungarian Academy of Sciences, Hungary

Krzysztof Kozlowski, Poznan University of Technology, Poland

Gerhard Kraetzschmar, Fraunhofer Institute for Autonomous Intelligent Systems, Germany

Cecilia Laschi, Scuola Superiore Sant'Anna, Italy

Loo Hay Lee, National University of Singapore, Singapore

Soo-Young Lee, KAIST, Korea

Graham Leedham, University of New South Wales (Asia), Singapore

Cees van Leeuwen, RIKEN BSI, Japan

Kauko Leiviskä, University of Oulu, Finland

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

Yangmin Li, University of Macau, China

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

Cheng-Yuan Liou, National Taiwan University, Taiwan

Vincenzo Lippiello, Università Federico II di Napoli, Italy

Honghai Liu, University of Portsmouth, U.K.

Luís Seabra Lopes, Universidade de Aveiro, Portugal

Brian Lovell, The University of Queensland, Australia

Peter Luh, University of Connecticut, U.S.A.

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

N. P. Mahalik, Gwangju Institute of Science and Technology, Korea

Bruno Maione, Politecnico di Bari, Italy

Frederic Maire, Queensland University of Technology, Australia

Om Malik, University of Calgary, Canada

Danilo Mandic, Imperial College, U.K.

Jacek Mandziuk, Warsaw University of Technology, Poland

Hervé Marchand, INRIA, France

Philippe Martinet, LASMEA, France

Aleix Martinez, Ohio State University, U.S.A.

Aníbal Matos, Faculdade de Engenharia da Universidade do Porto (FEUP), Portugal

Rene V. Mayorga, University of Regina, Canada

Barry McCollum, Queen's University Belfast, U.K.

Ken McGarry, University of Sunderland, U.K.

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

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

Basil Mertzios, (1)Thessaloniki Institute of Technology, (2) Democritus University, Greece

José Mireles Jr., Universidad Autonoma de Ciudad Juarez, Mexico

Sushmita Mitra, Indian Statistical Institute, India

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

Rafael Muñoz-Salinas, University of Cordoba, Spain

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

Ould Khessal Nadir, Okanagan College, Canada

Fazel Naghdy, University of Wollongong, Australia

Tomoharu Nakashima, Osaka Prefecture University, Japan

Andreas Nearchou, University of Patras, Greece

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

Sergiu Nedevschi, Technical University of Cluj-Napoca, Romania

Maria Neves, Instituto Superior de Engenharia do Porto, Portugal

Hendrik Nijmeijer, Eindhoven University of Technology, The Netherlands

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

Urbano Nunes, University of Coimbra, Portugal

Gustavo Olague, CICESE, Mexico

PROGRAM COMMITTEE (CONT.)

José Valente de Oliveira, Universidade do Algarve, Portugal

Andrzej Ordys, Kingston University in London, Faculty of Engineering, U.K.

Djamila Ouelhadj, University of Nottingham, ASAP GROUP (Automated Scheduling, Optimisation and Planning), U.K.

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

Christos Panayiotou, University of Cyprus, Cyprus

Evangelos Papadopoulos, NTUA, Greece

Panos Pardalos, University of Florida, U.S.A.

Michel Parent, INRIA, France

Thomas Parisini, University of Trieste, Italy

Igor Paromtchik, RIKEN, Japan

Gabriella Pasi, Università degli Studi di Milano Bicocca, Italy

Witold Pedrycz, University of Alberta, Canada

Carlos Eduardo Pereira, Federal University of Rio Grande do Sul - UFRGS, Brazil

Maria Petrou, Imperial College, U.K.

J. Norberto Pires, University of Coimbra, Portugal

Marios Polycarpou, University of Cyprus, Cyprus

Marie-Noëlle Pons, CNRS, France

Libor Preucil, Czech Technical University in Prague, Czech Republic

Joseba Quevedo, Technical University of Catalonia, Spain

Robert Reynolds, Wayne State University, U.S.A.

A. Fernando Ribeiro, Universidade do Minho, Portugal

Bernardete Ribeiro, University of Coimbra, Portugal

Robert Richardson, University of Manchester, U.K.

John Ringwood, National University of Ireland (NUI), Maynooth, Ireland

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

Kurt Rohloff, BBN Technologies, U.S.A.

Juha Röning, University of Oulu, Finland

Agostinho Rosa, IST, Portugal

Hubert Roth, University Siegen, Germany

António Ruano, CSI, Portugal

Carlos Sagüés, University of Zaragoza, Spain

Mehmet Sahinkaya, University of Bath, U.K.

Antonio Sala, Universidad Politecnica de Valencia, Spain

Abdel-Badeeh Salem, Ain Shams University, Egypt

Ricardo Sanz, Universidad Politécnica de Madrid, Spain

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

Nilanjan Sarkar, Vanderbilt University, U.S.A.

Jurek Sasiadek, Carleton University, Canada

Daniel Sbarbaro, Universidad de Concepcion, Chile

Carsten Scherer, Delft University of Technology, The Netherlands

Carla Seatzu, University of Cagliari, Italy

Klaus Schilling, University Würzburg, Germany

Yang Shi, University of Saskatchewan, Canada

Michael Short, University of Leicester, U.K.

Chi-Ren Shyu, University of Missouri-Columbia, U.S.A.

Bruno Siciliano, Università di Napoli Federico II, Italy

João Silva Sequeira, Instituto Superior Técnico, Portugal

Silvio Simani, University of Ferrara, Italy

Amanda Sharkey, University of Sheffield, U.K.

Michael Small, Hong Kong Polytechnic University, China

Burkhard Stadlmann, University of Applied Sciences Wels, Austria

Tarasiewicz Stanislaw, Université Laval, Canada

Aleksandar Stankovic, Northeastern University, U.S.A.

Raúl Suárez, Universitat Politecnica de Catalunya (UPC), Spain

PROGRAM COMMITTEE (CONT.)

Ryszard Tadeusiewicz, AGH University of Science and Technology, Poland

Tianhao Tang, Shanghai Maritime University, China

Adriana Tapus, University of Southern California, U.S.A.

József K. Tar, Budapest Tech Polytechnical Institution, Hungary

Daniel Thalmann, EPFL, Switzerland

Gui Yun Tian, University of Huddersfield, U.K.

Antonios Tsourdos, Cranfield University, U.K.

Nikos Tsourveloudis, Technical University of Crete, Greece

Ivan Tyukin, RIKEN Brain Science Institute, Japan

Masaru Uchiyama, Tohoku University, Japan

Nicolas Kemper Valverde, Universidad Nacional Autónoma de México, Mexico

Marc Van Hulle, K. U. Leuven, Belgium

Annamaria R. Varkonyi-Koczy, Budapest University of Technology and Economics, Hungary

Luigi Villani, Università di Napoli Federico II, Italy

Markus Vincze, Technische Universität Wien, Austria

Bernardo Wagner, University of Hannover, Germany

Axel Walthelm, sepp.med gmbh, Germany

Lipo Wang, Nanyang Technological University, Singapore

Alfredo Weitzenfeld, ITAM, Mexico

Dirk Wollherr, Technische Universität München, Germany

Sangchul Won, Pohang University of Science and Technology, Korea

Kainam Thomas Wong, The Hong Kong Polytechnic University, China

Jeremy Wyatt, University of Birmingham, U.K.

Alex Yakovlev, University of Newcastle, U.K.

Hujun Yin, University of Manchester, U.K.

Xinghuo Yu, Royal Melbourne Institute of Techology, Australia

Du Zhang, California State University, U.S.A.

Janusz Zalewski, Florida Gulf Coast University, U.S.A.

Marek Zaremba, Universite du Quebec, Canada

Dayong Zhou, University of Oklahoma, U.S.A.

Argyrios Zolotas, Loughborough University, U.K.

Albert Zomaya, The University of Sydney, Austrália

AUXILIARY REVIEWERS

Rudwan Abdullah, University of Stirling, U.K.

Luca Baglivo, University of Padova, Italy

Prasanna Balaprakash, IRIDIA, Université Libre de Bruxelles, Belgium

João Balsa, Universidade de Lisboa, Portugal

Alejandra Barrera, ITAM, Mexico

Frederik Beutler, Intelligent Sensor-Actuator-Systems Laboratory - Universität Karlsruhe (TH), Germany

Alecio Binotto, CETA SENAI-RS, Brazil

Nizar Bouguila, Concordia University, Canada

Dietrich Brunn, Intelligent Sensor-Actuator-Systems Laboratory - Universität Karlsruhe (TH), Germany

Maria Paola Cabasino, Dip.to Ingegneria Elettrica ed Elettronica Universita' di Cagliari, Italy

Joao Paulo Caldeira, EST-IPS, Portugal

Aneesh Chauhan, Universidade de Aveiro, Portugal

Paulo Gomes da Costa, FEUP, Portugal

Xevi Cufi, University of Girona, Spain

Sérgio Reis Cunha, FEUP, Portugal

Paul Dawson, Boise State University, U.S.A.

Mahmood Elfandi, Elfateh University, Libya

AUXILIARY REVIEWERS (CONT.)

Michele Folgheraiter, Politecnico di Milano, Italy

Diamantino Freitas, FEUP, Portugal

Reinhard Gahleitner, University of Applied Sciences Wels, Austria

Nils Hagge, Leibniz Universität Hannover, Institute for Systems Engineering, Germany

Onur Hamsici, The Ohio State Univeristy, U.S.A.

Renato Ventura Bayan Henriques, UFRGS, Brazil

Matthias Hentschel, Leibniz Universität Hannover, Institute for Systems Engineering, Germany

Marco Huber, Intelligent Sensor-Actuator-Systems Laboratory - Universität Karlsruhe (TH), Germany

Markus Kemper, University of Oldenbvurg, Germany

Vesa Klumpp, Intelligent Sensor-Actuator-Systems Laboratory - Universität Karlsruhe (TH), Germany

Daniel Lecking, Leibniz Universität Hannover, Institute for Systems Engineering, Germany

Gonzalo Lopez-Nicolas, University of Zaragoza, Spain

Cristian Mahulea, University of Zaragoza, Spain

Cristian Mahulea, Dep.to Informática e Ingeneiría de Sistemas Centro Politécnico Superior, Spain

Nikolay Manyakov, K. U. Leuven, Belgium

Antonio Muñoz, University of Zaragoza, Spain

Ana C. Murillo, University of Zaragoza, Spain

Andreas Neacrhou, University of Patras, Greece

Marco Montes de Oca, IRIDIA, Université Libre de Bruxelles, Belgium

Sorin Olaru, Supelec, France

Karl Pauwels, K. U. Leuven, Belgium

Luis Puig, University of Zaragoza, Spain

Ana Respício, Universidade de Lisboa, Portugal

Pere Ridao, University of Girona, Spain

Kathrin Roberts, Intelligent Sensor-Actuator-Systems Laboratory - Universität Karlsruhe (TH), Germany

Paulo Lopes dos Santos, FEUP, Portugal

Felix Sawo, Intelligent Sensor-Actuator-Systems Laboratory - Universität Karlsruhe (TH), Germany

Frederico Schaf, UFRGS, Brazil

Oliver Schrempf, Intelligent Sensor-Actuator-Systems Laboratory - Universität Karlsruhe (TH), Germany

Torsten Sievers, University of Oldenbvurg, Germany

Razvan Solea, Institute of Systems and Robotics, University of Coimbra, Portugal

Wolfgang Steiner, University of Applied Sciences Wels, Austria

Christian Stolle, University of Oldenbvurg, Germany

Alina Tarau, Delft University of Technology, The Netherlands

Rui Tavares, University of Evora, Portugal

Paulo Trigo, ISEL, Portugal

Haralambos Valsamos, University of Patras, Greece

José Luis Villarroel, University of Zaragoza, Spain

Yunhua Wang, University of Oklahoma, U.S.A.

Florian Weißel, Intelligent Sensor-Actuator-Systems Laboratory - Universität Karlsruhe (TH), Germany

Jiann-Ming Wu, National Dong Hwa University, Taiwan

Oliver Wulf, Leibniz Universität Hannover, Institute for Systems Engineering, Germany

Ali Zayed, Seventh of April University, Libya

Yan Zhai, University of Oklahoma, U.S.A.

SELECTED PAPERS BOOK

A number of selected papers presented at ICINCO 2007 will be published by Springer, in a book entitled Informatics in Control, Automation and Robotics IV. This selection will be done by the conference co-chairs and program co-chairs, among the papers actually presented at the conference, based on a rigorous review by the ICINCO 2007 program committee members.

FOREWORD

Welcome to the 4th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2007) held at the University of Angers. 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 require Information Technology support.

In this year Conference Program we have included oral presentations (full papers and short papers) as well as posters, organized in three simultaneous tracks: "Intelligent Control Systems and Optimization", "Robotics and Automation" and "Systems Modeling, Signal Processing and Control". Furthermore, ICINCO 2007 includes 2 satellite workshops and 3 plenary keynote lectures, given by internationally recognized researchers

The two satellite workshops that are held in conjunction with ICINCO 2007 are: Third International Workshop on Multi-Agent Robotic Systems (MARS 2007) and Third International Workshop on Artificial Neural Networks and Intelligent Information Processing (ANNIIP 2007).

As additional points of interest, it is worth mentioning that the Conference Program includes a plenary panel subject to the theme "Practical Applications of Intelligent Control and Robotics" and 3 Special Sessions focused on very specialized topics.

ICINCO has received 435 paper submissions, not including workshops, from more than 50 countries, in all continents. To evaluate each submission, a double blind paper review was performed by the program committee, whose members are researchers in one of ICINCO main topic areas. Finally, only 263 papers are published in these proceedings and presented at the conference; of these, 195 papers were selected for oral presentation (52 full papers and 143 short papers) and 68 papers were selected for poster presentation. The global acceptance ratio was 60,4% and the full paper acceptance ratio was 11,9%. After the conference, some authors will be invited to publish extended versions of their papers in a journal and a short list of about thirty papers will be included in a book that will be published by Springer with the best papers of ICINCO 2007.

In order to promote the development of research and professional networks the conference includes in its social program a Town Hall Reception in the evening of May 9 (Wednesday) and a Conference and Workshops Social Event & Banquet in the evening of May 10 (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 reviewers, who helped us with their expertise and valuable time. We would also like to deeply thank the invited speakers for their excellent contribution in sharing their knowledge and vision. Finally, a word of appreciation for the hard work of the secretariat; organizing a conference of this level is a task that can only be achieved by the collaborative effort of a dedicated and highly capable team.

Commitment to high quality standards is a major aspect of ICINCO that we will strive to maintain and reinforce next year, including the quality of the keynote lectures, of the workshops, of the papers, of the organization and other aspects of the conference. We look forward to seeing more results of R&D work in Informatics, Control, Automation and Robotics at ICINCO 2008, next May, at the Hotel Tivoli Ocean Park, Funchal, Madeira, Portugal.

Janan Zaytoon

CReSTIC, URCA, France

Juan Andrade Cetto

Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Spain

Jean-Louis Ferrier

Université d'Angers, France

Joaquim Filipe

Polytechnic Institute of Setúbal / INSTICC, Portugal

CONTENTS

INVITED SPEAKERS

KEYNOTE LECTURES

ROBOTICS AND AUTOMATION

FULL PAPERS

POSTERS

SPECIAL SESSION ON INTELLIGENT VEHICLE CONTROL SYSTEMS

INVITED SPEAKERS

KEYNOTE LECTURES

REAL TIME DIAGNOSTICS, PROGNOSTICS, & PROCESS MODELING

Dimitar Filev
The Ford Motor Company
U.S.A.

Abstract: Practical and theoretical problems related to the design of real time diagnostics, prognostics, & process modeling systems are discussed. Major algorithms for autonomous monitoring of machine health in industrial networks are proposed and relevant architectures for incorporation of intelligent prognostics within plant floor information systems are reviewed. Special attention is given to the practical realization of real time structure and parameter learning algorithms. Links between statistical process control and real time modeling based on the evolving system paradigm are analyzed relative to the design of soft sensing algorithms. Examples and case studies of industrial implementation of aforementioned concepts are presented.

BRIEF BIOGRAPHY

Dr. Dimitar P. Filev is a Senior Technical Leader, Intelligent Control & Information Systems with Ford Motor Company specializing in industrial intelligent systems and technologies for control, diagnostics and decision making. He is conducting research in systems theory and applications, modeling of complex systems, intelligent modeling and control and he has published 3 books, and over 160 articles in refereed journals and conference proceedings. He holds15 granted U.S. patents and numerous foreign patents in the area of industrial intelligent systems Dr. Filev is a recipient of the '95 Award for Excellence of MCB University Press and was awarded 4 times with the Henry Ford Technology Award for development and implementation of advanced intelligent control technologies. He is Associate Editor of Int. J. of General Systems and Int. J. of Approximate Reasoning. He is a member of the Board of Governors of the IEEE Systems, Man & Cybernetics Society and President of the North American Fuzzy Information Processing Society (NAFIPS). Dr. Filev received his PhD. degree in Electrical Engineering from the Czech Technical University in Prague in 1979.

SYNCHRONIZATION OF MULTI-AGENT SYSTEMS

Mark W. Spong
Donald Biggar Willett Professor of Engineering
Professor of Electrical and Computer Engineering
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
U.S.A.

Abstract: There is currently great interest in the control of multi-agent networked systems. Applications include mobile sensor networks, teleoperation, synchronization of oscillators, UAV's and coordination of multiple robots. In this talk we consider the output synchronization of networked dynamic agents using passivity theory and considering the graph topology of the inter-agent communication. We provide a coupling control law that results in output synchronization and we discuss the extension to state synchronization in addition to output synchronization. We also consider the extension of these ideas to systems with time delay in communication among agents and obtain results showing synchronization for arbitrary time delay. We will present applications of our results in synchronization of Kuramoto oscillators and in bilateral teleoperators.

BRIEF BIOGRAPHY

Mark W. Spong received the B.A. degree, magna cum laude and Phi Beta Kappa, in mathematics and physics from Hiram College, Hiram, Ohio in 1975, the M.S. degree in mathematics from New Mexico State University in 1977, and the M.S. and D.Sc. degrees in systems science and mathematics in 1979 and 1981, respectively, from Washington University in St. Louis. Since 1984 he has been at the University of Illinois at Urbana-Champaign where he is currently a Donald Biggar Willett Distinguished Professor of Engineering, Professor of Electrical and Computer Engineering, and Director of the Center for Autonomous Engineering Systems and Robotics. Dr. Spong is Past President of the IEEE Control Systems Society and a Fellow of the IEEE. Dr. Spong's main research interests are in robotics, mechatronics, and nonlinear control theory. He has published more than 200 technical articles in control and robotics and is co-author of four books. His recent awards include the Senior U.S. Scientist Research Award from the Alexander von Humboldt Foundation, the Distinguished Member Award from the IEEE Control Systems Society, the John R. Ragazzini and O. Hugo Schuck Awards from the American Automatic Control Council, and the IEEE Third Millennium Medal.

TOWARD HUMAN-MACHINE COOPERATION

Patrick Millot
Laboratoire d'Automatique, de Mécanique et d'Informatique Industrielle et Humaine
Université de Valenciennes
France

Abstract: In human machine systems human activities are mainly oriented toward decision-making: monitoring and fault detection, fault anticipation, diagnosis and prognosis, and fault prevention and recovery. The objectives combine the human-machine system performances (production quantity and quality) as well as the global system safety. In this context human operators may have a double role: (1) a negative role as they may perform unsafe or erroneous actions on the process, (2) a positive role as they can detect, prevent or recover an unsafe process behavior due to an other operator or to automated decision makers. Two approachs to these questions are combined in a pluridisciplinary research way : (1) human engineering which aims at designing dedicated assistance tools for human operators and at integrating them into human activities through a human machine cooperation, (2) cognitive psychology and ergonomics analysing the human activities, the need for such tools and their use. This paper focuses on the concept of cooperation and proposes a framework for implementation. Examples in Air Traffic Control and in Telecommunication networks illustrate these concepts.

BRIEF BIOGRAPHY

Born in 53 he received a PhD in Automatic Control (79) an is Docteur d'Etat es Sciences (87). He is full Professor at the University of Valenciennes since 89. He conducts research on Automation Sciences, Artificial Intelligence, Supervisory Control, Human Machine Systems, Human Reliability with applications to production telecommunication and transport systems (Air Traffic Control, Car Traffic, Trains Metro.). His scientific production covers about 175 publications, collective books, conference proceedings. Research Director of 35 PhD students and 9 HDR since 89, reviewer of 50 PhD Thesis and 9 HDR from other universities. Head of the research group "Human Machine Systems" in LAMIH since 87 till 04 (25 researchers). Vice-head then head of LAMIH between 96 and 05 (222 researchers and engineers). Vice Chairman of the University of Valenciennes since October 05 in charge of research.

Scientific head or Member of the scientific board or Manager of several regional research groups on Supervisory Control (GRAISYHM 96-02) on Transport System Safety (GRRT since 87,pôle ST2 since 01 with 80 researchers of 10 labs). Member of the French Council of the Universities (96-03), member of the scientific board of the french national research group in Automation Sciences supported by CNRS (96-01). Partner of several European projects and netwoks (HCM networks 93-96, 2 projects since 02 on Urban Guided Transport Management Systems and the Network of Excellence EURNEX since 04). Member of the IFAC Technical Committee 4.5 Human Machine Systems since 00. IPC member of several International Conferences and Journals.

ROBOTICS AND AUTOMATION

FULL PAPERS

ESTIMATION PROCESS FOR TIRE-ROAD FORCES AND VEHICLE SIDESLIP ANGLE

Guillaume Baffet, Ali Charara
Heudiasyc Laboratory (UMR CNRS 6599), Université de Technologie de Compiègne
Centre de recherche Royallieu, BP20529 - 60205 Compiègne, France
gbaffet@hds.utc.fr, acharara@hds.utc.fr

Daniel Lechner
INRETS-MA Laboratory (Department of Accident Mechanism Analysis)
Chemin de la Croix Blanche, 13300 Salon de Provence, France
daniel.lechner@inrets.fr

Keywords: State observers, vehicle dynamic, sideslip angle estimation, tire-force estimation, wheel cornering stiffness estimation, linear adaptive force model.

Abstract: This study focuses on the estimation of car dynamic variables for the improvement of vehicle safety, handling characteristics and comfort. More specifically, a new estimation process is proposed to estimate longitudinal/lateral tire-road forces, sideslip angle and wheel cornering stiffness. This method uses measurements from currently-available standard sensors (yaw rate, longitudinal/lateral accelerations, steering angle and angular wheel velocities). The estimation process is separated into two blocks: the first block contains an observer whose principal role is to calculate tire-road forces without a descriptive force model, while in the second block an observer estimates sideslip angle and cornering stiffness with an adaptive tire-force model. The different observers are based on an Extended Kalman Filter (EKF). The estimation process is applied and compared to real experimental data, notably sideslip angle and wheel force measurements. Experimental results show the accuracy and potential of the estimation process.

1 INTRODUCTION

The last few years have seen the emergence in cars of active security systems to reduce dangerous situations for drivers. Among these active security systems, Anti-lock Braking Systems (ABS) and Electronic Stability Programs (ESP) significantly reduce the number of road accidents. However, these systems may improved if the dynamic potential of a car is well known. For example, information on tire-road friction means a better definition of potential trajectories, and therefore a better management of vehicle controls. Nowadays, certain fundamental data relating to vehicle-dynamics are not measurable in a standard car for both technical and economic reasons. As a consequence, dynamic variables such as tire forces and sideslip angle must be observed or estimated.

Vehicle-dynamic estimation has been widely discussed in the literature, e.g. ((Kiencke and Nielsen, 2000), (Ungoren et al., 2004), (Lechner, 2002), (Stephant et al., 2006)). The vehicle-road system is usually modeled by combining a vehicle model with a tire-force model in one block. One particularity of this study is to separate the estimation modeling into two blocks (shown in figure 1), where the first block concerns the car body dynamic while the second is devoted to the tire-road interface dynamic. The first block contains an Extended Kalman Filter (denoted as $O_{1,4w}$) constructed with a four-wheel vehicle

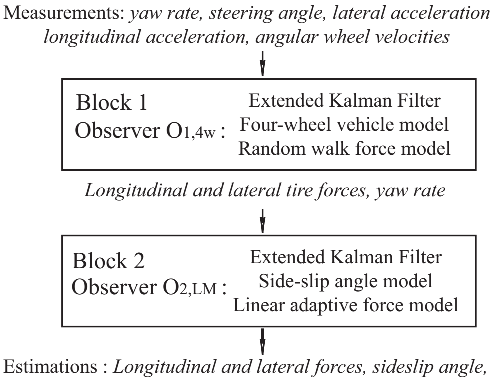

Measurements: *yaw rate, steering angle, lateral acceleration*
longitudinal acceleration, angular wheel velocities
Block 1
Observer O1,4w :
Extended Kalman Filter
Four-wheel vehicle model
Random walk force model
Longitudinal and lateral tire forces, yaw rate
Block 2
Observer O2,LM :
Extended Kalman Filter
Side-slip angle model
Linear adaptive force model
Estimations : *Longitudinal and lateral forces, sideslip angle,*
yaw rate, cornering stiffness

Figure 1: Estimation process. Observers $O_{1,4w}$ and $O_{2,LAM}$.

model and a random walk force model. The first observer $O_{1,4w}$ estimates longitudinal/lateral tire forces and yaw rate, which are inputs to the observer in the second block (denoted as $O_{2,LAM}$). This second observer is developed from a sideslip angle model and a linear adaptive force model.

Some studies have described observers which take road friction variations into account ((Lakehal-ayat et al., 2006), (Rabhi et al., 2005), (Ray, 1997)). In the works of (Lakehal-ayat et al., 2006) road friction is considered as a disturbance. Alternatively, as in (Rabhi et al., 2005), the tire-force parameters are identified with an observer, while in (Ray, 1997) tire forces are modeled with an integrated random walk model. In this study a linear adaptive tire force model is proposed (in block 2) with an eye to studying road friction variations.

The rest of the paper is organized as follows. The second section describes the vehicle model and the observer $O_{1,4w}$ (block 1). Next, the third section presents the sideslip angle and cornering stiffness observer ($O_{2,LAM}$ in block 2). In the fourth section an observability analysis is performed. The fifth section provides experimental results: the two observers are evaluated with respect to sideslip angle and tire force measurements. Finally, concluding remarks are given in section 6.

2 BLOCK 1: OBSERVER FOR TIRE-ROAD FORCE

This section describes the first observer $O_{1,4w}$ constructed from a four-wheel vehicle model (figure 2), where $\dot{\psi}$ is the yaw rate, β the center of gravity sideslip angle, V_g the center of gravity velocity, and L_1 and L_2 the distance from the vehicle center of gravity to the front and rear axles respectively. $F_{x,y,i,j}$ are the longitudinal and lateral tire-road forces, $\delta_{1,2}$ are the front left and right steering angles respectively, and E is the vehicle track (lateral distance from wheel to wheel).

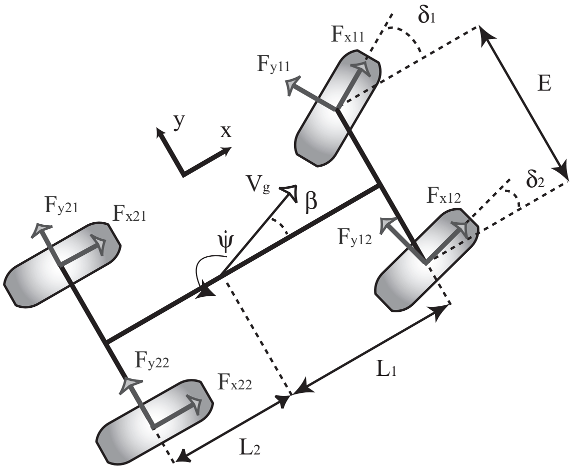


Figure 2: Four-wheel vehicle model.

In order to develop an observable system (notably in the case of null steering angles), rear longitudinal forces are neglected relative to the front longitudinal forces. The simplified equation for yaw acceleration (four-wheel vehicle model) can be formulated as the following dynamic relationship ($O_{1,4w}$ model):

$$\ddot{\psi} = \frac{1}{I_z}\begin{bmatrix} L_1[F_{y11}\cos(\delta_1)+F_{y12}\cos(\delta_2) \\ +F_{x11}\sin(\delta_1)+F_{x12}\sin(\delta_2)] \\ -L_2[F_{y21}+F_{y22}] \\ +\frac{E}{2}[F_{y11}\sin(\delta_1)-F_{y12}\sin(\delta_2) \\ +F_{x12}\cos(\delta_2)-F_{x11}\cos(\delta_1)] \end{bmatrix}, \quad (1)$$

where m the vehicle mass and I_z the yaw moment of inertia. The different force evolutions are modeled with a random walk model:

$$[\dot{F_{xij}}, \dot{F_{yij}}] = [0,0], \quad i=1,2 \quad j=1,2. \quad (2)$$

The measurement vector Y and the measurement model are:

$$\begin{array}{l} Y = [\dot{\psi}, \gamma_y, \gamma_x] = [Y_1, Y_2, Y_3], \\ Y_1 = \dot{\psi}, \\ Y_2 = \frac{1}{m}[F_{y11}\cos(\delta_1)+F_{y12}\cos(\delta_2) \\ +(F_{y21}+F_{y22})+F_{x11}\sin(\delta_1)+F_{x12}\sin(\delta_2)], \\ Y_3 = \frac{1}{m}[-F_{y11}\sin(\delta_1)-F_{y12}\sin(\delta_2) \\ +F_{x11}\cos(\delta_1)+F_{x12}\cos(\delta_2)], \end{array} \quad (3)$$

where γ_x and γ_y are the longitudinal and lateral accelerations respectively.

The $O_{1,4w}$ system (association between equations (1), random walk force equation (2), and the measurement equations (3)) is not observable in the case where F_{y21} and F_{y22} are state vector components. For example, in equation (1, 2, 3) there is no relation allowing to distinguish the rear lateral forces F_{y21} and F_{y22} in the sum (F_{y21}+F_{y22}): as a consequence only the sum ($F_{y2} = F_{y21}+F_{y22}$) is observable. Moreover, when driving in a straight line, yaw rate is small, δ_1 and δ_2 are approximately null, and hence there is no significant knowledge in equation (1, 2, 3) differentiating F_{y11} and F_{y12} in the sum ($F_{y11}+F_{y12}$), so only the sum ($F_{y1} = F_{y11}+F_{y12}$) is observable. These observations lead us to develop the $O_{1,4w}$ system with a state vector composed of force sums:

$$X = [\dot{\psi}, F_{y1}, F_{y2}, F_{x1}], \quad (4)$$

where F_{x1} is the sum of front longitudinal forces ($F_{x1} = F_{x11}+F_{x12}$). Tire forces and force sums are associated according to the dispersion of vertical forces:

$$F_{x11} = \frac{F_{z11}F_{x1}}{F_{z12}+F_{z11}}, \quad F_{x12} = \frac{F_{z12}F_{x1}}{F_{z12}+F_{z11}}, \tag{5}$$

$$F_{y11} = \frac{F_{z11}F_{y1}}{F_{z12}+F_{z11}}, \quad F_{y12} = \frac{F_{z12}F_{y1}}{F_{z12}+F_{z11}}, \tag{6}$$

$$F_{y21} = \frac{F_{z21}F_{y2}}{F_{z22}+F_{z21}}, \quad F_{y22} = \frac{F_{z22}F_{y2}}{F_{z22}+F_{z21}}, \tag{7}$$

where F_{zij} are the vertical forces. These are calculated, neglecting roll and suspension movements, with the following load transfer model:

$$F_{z11} = \frac{L_2 mg - h_{cog} m\gamma_x}{2(L_1+L_2)} - \frac{L_2 h_{cog} m\gamma_y}{(L_1+L_2)E}, \tag{8}$$

$$F_{z12} = \frac{L_2 mg - h_{cog} m\gamma_x}{2(L_1+L_2)} + \frac{L_2 h_{cog} m\gamma_y}{(L_1+L_2)E}, \tag{9}$$

$$F_{z21} = \frac{L_1 mg + h_{cog} m\gamma_x}{2(L_1+L_2)} - \frac{L_2 h_{cog} m\gamma_y}{(L_1+L_2)E}, \tag{10}$$

$$F_{z22} = \frac{L_1 mg + h_{cog} m\gamma_x}{2(L_1+L_2)} + \frac{L_2 h_{cog} m\gamma_y}{(L_1+L_2)E}, \tag{11}$$

h_{cog} being the center of gravity height and g the gravitational constant. The load transfer model follows the assumption of the superposition principle of independent longitudinal and lateral acceleration contributions (Lechner, 2002). The input vectors U of $O_{1,4w}$ observer is:

$$U = [\delta_1, \delta_2, F_{z11}, F_{z12}, F_{z21}, F_{z22}]. \tag{12}$$

As regards the vertical force inputs, these are calculated from lateral and longitudinal accelerations with the load transfer model.

3 BLOCK 2: OBSERVER FOR SIDESLIP ANGLE AND CORNERING STIFFNESS

This section presents the observer $O_{2,LAM}$ constructed from a sideslip angle model and a tire-force model. The sideslip angle model is based on the single-track model (Segel, 1956), with neglected rear longitudinal force:

$$\dot{\beta} = \frac{F_{x1}\sin(\delta-\beta) + F_{y1}\cos(\delta-\beta) + F_{y2}\cos(\beta)}{mV_g} - \dot{\psi}. \tag{13}$$

Rear and front sideslip angles are calculated as:

$$\begin{aligned} \beta_1 &= \delta - \beta - L_1\dot{\psi}/V_g, \\ \beta_2 &= -\beta + L_2\dot{\psi}/V_g, \end{aligned} \tag{14}$$

where δ is the mean of front steering angles.

The dynamic of the tire-road contact is usually formulated by modeling the tire-force as a function of the slip between tire and road ((Pacejka and Bakker, 1991), (Kiencke and Nielsen, 2000), (Canudas-De-Wit et al., 2003)). Figure 3 illustrates different lateral tire-force models (linear, linear adaptive and Burckhardt for various road surfaces (Kiencke and Nielsen, 2000)). In this study lateral wheel slips are assumed to be equal to the wheel sideslip angles. In current driving situations, lateral tire forces may be considered linear with respect to sideslip angle (linear model):

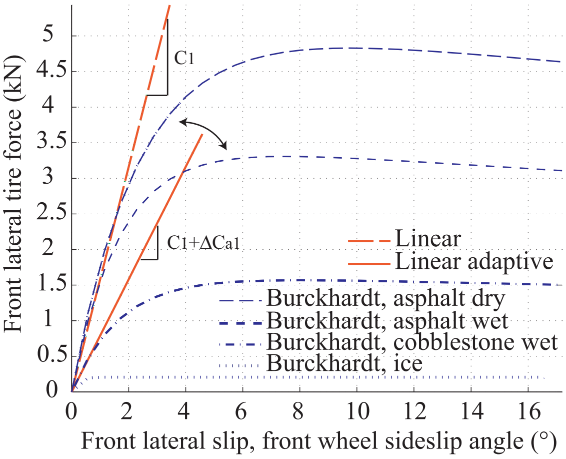


Figure 3: Lateral tire force models: linear, linear adaptive, Burckhardt for various road surfaces.

$$F_{yi}(\beta_i) = C_i\beta_i, \qquad i = 1,2, \tag{15}$$

where C_i is the wheel cornering stiffness, a parameter closely related to tire-road friction.

When road friction changes or when the nonlinear tire domain is reached, ”real” wheel cornering stiffness varies. In order a take the wheel cornering stiffness variations into account, we proposed an adaptive tire-force model (named the linear adaptive tire-force model, illustrated in figure 3). This model is based on the linear model at which a readjustment variable ΔC_{ai} is added to correct wheel cornering stiffness errors:

$$F_{yi}(\beta_i) = (C_i + \Delta C_{ai})\beta_i. \tag{16}$$

The variable ΔC_{ai} is included in the state vector of the $O_{2,LAM}$ observer and it evolution equation is formulated according to a random walk model ($\Delta\dot{C_{ai}} = 0$). State $X' \in R^3$, input $U' \in R^4$ and measurement $Y' \in R^3$ are chosen as:

$$\begin{aligned} X' &= [x'_1, x'_2, x'_3] = [\beta, \Delta C_{a1}, \Delta C_{a2}], \\ U' &= [u'_1, u'_2, u'_3, u'_4] = [\delta, \dot{\psi}, V_g, F_{x1}], \\ Y' &= [y'_1, y'_2, y'_3] = [F_{y1}, F_{y2}, \gamma_y]. \end{aligned} \tag{17}$$

The measurement model is

$$\begin{array}{rcl} y'_1 & = & (C_1 + x'_2)\beta_1, \\ y'_2 & = & (C_2 + x'_3)\beta_2, \\ y'_3 & = & \frac{1}{m}[(C_1 + x'_2)\beta_1 \cos(u'_1) + (C_2 + x'_3)\beta_2 \\ & + & u'_4 \sin(u'_1)]. \end{array} \tag{18}$$

where

$$\begin{array}{rcl} \beta_1 & = & u'_1 - x'_1 - L_1 u'_2 / u'_3, \\ \beta_2 & = & -x'_1 + L_2 u'_2 / u'_3. \end{array} \tag{19}$$

Consider the state estimation denoted as $\widehat{X}' = [\widehat{x_1}', \widehat{x_2}', \widehat{x_3}']$, the state evolution model of $O_{2,LAM}$ is:

$$\begin{array}{rcl} \dot{\widehat{x_1}}' & = & \frac{1}{mu_3}[u'_4 \sin(u'_1 - \widehat{x_1}') + F_{yw1,aux} \cos(u'_1 - \widehat{x_1}') \\ & + & F_{yw2,aux} \cos(\widehat{x_1}')] - u'_2, \\ \dot{\widehat{x_2}}' & = & 0, \\ \dot{\widehat{x_3}}' & = & 0, \end{array} \tag{20}$$

where the auxiliary variables $F_{yw1,aux}$ and $F_{yw2,aux}$ are calculated as:

$$\begin{array}{rcl} F_{yw1,aux} & = & (C_1 + \widehat{x_2}')(u'_1 - \widehat{x_1}' - L_1 u'_2 / u'_3), \\ F_{yw2,aux} & = & (C_2 + \widehat{x_3}')(-\widehat{x_1}' + L_2 u'_2 / u'_3). \end{array} \tag{21}$$

4 OBSERVABILITY

The different observers ($O_{1,4w}$, $O_{2,LAM}$) were developed according to an extended Kalman filter method (Kalman, 1960), (Mohinder and Angus, 1993).

The two observer systems are nonlinear, so the two observability functions were calculated using a lie derivative method (Nijmeijer and der Schaft, 1990). Ranks of the two observability functions corresponded to the state vector dimensions, so systems $O_{1,4w}$ and $O_{2,LAM}$ were locally observable. Concerning the observability of the complete systems ($O_{1,4w}$ and $O_{2,LAM}$), a previous work (Baffet et al., 2006a) showed that a similar system (in one block) is locally observable.

5 EXPERIMENTAL RESULTS

The experimental vehicle (see figure 4) is a Peugeot 307 equipped with a number of sensors including GPS, accelerometer, odometer, gyrometer, steering angle, correvit and dynamometric hubs. Among these sensors, the correvit (a non-contact optical sensor) gives measurements of rear sideslip angle and vehicle velocity, while the dynamometric hubs are wheel-force transducers.

This study uses an experimental test representative of both longitudinal and lateral dynamic behaviors. The vehicle trajectory and the acceleration diagram are shown in figure 5. During the test, the vehicle first accelerated up to $\gamma_x \approx 0.3g$, then negotiated a slalom at an approximate velocity of $12m/s$ ($-0.6g < \gamma_y < 0.6g$), before finally decelerating to $\gamma_x \approx -0.7g$. The results are presented in two forms: figures of estimations/measurements and tables of normalized errors. The normalized error ε_z for an estimation z is defined in (Stephant et al., 2006) as

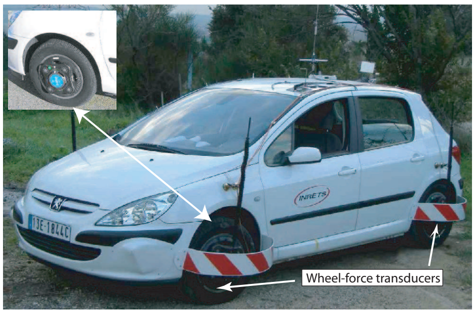


Figure 4: Laboratory's experimental vehicle.

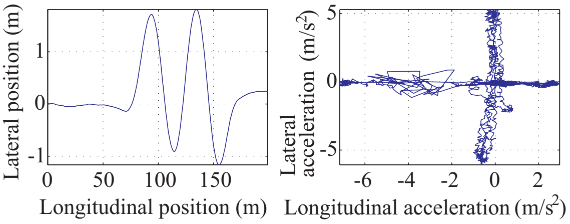


Figure 5: Experimental test, vehicle positions, acceleration diagram.

$$\varepsilon_z = 100(\|z - z_{measurement}\|)/(\max \|z_{measurement}\|). \tag{22}$$

5.1 Block 1: Observer $O_{1,4w}$ Results

Figure 6 and table 1 present $O_{1,4w}$ observer results. The state estimations were initialized using the maximum value of the measurements during the test (for instance, the estimation of the front lateral force F_{y1}

Table 1: Maximum absolute values, $O_{1,4w}$ normalized mean errors and normalized standard deviation (Std).

	max $\|\|$	Mean	Std
F_{y1}	5816 N	3.1%	4.0%
F_{y2}	3782 N	2.9%	5.4%
F_{x1}	9305 N	3.1%	4.1%
$\dot{\psi}$	24.6 $^o/s$	0.4%	2.6%

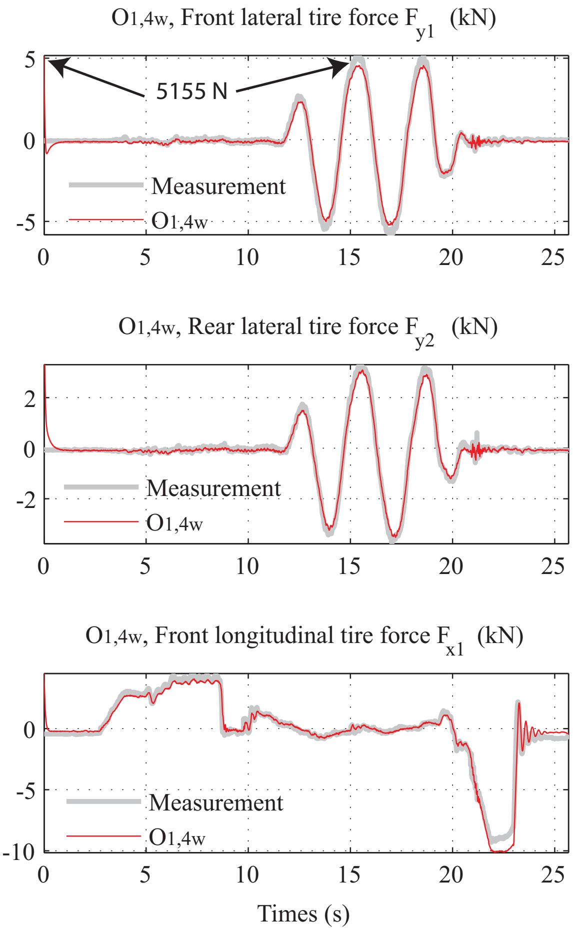


Figure 6: Experimental test. $O_{1,4w}$ results in comparison with measurements.

was set to 5155 N). In spite of these false initializations the estimations converge quickly to the measured values, showing the good convergence properties of the observer. Moreover, the $O_{1,4w}$ observer produces satisfactory estimations close to measurements (normalized mean and standard deviations errors are less than 7 %). These good experimental results confirm that the observer approach may be an appropriate way for the estimation of tire-forces.

5.2 Block 2: Observer $O_{2,LAM}$ Results

During the test, (F_{x1},F_{y1},F_{y2}) inputs of $O_{2,LAM}$ were originally those from the $O_{1,4w}$ observer. The V_g input of $O_{2,LAM}$ was obtained from the wheel angular velocities. In order to demonstrate the improvement provided by the observer using the *linear adaptive force model* ($O_{2,LAM}$, equation 16), another observer constructed with a *linear fixed force model* is used in comparison (denoted O_{rl}, equation 15, described in (Baffet et al., 2006b)). The robustness of the two observers is tested with respect to tire-road friction variations by performing the tests with different cornering stiffness parameters ($[C_1,C_2]*0.5,1,1.5$). The observers were evaluated for the same test presented in section 5.

Figure 7 shows the estimation results of observer O_{rl} for rear sideslip angle. Observer O_{rl} gives good results when cornering stiffnesses are approximately known ($[C_1,C_2]*1$). However, this observer is not robust when cornering stiffnesses change ($[C_1,C_2]*0.5,1.5$).

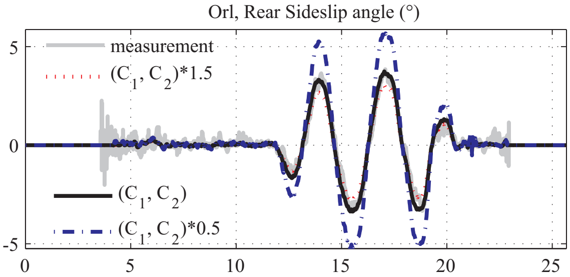


Figure 7: Observer O_{rl} using a fixed linear force model, rear sideslip angle estimations with different cornering stiffness settings.

Figure 8 and table 2 show estimation results for the adaptive observer $O_{2,LAM}$. The performance robustness of $O_{2,LAM}$ is very good, since sideslip angle is well estimated irrespective of cornering stiffness settings. This result is confirmed by the normalized mean errors (Table 2) which are approximately constant (about 7 %). The front and rear cornering stiffness estimations ($C_i+\Delta C_i$) converge quickly to the same values after the beginning of the slalom at 12 s.

Table 2: Observer O_{LAM}, rear sideslip angle estimation results, maximum absolute value, normalized mean errors.

$O_{2,LAM}$	$0.5(C_1,C_2)$	(C_1,C_2)	$1.5(C_1,C_2)$
max $\|\beta_2\|$	3.0^o	3.0^o	3.0^o
Mean	7.4%	7.0%	7.2%

6 CONCLUSIONS AND FUTURE WORK

This study deals with two vehicle-dynamic observers constructed for use in a two-block estimation process. Experimental results show that the first observer $O_{1,4w}$ gives force estimations close to the measurements, and the second observer $O_{2,LAM}$ provides good

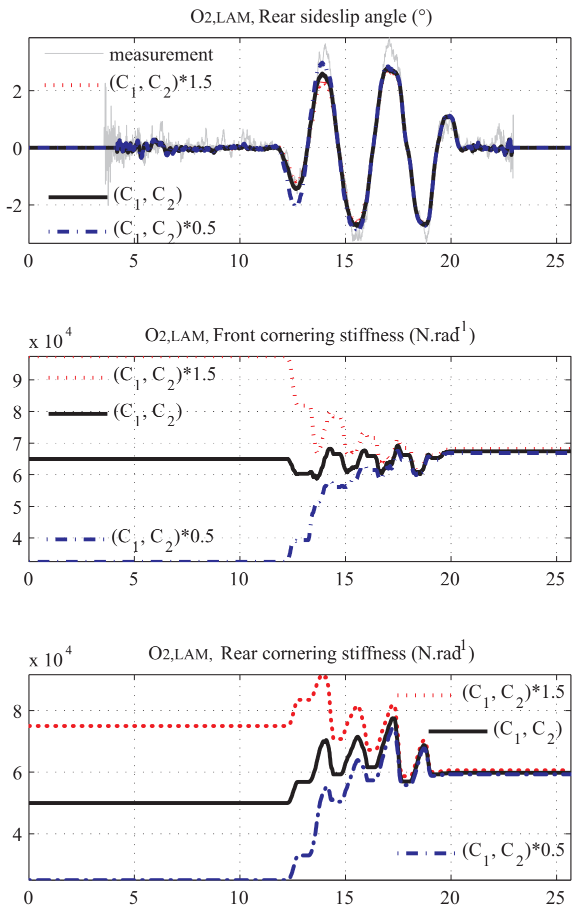


Figure 8: $O_{2,LAM}$ adaptive observer, Sideslip angle estimation results, Front and rear cornering stiffness estimations $C_i + \Delta C_i$, with different cornering stiffness settings.

sideslip angle estimations with good robustness properties relative to cornering stiffness changes. This result justifies the use of an adaptive tire-force model to take into account road friction changes.
Future studies will improve vehicle/road models in order to widen validity domains for observers. Subsequent vehicle/road models will take into account roll and vertical dynamics.

REFERENCES

Baffet, G., Stephant, J., and Charara, A. (2006a). Sideslip angle lateral tire force and road friction estimation in simulations and experiments. Proceedings of the IEEE conference on control application CCA Munich Germany.

Baffet, G., Stephant, J., and Charara, A. (2006b). Vehicle sideslip angle and lateral tire-force estimations in standard and critical driving situations: Simulations and experiments. Proceedings of the 8th International Symposium on Advanced Vehicle Control AVEC2006 Taipei Taiwan.

Canudas-De-Wit, C., Tsiotras, P., Velenis, E., Basset, M., and Gissinger, G. (2003). Dynamic friction models for road/tire longitudinal interaction. volume 39, pages 189–226. Vehicle System Dynamics.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. volume 82, pages 35–45. Transactions of the ASME - PUBLISHER of Basic Engineering.

Kiencke, U. and Nielsen, L. (2000). Automotive control system. Springer.

Lakehal-ayat, M., Tseng, H. E., Mao, Y., and j. Karidas (2006). Disturbance observer for lateral velocity estimation. Proceedings of the 8th International Symposium on Advanced Vehicle Control AVEC2006 Taipei Taiwan.

Lechner, D. (2002). Analyse du comportement dynamique des vehicules routiers legers: developpement d'une methodologie appliquee a la securite primaire. Ph.D. dissertation Ecole Centrale de Lyon France.

Mohinder, S. G. and Angus, P. A. (1993). Kalman filtering theory and practice. Prentice hall.

Nijmeijer, H. and der Schaft, A. J. V. (1990). Nonlinear dynamical control systems. Springer-Verlag.

Pacejka, H. B. and Bakker, E. (1991). The magic formula tyre model. pages 1–18. Int. colloq. on tyre models for vehicle dynamics analysis.

Rabhi, A., M'Sirdi, N. K., Zbiri, N., and Delanne, Y. (2005). Vehicle-road interaction modelling for estimation of contact forces. volume 43, pages 403–411. Vehicle System Dynamics.

Ray, L. (1997). Nonlinear tire force estimation and road friction identification : Simulation and experiments. volume 33, pages 1819–1833. Automatica.

Segel, M. L. (1956). Theorical prediction and experimental substantiation of the response of the automobile to steering control. volume 7, pages 310–330. automobile division of the institut of mechanical engineers.

Stephant, J., Charara, A., and Meizel, D. (Available online 5 June 2006). Evaluation of a sliding mode observer for vehicle sideslip angle. Control Engineering Practice.

Ungoren, A. Y., Peng, H., and Tseng, H. E. (2004). A study on lateral speed estimation methods. volume 2, pages 126–144. Int. J. Vehicle Autonomous Systems.

INTEGRATED DESIGN OF A MECHATRONIC SYSTEM
The Pressure Control in Common Rails

Paolo Lino and Bruno Maione
Dipartimento di Elettrotecnica ed Elettronica, Politecnico di Bari, via Re David 200, 70125, Bari, Italy
lino@deemail.poliba.it, maione@poliba.it

Keywords: Mechatronic systems, Virtual prototyping, AMESim$^{®}$, Modelling, Automotive control.

Abstract: This paper describes the integrated design of the pressure control in a common-rail injection system. Mechanical elements and the embedded controller are considered as a whole, using a multi-disciplinary approach to modelling and simulation. The virtual prototype, which provides the detailed geometrical/physical model of the mechanical parts, plays the role of a surrogate of a reference hardware prototype in reduced-order modelling, validation, and/or in tuning the control parameters. The results obtained by the proposed approach are compared and validated by experiments.

1 INTRODUCTION

The innovative characters of mechatronic systems lie with a special design process thinking about the mechanical parts and the embedded controllers as a whole. On the contrary, the traditional design of devices combining mechanical and electronic elements considers components belonging to different physical domain separately. Moreover controllers are conceived for already existing plants. Hence the physical properties and the dynamical behaviour of parts, in which energy conversion plays a central role, are not determined by the choices of the control engineers and therefore are of little concern to them. Their primary interests, indeed, are signal processing and information management, computer power requirements, choice of sensors and sensor locations, and so on. So it can happen that poorly designed mechanical parts do never lead to good performance, even in presence of advanced controllers. On the other hand, a poor knowledge of how controllers can directly influence and balance for defects or weaknesses in mechanical components does not help in achieving quality and good performance of the whole process. Aiming at an integrated design, in mechatronic systems the interactions among mechanical, electronic and information processing elements are considered in all the design steps, beginning with the early stages (Stobart et al., 1999),(van Amerongen and Breedveld, 2003).

If the design choices must be made before assembling interacting parts of a physical prototype and if these parts belong to different physical domains (such as mechanics, electronics, hydraulics and control) the virtual prototyping approaches, which are surrogates for preliminary physical realizations, face many difficulties. The first one is due to the fact that each physical domain has specific modelling and simulation tools. Unfortunately, indeed, there is a gap between tools for evaluating the designed components in different domains. For example, mechanical engineers usually refer to finite element approaches/packages as an important tool for dimensioning mechanical parts and for evaluating their dynamical properties. Only after a complex order-reduction process (modal analysis) the so obtained models are reduced in a form suitable for the control analysis and design. On the other hand, control modelling tools are based on operators (transfer functions, frequency response) and/or on state equations descriptions, which, in many cases, do not have a straightforward connection with the critical parameters of physical design. However, it is clear that an appropriate modelling and simulation approach can not be fitted into the limitations of one formalism at time, particularly in the early stages of the design process. Hence, it is necessary a combination of different methodologies in a multi-formalism approach to modelling supported by an appropriate simulation environment.

In mechatronic applications, the Bond Graphs introduced by (Paynter, 1960) provide a continuous system modelling approach, oriented to a class of interconnected and completely different systems and targeted to many user groups. Bond Graphs are very useful in analysing and designing components built from different energetic media and can represent dynamical systems at higher level of abstraction than differential equations (van Amerongen and Breedveld, 2003), (Ferretti et al., 2004).

In this paper, we referred to AMESim® (Advanced Modelling Environment for performing Simulations of engineering systems) (IMAGINE S.A., 2004) a Bond Graph-based, multi-domain modelling/optimization tool for the virtual prototyping of the physical/geometrical characteristics of a Compressed Natural Gas (CNG) injection system. In a first step, we used this tool to obtain a virtual prototype, as similar as possible to the actual final hardware. Then, with reference to this prototype, we also determined a reduced order model in form of transfer function and/or state space model, more suitable for analytical (or empirical) tuning of the pressure controller of the CNG injection systems. Using the virtual prototype in these early design stages enabled the evaluation of the influence of the geometrical/physical alternatives on the reduced model used for the controller tuning. Then, based on this reduced model, the controller settings were designed and adjusted in accordance with the early stages of the mechanical design process. Finally, the detailed physical/geometric model of the mechanical parts, created by the AMESim package, was exported ad used as a module in a simulation program, which enabled the evaluation of the controller performance in the closed-loop system. In other words, the detailed simulation model surrogated for a real hardware.

2 THE INTEGRATED DESIGN APPROACH

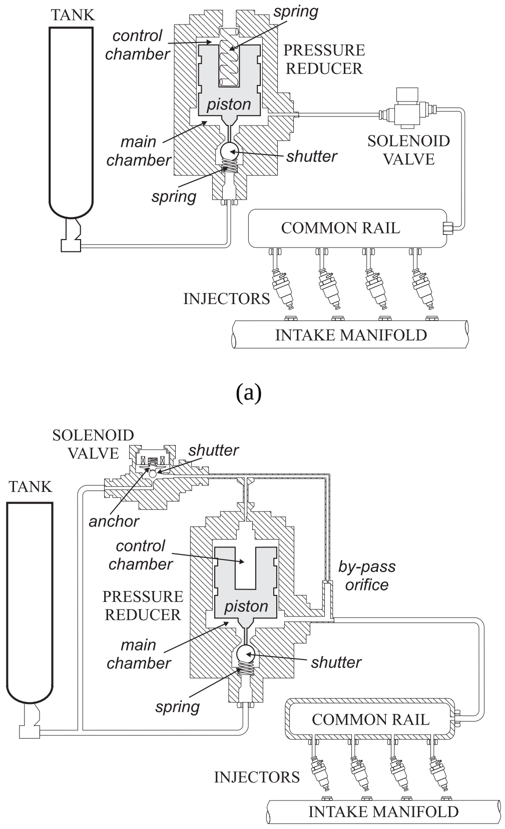
In this paper, we consider the opportunity of integrating different models, at different level of details, and different design tools, to optimize the design of the mechanical and control systems as a whole. To sum up, the integrated design can take advantage of peculiarities of different specific software packages. The effectiveness of the approach is illustrated by means of a case study, the pressure control of a CNG injection system for internal combustion engines, which represents a benchmark for the evaluation of performances of the approach. Experimental results give a feedback of benefits of the integration of the mechanical and control subsystems design.

The integrated design involves the development of a virtual prototype of the considered system using an advanced modelling tool (AMESim in this paper), which is employed for the analysis of the system performances during the different developing steps. Actually, the virtual prototype could be assumed as a reliable model of the real system. Further, a low order analytical model of the real system can be developed to simplify the controller design. Since AMESim can represent physical phenomena at different level of details, it is exploited to verify assumptions in building the analytical model. Then, the control system is designed using a specific software package, i.e. MATLAB/Simulink®, and tested on the virtual prototype. The virtual prototype allows to perform safer, less expensive, and more reliable tests than using the real system.

3 APPLICATION EXAMPLE

We consider a system composed of the following elements (Fig. 1): a fuel tank, storing high pressure gas, a mechanical pressure reducer, a solenoid valve and the fuel metering system, consisting of a common rail and four electro-injectors. Two different configurations were compared for implementation, with different arrangements of the solenoid valve affecting system performances (i.e. cascade connection, figure 1(a), and parallel connection, Fig.1(b), respectively). Detailed AMESim models were developed for each of them, providing critical information for the final choice. Few details illustrate the injection operation for both layouts.

With reference to Fig. 1(a), the pressure reducer receives fuel from the tank at a pressure in the range between 200 and 20 bars and reduces it to a value of about 10 bar. Then the solenoid valve suitably regulates the gas flow towards the common rail to control pressure level and to damp oscillations due to injections. Finally, the electronically controlled injectors send the gas to the intake manifold for obtaining the proper fuel air mixture. The injection flow only depends on rail pressure and injection timings, which are precisely driven by the Electronic Control Unit (ECU). The variable inflow section of the pressure reducer is varied by the axial displacement of a spherical shutter coupled with a moving piston. Piston and shutter dynamics are affected by the applied forces: gas pressure in a main chamber acts on the



(a)

(b)

Figure 1: Block schemes of the common rail CNG injection systems; (a) cascade connection of solenoid valve; (b) parallel connection of solenoid valve.

piston lower surface pushing it at the top, and elastic force of a preloaded spring holden in a control chamber pushes it down and causes the shutter to open. The spring preload value sets the desired equilibrium reducer pressure: if the pressure exceeds the reference value the shutter closes and the gas inflow reduces, preventing a further pressure rise; on the contrary, if the pressure decreases, the piston moves down and the shutter opens, letting more fuel to enter and causing the pressure to go up in the reducer chamber (see (Maione et al., 2004) for details).

As for the second configuration (Fig. 1(b)), the fuel from the pressure reducer directly flows towards the rail, and the solenoid valve regulates the intake flow in a secondary circuit including the control chamber. The role of the force applied by the preloaded spring of control chamber is now played by the pressure force in the secondary circuit, which can be controlled by suitably driving the solenoid valve. When the solenoid valve is energized, the fuel enters the control chamber, causing the pressure on the upper surface of the piston to build up. As a consequence, the piston is pushed down with the shutter, letting more fuel to enter in the main chamber, where the pressure increases. On the contrary, when the solenoid valve is non-energized, the pressure on the upper side of the piston decreases, making the piston to raise and the main chamber shutter to close under the action of a preloaded spring (see (Lino et al., 2006) for details).

On the basis of a deep analysis performed on AMESim virtual prototypes the second configuration was chosen as a final solution, because it has advantages in terms of performances and efficiency. To sum up, it guarantees faster transients as the fuel can reach the common rail at a higher pressure. Moreover, leakages involving the pressure reducer due to the allowance between cylinder and piston are reduced by the lesser pressure gradient between the lower and upper piston surfaces. Finally, allowing intermediate positions of the shutter in the pressure reducer permits a more accurate control of the intake flow from the tank and a remarkable reduction of the pressure oscillations due to control operations. A detailed description of the AMESim model of the system according the final layout is presented in section 4 (Fig. 2).

4 AMESIM MODEL OF THE CNG INJECTION SYSTEM

AMESim is a virtual prototyping software produced by IMAGINE S.A., which is oriented to lumped parameter modelling of physical elements, interconnected by ports enlightening the energy exchanges between pairs of elements and between an element and its environment. AMESim enables the modelling of components from different physical domains and the integration of these different elements in an overall system framework. It also guarantees a flexible architecture, capable of including new components defined by the users.

By assumption, the pressures distribution within the control chamber, the common rail and the injectors is uniform, and the elastic deformations of solid parts due to pressure changes are negligible. The pipes are considered as incompressible ducts with friction and a non uniform pressure distribution. Temperature variations are taken into account, affecting the pressure dynamics in each subcomponent. Besides, only heat exchanges through pipes are considered, by properly computing a thermal exchange coefficient. The tank pressure plays the role of a maintenance input, and it is modelled by a constant

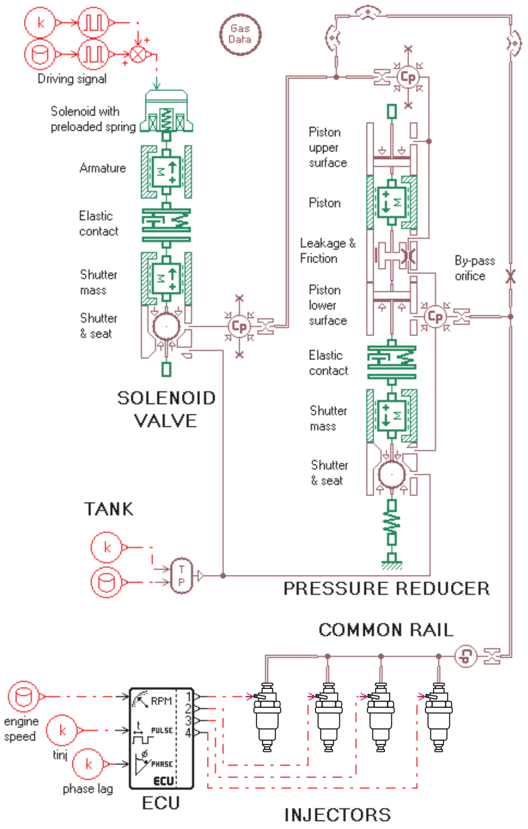


Figure 2: AMESim model of the CNG injection system.

pneumatic pressure source. To simplify the AMESim model construction some supercomponents have been suitably created, collecting elements within a single one.

4.1 Pressure Reducer and Common Rail

The main components for modelling the pressure reducer are the *Mass block with stiction and coulomb friction and end stops*, which computes the piston and the shutter dynamics through the Newton's second law of motion, a *Pneumatic ball poppet with conical seat*, two *Pneumatic piston*, and an *Elastic contact* modelling the contact between the piston and the shutter. The *Pneumatic piston* components compute the pressure forces acting upon the upper and lower piston surfaces. The viscous friction and leakage due to contact between piston and cylinder are taken into account through the *Pneumatic leakage and viscous friction* component, by specifying the length of contact, the piston diameter and the clearance. Finally, a *Variable volume with pneumatic chamber* is used to compute the pressure dynamics as a function of temperature T and intake and outtake flows $\dot{m}_{in}$, $\dot{m}_{out}$, as well as of volume changes due to mechanical part motions, according to the following equation:

$$\dot{p} = \frac{RT}{V}\left(\dot{m}_{in} - \dot{m}_{out} + \rho\frac{dV}{dt}\right), \qquad (1)$$

where p is the fuel pressure, ρ the fuel density and V the taken up volume. The same component is used to model the common rail by neglecting the volume changes.

Both pressure and viscous stresses contribute to drag forces acting on a body immersed in a moving fluid. In particular, the total drag acting on a body is the sum of two components: the pressure or form drag, due to pressure gradient, and the skin friction or viscous drag, i.e. *Drag force = Form drag + Skin friction drag*. By introducing a drag coefficient C_D depending on Reynolds number, the drag can be expressed in terms of the relative speed v (Streeter et al., 1998):

$$Drag = C_D \rho A v^2/2. \qquad (2)$$

Moving shutters connecting two different control volumes are subject to both form drag and skin friction drag. The former one is properly computed by AMESim algorithms for a variety of shutters, considering different poppet and seat shapes. As for the latter, it is computed as a linear function of the fluid speed by the factor of proportionality. It can be obtained by noting that for a spherical body it holds *Form Drag* $= 2\pi D\mu v$ (Streeter et al., 1998), being μ the absolute viscosity and D the shutter diameter. The moving anchor in the solenoid valve experiences a viscous drag depending on the body shape. In particular, the drag coefficient of a disk is $C_D = 1.2$, and the skin friction drag can be computed using eq. 2. Since, by hypothesis, the anchor moves within a fluid with uniform pressure distribution, the form drag is neglected.

4.2 Pipes

The continuity and momentum equations are used to compute pressures and flows through pipes so as to take into account wave propagation effects. In case of long pipes with friction, a system of nonlinear partial differential equations is obtained, which is implemented in the *Distributive wave equation submodel of pneumatic pipe* component from the pneumatic library. This is the case of pipes connecting pressure reducer and common rail. The continuity and momentum equations can be expressed as follows (Streeter et al., 1998):

$$\frac{\partial\rho}{\partial t} + \rho\frac{\partial v}{\partial x} = 0, \qquad (3)$$

$$\frac{\partial u}{\partial t} + \frac{\alpha^2}{\rho}\frac{\rho}{x} + \frac{f}{2d}u|u| = 0, \qquad (4)$$

where α is the sound speed in the gas, d is the pipe diameter, f is the D'Arcy friction coefficient depending on the Reynolds number. AMESim numerically solves the above equations by discretization.

For short pipes, the *Compressibility + friction submodel of pneumatic pipe* from is used, allowing to compute the flow according the following equation:

$$q = \sqrt{\frac{2D\Delta p}{L\rho f}}, \qquad (5)$$

where Δp is the pressure drop along the pipe of length L. The pipes connecting common rail and injectors are modelled in such a way.

Heat transfer exchanges are accounted for by the above mentioned AMESim components, provided that a heat transfer coefficient is properly specified. For a cylindrical pipe of length L consisting of a homogeneous material with constant thermal conductivity k and having an inner and outer convective fluid flow, the thermal flow Q is given by (Zucrow and Hoffman, 1976):

$$Q = \frac{2\pi k L \Delta T}{\ln \frac{r_o}{r_i}}, \qquad (6)$$

where ΔT is the temperature gradient between the internal and external surfaces, and r_o and r_i are the external and internal radiuses, respectively. With reference to the outside surface of the pipe, the heat-transfer coefficient U is:

$$U_o = \frac{k}{r_o \ln \frac{r_o}{r_i}}. \qquad (7)$$

4.3 Solenoid Valve and Magnetic Circuits

The AMESim model for the solenoid valve is composed of the following elements: a solenoid with preloaded spring, two moving masses with end stops subject to viscous friction and representing the magnet anchor and the shutter respectively, and a component representing the elastic contact between the anchor and the shutter. The intake section depends on the axial displacement of the shutter over the conical seat and is computed within the *Pneumatic ball poppet with conical seat* component, which also evaluates the drags acting on the shutter. The solenoid valve is driven by a peak-hold modulated voltage. The resulting current consists of a peak phase followed by a variable duration hold phase. The valve opening time is regulated by varying the ratio between the hold phase duration and signal period, namely the control signal *duty cycle*. This signal is reconstructed by using a *Data from ASCII file signal source* that drives a *Pulse Width Modulation* component.

To compute the magnetic force applied to the anchor, a supercomponent (*Solenoid with preloaded spring* in Fig. 2) modelling the magnetic circuit has been suitably built, as described in the following. The magnetic flux within the whole magnetic circuit is given by the Faraday law:

$$\dot{\varphi} = (e_{ev} - R_{ev} i_{ev})/n, \qquad (8)$$

where φ is the magnetic flux, R the n turns winding resistance, e_{ev} the applied voltage and i_{ev} the circuit current. Flux leakage and eddy-currents have been neglected. The magnetomotive-force MMF able to produce the magnetic flux has to compensate the magnetic tension drop along the magnetic and the air gap paths. Even though most of the circuit reluctance is applied to the air gap, nonlinear properties of the magnet, due to saturation and hysteresis, sensibly affect the system behaviour. The following equation holds:

$$MMF = MMF_s + MMF_a = H_s l_s + H_a l_a, \qquad (9)$$

where H is the magnetic field strength and l is the magnetic path length, within the magnet and the gap respectively. The air gap length depends on the actual position of the anchor. The magnetic induction within the magnet is a nonlinear function of H. It is assumed that the magnetic flux cross section is constant along the circuit, yielding:

$$B = \varphi / A_m = f(H_s) = \mu_0 H_a, \qquad (10)$$

where A_m is the air gap cross section and μ_0 is the magnetic permeability of air. The $B - H$ curve is the hysteresis curve of the magnetic material. Arranging the previous equations yields to φ, B and H. The resulting magnetic force and circuit current are:

$$F_{ev} = A_m B^2 / \mu_0, \qquad (11)$$

$$i_{ev} = MMF/n. \qquad (12)$$

The force computed by the previous equation is applied to the mass component representing the anchor, so that the force balance can be properly handled by AMESim.

4.4 Injectors

The injectors are solenoid valves driven by the ECU in dependence of engine speed and load. The whole injection cycle takes place in a 720° interval with a delay between each injection of 180°. A supercomponent including the same elements as for the solenoid valve has been built to model the electro-injectors. The command signal generation is demanded to the ECU component, which provides a square signal driving each injector and depending on the current engine speed, injector timings and pulse phase angle.

5 CONTROLLER DESIGN

In designing an effective control strategy for the injection pressure it is necessary to satisfy physical and technical constraints. In this framework, model predictive control (MPC) techniques are a valuable choice, as they have shown good robustness in presence of large parametric variations and model uncertainties in industrial processes applications. They predict the output from a process model and then impress a control action able to drive the system to a reference trajectory (Rossiter, 2003). A 2$^{\text{nd}}$ order state space analytical model of the plant (Lino et al., 2006) is used to derive a predictive control law for the injection pressure regulation. The model trades off between accuracy in representing the dynamical behaviour of the most significant variables and the need of reducing the computational effort and complexity of controller structure and development. The design steps are summarized in the following. Firstly, the model is linearized at different equilibrium points, in dependence of the working conditions set by the driver power request, speed and load. From the linearized models it is possible to derive a discrete transfer function representation by using a backward difference method. Finally, a discrete Generalised Predictive Contrl (GPC) law suitable for the implementation in the ECU is derived from the discrete linear models equations.

By considering the *duty cycle* of the signal driving the solenoid valve and the rail pressure as the input u and output y respectively, a family of ARX models can be obtained, according the above mentioned design steps (Lino et al., 2006):

$$\left(1-a_1q^{-1}\right)y(t)=\left(b_0q^{-1}-b_1q^{-2}\right)u(t), \qquad (13)$$

where q^{-1} is the shift operator and a_1, b_0, b_1 are constant parameters. The j-step optimal predictor of a system described by eq. 13 is (Rossiter, 2003):

$$\hat{y}(t+j|t)=G_j\Delta u(t+j-1)+F_jy(t), \qquad (14)$$

where G_j and F_j are polynomials in q^{-1}, and Δ is the discrete derivative operator. Let $\mathbf{r}$ be the vector of elements $y(t+j),\ j=1,\ldots,N$, depending on known values at time t. Then eq. (14) can be expressed in the matrix form $\hat{\mathbf{y}}=\mathbf{G}\tilde{\mathbf{u}}+\mathbf{r}$, being $\tilde{\mathbf{u}}=[\Delta u(t),\ldots,\Delta u(t+N-1)]^{\mathrm{T}}$, and $\mathbf{G}$ a lower triangular $N\times N$ matrix (Rossiter, 2003).

If the vector $\mathbf{w}$ is the sequence of future reference-values, a cost function taking into account the future errors can be introduced:

$$J=E\left\{(\mathbf{G}\tilde{\mathbf{u}}+\mathbf{r}-\mathbf{w})^{\mathrm{T}}(\mathbf{G}\tilde{\mathbf{u}}+\mathbf{r}-\mathbf{w})+\lambda\tilde{\mathbf{u}}^{\mathrm{T}}\tilde{\mathbf{u}}\right\}, \qquad (15)$$

where λ is a sequence of weights on future control actions. The minimization of J with respect of $\tilde{\mathbf{u}}$ gives the optimal control law for the prediction horizon N:

$$\tilde{\mathbf{u}}=\left(\mathbf{G}^{\mathrm{T}}\mathbf{G}+\lambda\mathbf{I}\right)^{-1}\mathbf{G}^{\mathrm{T}}(\mathbf{w}-\mathbf{r}). \qquad (16)$$

At each step, the first computed control action is applied and then the optimization process is repeated after updating all vectors. It can be shown (Lino et al., 2006) that the resulting control law for the case study becomes:

$$\Delta u(t)=k_1w(t)+\left(k_2+k_3q^{-1}\right)y(t)+k_4\Delta u(t-1), \qquad (17)$$

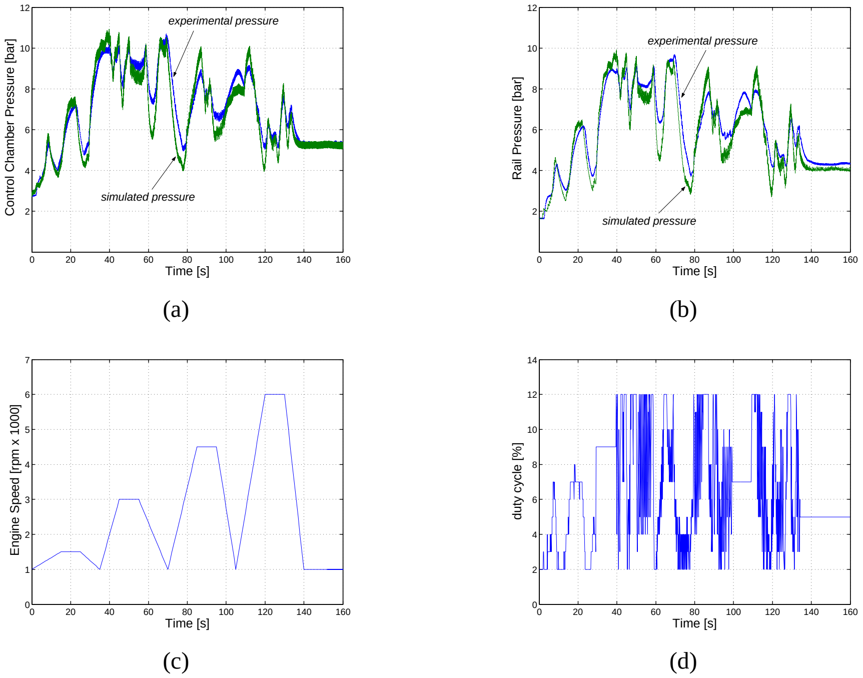
where $[k_1,k_2,k_3,k_4]$ depends on N.

6 SIMULATION AND EXPERIMENTAL RESULTS

To assess the effectiveness of the AMESim model in predicting the system behaviour, a comparison of simulation and experimental results has been performed. Since, for safety reasons, air is used as test fluid, the experimental setup includes a compressor, providing air at a constant input pressure and substituting the fuel tank. The injection system is equipped with four injectors sending the air to a discharging manifold. Moreover, a PC system with a National Instrument acquisition board is used to generate the engine speed and load signals, and a programmable MF3 development master box takes the role of ECU driving the injectors and the control valve.

Figure 3 refers to a typical transient operating condition, and compares experimental and simulation results. With a constant 40 bar input pressure, the system behaviour for a constant $t_j=3$ms injectors opening time interval, while varying engine speed and solenoid valve driving signal has been evaluated. The engine speed is composed of ramp profiles (Fig. 3(c)), while the *duty cycle* changes abruptly within the interval $[2\%, 12\%]$ (Fig. 3(d)). Figures 3(a) and 3(b) show that the resulting dynamics is in accordance with the expected behaviour. A maximum error of 10% confirms the model validity.

After the validation process, the AMESim virtual prototype was used to evaluate the GPC controller performances in simulation by employing the AMESim-Simulink interface, which enabled us to export AMESim models within the Simulink environment. The interaction between the two environments operates in a *Normal* mode or a *Co-simulation* mode. As for the former, a compiled S-function containing the AMESim model is generated and included in the Simulink block scheme, and then integrated by the



(a)

(b)

(c)

(d)

Figure 3: AMESim simulation and experimental results when varying *duty cycle.* and engine speed, with a constant $t_j = 3$ms; (a) control chamber pressure; (b) common rail pressure; (c) engine speed; (d) control signal *duty cycle*.

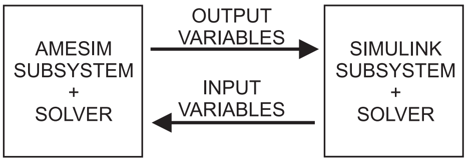


Figure 4: AMESim-MATLAB co-simulation interface.

Simulink solver. As for the latter, which is the case considered in this paper, AMESim and Simulink cooperate by integrating the relevant portions of models, as shown in figure 4.

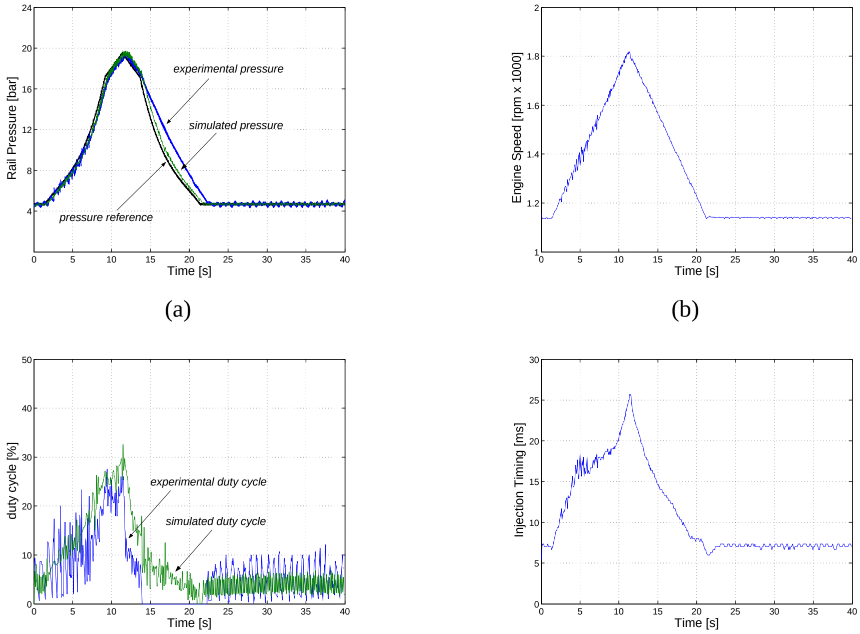
The GPC controller was tuned referring to models linearized at the starting equilibrium point, according to design steps of Section 5. The test considered ramp variations of the engine speed and load, for the system controlled by a GPC with a $N = 5$ $(0.5s)$ prediction horizon. The input air pressure from the compressor was always 30bar. The rail pressure reference was read from a static map depending on the working condition and had a sort of ramp profile as well. The final design step consisted in the application of the GPC control law to the real system.

In Fig. 5, the engine speed accelerates from 1100rpm to 1800rpm and then decelerates to 1100rpm, within a 20s time interval (Fig. 5(b)).

The control action applied to the real system guarantees a good reference tracking, provided that its slope does not exceed a certain value (Fig. 5(a), time intervals [0, 14] and [22, 40]). Starting from time 14s, the request of a quick pressure reduction causes the control action to close the valve completely (Fig. 5(c)) by imposing a *duty cycle* equal to 0. Thanks to injections, the rail pressure (Fig. 5(a)) decreases to the final 5bar reference value, with a time constant depending on the system geometry; the maximum error amplitude cannot be reduced due to the actuation variable saturation. Fig. 5(d) shows the injectors' exciting time during the experiment. It is worth to note that simulation and experimental results are in good accordance, supporting the proposed approach.

7 CONCLUSIONS

In this paper, we presented a procedure for integrating different models and tools for a reliable design, optimization and analysis of a mechatronic system as a whole, encompassing the real process and the control system. The effectiveness of the methodology has been illustrated by introducing a practical case study involving the CNG injection system for internal com-



(a)

(b)

(c)

(d)

Figure 5: AMESim model and real system responses for speed and load ramp variations and a 30bar input pressure, when controlled by a GPC with N = 5; (a) common rail pressure; (b) engine speed (c) *duty cycle*; (d) injectors exciting time interval.

bustion engines. The design process was organized in few steps: analysis of different candidate configurations carried out with the help of virtual prototypes developed in the AMESim environment; design and performance evaluation of controllers designed on a simpler model of the plant employing the virtual prototypes; validation of the control law on the real system. The implementation of the control law on the real system represented a benchmark for the evaluation of performances of the approach. Experimental results gave a feedback of benefits of the integration of the mechanical and control subsystems design, and proved the validity of the methodology.

REFERENCES

Ferretti, G., Magnani, G., and Rocco, P. (2004). Virtual Prototyping of Mechatronic Systems. *Annual Reviews in Control*, (28):193206.

Lino, P., Maione, B., Amorese, C., and DeMatthaeis, S. (2006). Modeling and Predictive Control of a New Injection System for Compressed Natural Gas Engines. In *IEEE CCA 2006 International Conference*, Munich, Germany.

Maione, B., Lino, P., DeMatthaeis, S., Amorese, C., Manodoro, D., and Ricco, R. (2004). Modeling and Control of a Compressed Natural Gas Injection System. *WSEAS Transactions on Systems*, 3(5):2164–2169.

Paynter, H. (1960). *Analysis and Design of Engineering Systems*. M.I.T. Press, Cambridge, MA.

Rossiter, J. (2003). *Model-Based Predictive Control: a Practical Approach*. CRC Press, New York.

Stobart, R., A.May, Challen, B., and Morel, T. (1999). New Tools for Engine Control System Development. *Annual Reviews in Control*, (23):109–116.

Streeter, V., Wylie, K., and Bedford, E. (1998). *Fluid Mechanics*. McGraw-Hill, New York, 9th edition.

IMAGINE S.A. (2004). *AMESim User Manual v4.2*. Roanne, France.

van Amerongen, J. and Breedveld, P. (2003). Modelling of Physical Systems for the Design and Control of Mechatronic Systems. *Annual Reviews in Control*, (27):87–117.

Zucrow, M. and Hoffman, J. (1976). *Gas Dynamics*. John Wiley & Sons, New York.

ON THE FORCE/POSTURE CONTROL OF A CONSTRAINED SUBMARINE ROBOT

Ernesto Olguín-Díaz and Vicente Parra-Vega
Robotics and Advanced Manufacturing, CINVESTAV-IPN, Saltillo, México
(ernesto.olguin,vicente.parra)@cinvestav.edu.mx

Keywords: Underwater vehicles, Force/position control, Sliding Modes Control.

Abstract: An advanced, yet simple, controller for submarine robots is proposed to achieve simultaneously tracking of time-varying contact force and posture, without any knowledge of its dynamic parameters. Structural properties of the submarine robot dynamic are used to design passivity-based controllers. A representative simulation study provides additional insight into the closed-loop dynamic properties for regulation and tracking cases. A succinct discussion about inherent properties of the open-loop and closed-loop systems finishes this work.

1 INTRODUCTION

In the last decade, we have witness a surprising leap on scientific knowledge and technological achievements for submarine robots, from a simple torpedo to modern UAV. Those vehicles are at the same time tantamount scientific and technological challenges to robotics, control and mechatronic researchers. First efforts for free motion submarine robots were focused for motion control, with sensing capabilities, similar to the case of fix-base robots in our labs. Of course, stable contact for submarine robots is a more complex problem in comparison to the typical force control problem of robot manipulators fixed to ground because not only due complementary complex dynamics such as buoyancy and added masses, but to de fact that the vehicle reference frame is not longer inertial.

However, some tasks requires more challenging goal of establishing stable contact, which leads us to study the problem of simultaneous force and pose (position and orientation) control of submarine robots. When the submarine robot achieves contact, like pushing itself against a wall or polishing a sunken surface vessel or manipulating tools on submarine pipe lines, forces are presented, and little is known about these contact forces, let alone exploit them either for design or control. The main general reason that this problem remains rather open is that we really know so little about, on one hand, in how to model and control properly a fully immersed vehicle constrained by rigid object; and on the other hand, the submarine force control technology lies behind system requirements, such as very fast sampling rates, even when the bandwidth of the submarine robot is very low.

Despite brilliant control schemes for free motion submarine robots in the past few years, in particular those of (Yoerger, 1985; Smallwood, 2001; Smallwood, 2004) those schemes does not guarantee formally convergence of tracking errors, let alone simultaneous convergence of posture and contact force tracking errors, there are several results that suggest empirically that a simple PD control structure behaves as stiffness control for submarine robots. So far the simultaneous convergence of time-varying contact forces and posture remains an open problem for the PD-like controller (notice that for submarine robots the role of model-free controllers are very important because it is very hard to known exactly the dynamic model and its dynamic parameters).

Recently, some efforts have focused on how to obtain simple control structures to control the time-varying pose of the UAV under the assumption that the relative velocities are low. For force control, virtually none complete control system has been published. However, to move forward more complex force controllers, we believe that more understanding of the structural properties of submarine robots in stable contact to rigid objects are required.

To understand more about this problem, we show that the open-loop rigid dynamics of submarine robots subject to holonomic constraints (rigid contact) exhibits similar properties of fixed-base robots. Then almost any force/position control scheme with the proper adequation can be used. In this paper we have chosen the orthogonalization principle (Parra-Vega and Arimoto, 1996) to propose a simple controller with advanced tracking stability properties.

1.1 Contribution

Despite the complexity of the problem, in this paper, the performance of a robot submarine carrying out a very simple contact tasks under a novel orthogonalization-based model-free controller is analyzed and discussed, with emphasize on structural properties. The closed-loop shows robust tracking of time-varying contact force and posture, without any knowledge of submarine dynamics. The proposed controller stands itself as a new controller which guarantees simultaneous tracking of submarine robot with a model-free decentralized control structure.

1.2 Organization

Section 2 introduces the nonlinear differential algebraic equations of the robot submarine dynamics in error coordinates. Section 3 presents the control design. Simulation data is presented for a Testbed robot of IFREMER in 4, and finally section 5 discusses some structural properties useful for further understand robot submarine in contact tasks. Conclusions are stated in Section 6.

2 THE MODEL

2.1 A Free Floating Submarine

The model of a submarine can be obtained via Kirchhoff formulation (Fossen, 1994), the inclusion of hydrodynamic effects such as added mass, friction and buoyancy and the account of external forces/torques like contact effects (Olguín-Díaz, 1999). The model is then expressed by the next set of equations:

$$M_v\dot{\nu}+C_v(\nu)\nu+D_v(\cdot)\nu+g_v(q) \;=\; u+F_c^{(v)} \qquad (1)$$

$$\nu \;=\; J_\nu(q)\dot{q} \qquad (2)$$

From this set, (1) is called the dynamic equation while (2) is called the kinematic equation. Both had been used long widely to express the dynamical behavior of free moving object in the 3D space.

The generalized coordinates vector $q \in \mathbb{R}^6$ is given on one hand by the 3 Cartesian positions x,y,z of the origin of the submarine frame (Σ_v) with respect to a inertial frame (Σ_0) fixed in some place on the surface of the Earth; and on the other hand by any set of attitude parameters that represent the rotation of the vehicle's frame with respect to the inertial one. Most common sets of attitude representation such a Euler angles, in particular roll-pitch-yaw (ϕ,θ,ψ), use only 3 variables (which is the minimal number of orientation variables). Then, for a submarine vehicle, the generalized coordinates $q=(x_v,y_v,z_v,\phi_v,\theta_v,\psi_v)$ represents its 6 degrees of freedom.

The vehicle's velocity $\nu \in \mathbb{R}^6$ is the vector representing both linear and angular velocity of the submarine expressed in its own vehicle's frame. This vector is then defined as $\nu=(\mathbf{v}_v^{(v)^T},\omega_v^{(v)^T})^T$. The relationship between this vector and the generalized coordinates is given by the kinematic equation. The linear operator $J_\nu(q) \in \mathbb{R}^{6\times 6}$ in (2) is built by the concatenation of two transformations. The first is defined as:

$$J_q(q)=\begin{bmatrix} I & 0 \\ 0 & J_\theta(\phi_v,\theta_v,\psi_v)\end{bmatrix} \in \mathbb{R}^{6\times 6}$$

Where $J_\theta \in \mathbb{R}^{3\times 3}$ is an operator that converts time derivatives of attitude (orientation) parameters in angular velocity: $\omega_v^{(0)}=J_\theta(\theta_v)\dot{\theta}_v$, with $\theta_v=(\phi_v,\theta_v,\psi_v)^T$ defined as the orientation parameter vector. The operator $J_q(q)$ maps the first time derivative of the generalized coordinates to linear and angular velocities of the submarine vehicle, both expressed in the inertial frame Σ_0.

The second operator transforms a 6 dimension tensor from the inertial frame to vehicle's frame by $J_R(q)=diag\{R_0^v,R_0^v\} \in \mathbb{R}^{6\times 6}$. Thus the *Jacobian* of the vehicle used in the kinematic equation is a linear operator defined as:

$$J_\nu(q)=J_R^T(q)J_q(q)$$

In the dynamic equation (1), matrices $M_v,C_v(\nu),D_v(\cdot) \in \mathbb{R}^{6\times 6}$ are respectively Inertia matrix, Coriolis matrix and Damping matrix. M_v includes the terms of classical inertia plus the hydrodynamic coefficients of the added mass effect (due to the amount of extra energy needed to displace the surrounding water when the submarine is moving). This Inertia matrix is constant, definite positive and symmetric only when the submarine is complete immersed and the relative water incidence velocity is small (Fossen, 1994). This condition is met for most common activities of a ROV, for example during sampling recollection in not shallow waters. The Coriolis vector $C_v(\nu)\nu$ represents the Coriolis forces, gyroscopic terms of the classical inertia effects plus the quadratic velocity terms induced by the added mass. The Coriolis matrix in this representation does not depend on the position but only on the velocity, in contrast to the same expression for a Robot Manipulator (Spong M.W., 1989). However it fulfills the classic relationship for Lagrangian systems: $\dot{M}_v - 2C_v(\nu) = Q; Q + Q^T = 0$, and is indeed skew symmetric. The Damping matrix represents all the hydrodynamic effects of energy dissipation. For that reason it is a strictly positive definite matrix, $D_v(q,\nu,t) > 0$ (Fossen, 1994). Its arguments are commonly the vehicle's orientation ϕ_v, θ_v, ψ_v, the generalized velocity ν, and the velocity of the surrounding water. The diagonal components represents the drag forces while the off-diagonal components represent the lift forces.

Finally, vectors $g_v(q), u, F_c^{(\nu)} \in \mathrm{IR}^6$ are all force wrenches (force-torque vector) in the vehicle's frame. They represent respectively: gravity, input control and the contact force. Gravity vector includes buoyancy effects and it does not depend on velocity but on the orientation (attitude) of the submarine with respect to the inertial frame. The contact force wrench is the one applied by the environment *to* the submarine. The input control are the forces/torques induced by the submarine thrusters in the vehicle frame.

The dynamic model (1)-(2) can be rearranged by replacing (2) and its time derivative into (1). The result is one single equation model:

$$M_q(q)\ddot{q} + C_q(q,\dot{q})\dot{q} + D_q(\cdot)\dot{q} + g_q(q) = u_q + \tau_c \quad (3)$$

which has the form of any Lagrangian system. Its components fulfills all properties of such systems i.e. definite positiveness of inertia and damping matrices, skew symmetry of Coriolis matrix and appropriate bound of all components (Sagatun, 1991).

The control input in this equation is obtained by a linear transformation of the real input using the linear operator given by the kinematic equation:

$$u_q = J_\nu^T(q)u. \quad (4)$$

This last equation becomes important because it relates the nominal control signal u_q to be designed in the general coordinates space with the real physical input u that shall be implemented in the real vessel. The contact effect is also obtained by the same transformation. However it can be expressed directly from the contact wrench in the inertial frame (Σ_0) by the next relationship

$$\tau_c = J_\nu^T(q)F_c^{(\nu)} = J_q^T(q)F_c^{(0)} \quad (5)$$

where the contact force $F_c^{(0)}$ is the one expressed in the inertial frame. By simplicity it will be noted as F_c from this point further. The relationship with the one expressed in the vehicle's frame is given by $F_c = J_R^T(q)F_c^{(\nu)}$. This wrench represents the contact forces/torques exerted by the environment to the submarine as if measured in a non moving (inrtial) frame. These forces/torques are given by the normal force of an holonomic constraint when in contact and the friction due to the same contact. For simplicity in this work, tangential friction is not considered.

2.2 Contact Force Due to a Holonomic Constraint

A holonomic constraint can be expressed as a function of the generalized coordinates of the submarine as

$$\varphi(q) = 0 \quad (6)$$

with $\varphi(q) \in \mathrm{IR}^r$. The number r stand for the number of contact points between the submarine and the motionless rigid object. Stable contact appears only when the robot submarine does not deattach from the object modeled by $\varphi(q) = 0$. In this case, the time derivatives of this constraint are zero

$$\frac{d^n\varphi(q)}{dt^n} = 0; \ \forall\, n \geq 1 \quad (7)$$

For $n = 1$ the last constraint becomes:

$$\dot{\varphi}(q) = J_\varphi(q)\dot{q} = 0 \quad (8)$$

where $J_\varphi(q) = \frac{\partial\varphi(q)}{\partial q} \in \mathrm{IR}^{r\times n}$ is the constraint jacobian. Last equation means that velocities of the submarine in the directions of constraint jacobian are restricted to be zero. This directions are then normal to the constraint surface $\varphi(q)$ at the contact point. As a consequence, the normal component of the contact force has exactly the same direction as those defined by $J_\varphi(q)$. Then the contact force wrench can be expressed as

$$F_c = J_{\varphi+}^T(q)\lambda \quad (9)$$

where $J_{\varphi+}(q) \stackrel{\triangle}{=} \frac{J_\varphi}{\|J_\varphi\|}$ is a normalized version of the constraint jacobian; and λ is the magnitude of the normal contact force at the origin of vehicle frame: $\lambda = \|F_c\|$.

The free moving model expressed by (1)-(2), when in contact with the holonomic constraint can be rewritten as:

$$\begin{aligned} M_v\dot{\nu} + h_v(q,\nu,t) &= u + J_R^T(q)J_{\varphi+}^T(q)\lambda && (10)\\ \nu &= J_\nu(q)\dot{q} && (11)\\ \varphi(q) &= 0 && (12) \end{aligned}$$

where $h_v(q,\nu,t) = C_v(\nu)\nu + D_v(q,\nu,t)\nu + g_v(q)$.

Equivalently, the model (3) is also expressed as

$$
\begin{aligned}
M_q(q)\ddot{q} + h_q(q,\dot{q},t) &= u_q + J_{\bar{\varphi}}^T(q)\lambda \qquad (13)\\
\varphi(q) &= 0 \qquad (14)
\end{aligned}
$$

with $h_q(q,\dot{q},t) = C_q(q,\dot{q})\dot{q} + D_q(q,\dot{q},t)\dot{q} + g_q(q)$ and $J_{\bar{\varphi}}(q) = J_{\varphi+}(q)J_q(q)$. Equations (13)-(14) are a set of Differential Algebraic Equations index 2 (DAE-2). To solve them numerically, a DAE solver is required. This last representation has the same structure and properties as those reported in (Parra-Vega, 2001).

2.3 Numerical Considerations

To compute the value of λ, the constrained Lagrangian that fulfils the constrained movement, can be calculated using the second derivative of (7): $\ddot{\varphi}(q) = 0$. Then using the dynamic equation (either of them) and after some algebra its expression becomes either:

$$
\begin{aligned}
\lambda &= \left[J_\varphi J_\nu^{-1} M_v^{-1} J_R^T J_{\varphi+}^T\right]^{-1} \Big(J_\varphi J_\nu^{-1} M_v^{-1}\left(h_v - u\right)\\
&\qquad - \frac{d}{dt}\left(J_\varphi J_\nu^{-1}\right)\nu\Big) \qquad (15)\\
&= \left[J_\varphi M_q^{-1} J_{\bar{\varphi}}^T\right]^{-1} \Big(J_\varphi M_q^{-1}\left(h_q - u_q\right) - \dot{J}_\varphi \dot{q}\Big) \qquad (16)
\end{aligned}
$$

The set of equations (10)-(11)-(15) or the set (13)-(16) describes the constrained motion of the submarine when in contact to infinitely rigid surface described by (6). Numerical solutions of these sets can be obtained by simulation, however the numerical solution, using a DAE solver, can take too much effort to converge due to the fact that these sets of equation represent a highly stiff system. In order to minimize this numerical drawback, the holonomic constraint has been treated as a numerically compliant surface which dynamic is represented by

$$
\ddot{\varphi}(q) + D\dot{\varphi}(q) + P\varphi(q) = 0 \qquad (17)
$$

This is known in the force control literature of robot manipulators as constrained stabilization method, which bounds the nonlinear numerical error of integration of the backward integration differentiation formula. With a appropriate choice of parameters $P \gg 1$ and $D \gg 1$, the solution of $\varphi(q,t) \rightarrow 0$ is bounded. This dynamic is chosen to be fast enough to allow the numerical method to work properly. In this way, it is allowed very small deviation on the computation of λ, typically in the order of -10^6 or less, which may produce, according to some experimental comparison, less than 0.001% numerical error. Then, the value of the normal contact force magnitude becomes:

$$
\begin{aligned}
\lambda &= \left[J_\varphi J_\nu^{-1} M_v^{-1} J_R^T J_{\varphi+}^T\right]^{-1} \Big(J_\varphi J_\nu^{-1} M_v^{-1}\left(h_v - u\right)\\
&\qquad - \frac{d}{dt}\left(J_\varphi J_\nu^{-1}\right)\nu - DJ_\varphi J_\nu^{-1}\nu - P\varphi(q)\Big) \qquad (18)\\
&= \left[J_\varphi M_q^{-1} J_{\bar{\varphi}}^T\right]^{-1} \Big(J_\varphi M_q^{-1}\left(h_q - u_q\right) - \dot{J}_\varphi \dot{q}\\
&\qquad - DJ_\varphi \dot{q} - P\varphi(q)\Big) \qquad (19)
\end{aligned}
$$

It can be pointed out that equation (19) compute the contact force magnitude using the generalized coordinates variables only. From this equation, desired force can be easily obtained. Also note that the preliminary inverse operator is only expressed by the inertia matrix inverse and the two constraint jacobians. The first is the gradient of the contact surface, while the second is the normalized gradient with a simple kinematic variation given by the operator J_q.

3 CONTROL DESIGN

The introduction of a so called Orthogonalization Principle has been a key in solving, in a wide sense, the force control problem of a robot manipulators. This physical-based principle states that the projection (dot product) of contact forces and joint generalized velocities is zero. Then passivity arises from torque input to generalized velocities. To preserve passivity, then, the closed-loop system must satisfy the passivity inequality for controlled generalized error velocities. This is true for robot manipulators with fixed frame, and here we prove that this also happens with robot whose reference frame is not inertial (fixed).

3.1 Orthogonalization Principle and Linear Parametrization

Similar to (Liu et al., 1997), an orthogonal projection of $J_\varphi(q)$ arises onto the tangent space at the contact point. This tangent space can be calculated by the definition of the following operator

$$
Q(q) \overset{\triangle}{=} I_n - J_\varphi^T(q)J_\varphi(q) \in \mathbb{R}^{n\times n} \qquad (20)
$$

where I_n is an identity matrix of proper size. Notice that $rank\{Q(q)\} = n - r$ since $rank\{J_\varphi(q)\} = r$. Also notice that $Q\dot{q} = \dot{q}$ due to the definition of Q and (8). Therefore, and according to the Orthogonalization Principle the integral of $(\tau, \dot{q})$ is upper bounded by $-\mathcal{H}(t_0)$, for $\mathcal{H}(t) = K + P$, and passivity arise for fully immersed submarines without inertial frame. Therefore, passivity-based control is a powerful option to control such systems.

On the other hand it is known that equation (13) can be linearly parameterized as follows

$$M_q(q)\ddot{q}+C_q(q,\dot{q})\dot{q}+D_q(\cdot)\dot{q}+g_q(q)=Y(q,\dot{q},\ddot{q})\Theta \tag{21}$$

where the regressor $Y(q,\dot{q},\ddot{q})\in \mathrm{I\!R}^{n\times p}$ is composed of known nonlinear functions and $\Theta\in\mathrm{I\!R}^p$ by p unknown but constant parameters.

3.2 Change of Coordinates

A variation Y_r of regressor Y can be designed such that

$$\begin{aligned} Y_r(q,\dot{q},\dot{q}_r,\ddot{q}_r)\Theta \;=& \\ M_q(q)\ddot{q}_r+\left[C_q(q,\dot{q})+D_q(\cdot)\right]\dot{q}_r+g_q(q)& \end{aligned} \tag{22}$$

where $(\dot{q}_r,\ddot{q}_r)$ are some time functions *yet to be defined.* Then the open loop equation (13) can be written by adding and subtracting (22) as

$$\begin{aligned} M_q(q)\dot{s} \;=\;& -\left[C_q(q,\dot{q})+D_q(\cdot)\right]s \\ & -Y_r(q,\dot{q},\dot{q}_r,\ddot{q}_r)\Theta+J_\varphi^T(q)\lambda+u_q \end{aligned} \tag{23}$$

where $s\overset{\triangle}{=}\dot{q}-\dot{q}_r$ is called the extended error. The problem of control design for the open loop equation (23) is to find u_q such that exponential convergence arises when $Y_r\Theta$ is not available.

3.3 Orthogonal Nominal Reference

Consider that $q_d(t)$ and $\lambda_d(t)$ are the desired trajectories functions for position and contact force. Consider as well, the tracking errors defined as $\tilde{q}\overset{\triangle}{=}q(t)-q_d(t)$ for position error while $\tilde{\lambda}\overset{\triangle}{=}\lambda(t)-\lambda_d(t)$ stand for the force contact error. Consider also the next set of definitions:

$$S_p \;\overset{\triangle}{=}\; \dot{\tilde{q}}+\sigma\tilde{q} \tag{24}$$

$$S_{dp} \;\overset{\triangle}{=}\; S_p(t_0)e^{-\alpha(t-t_0)} \tag{25}$$

$$S_{qp} \;\overset{\triangle}{=}\; S_p-S_{dp} \tag{26}$$

$$S_F \;\overset{\triangle}{=}\; \int_{t_0}^{t}\tilde{\lambda}dt \tag{27}$$

$$S_{dF} \;\overset{\triangle}{=}\; S_F(t_0)e^{-\eta(t-t0)} \tag{28}$$

$$S_{qF} \;\overset{\triangle}{=}\; S_F-S_{dF} \tag{29}$$

with $\alpha>0$, $\eta>0$. Finally, consider the following reference $\dot{q}_r$ function:

$$\begin{aligned} \dot{q}_r \;=\; & Q\left(\dot{q}_d-\sigma\tilde{q}+S_{dp}-\gamma_1\int_{t_0}^{t}sgn\{S_{qp}(t)\}dt\right) \\ & +\beta J_\varphi^T\left(S_F-S_{dF}+\gamma_2\int_{t_0}^{t}sgn\{S_{qF}(t)\}dt\right) \end{aligned} \tag{30}$$

where the parameters β, σ, γ_1 and γ_2 are constant matrices of appropriate dimensions; and $sgn(y)$ stands for the entrywise signum function of vector y,

Using (30) into s, it arises

$$s=Q(q)S_{vp}-\beta J_\varphi^T(q)S_{vF} \tag{31}$$

where the orthogonal extended position and force manifolds S_{vp} and S_{vF}, respectively, are given by

$$S_{vp} \;=\; S_{qp}+\gamma_1\int sgn(S_{qp}(\varsigma))d\varsigma \tag{32}$$

$$S_{vF} \;=\; S_{qF}+\gamma_2\int sgn(S_{qF}(\varsigma))d\varsigma \tag{33}$$

Notice that although the time derivative of $\dot{q}_r$ is discontinuous, it will not be used in the controller. In the next subsection we show that if $\dot{S}_{vp}$, $\dot{S}_{vF}$ are bounded, and for large enough γ_1,γ_2, then simultaneous tracking of contact force and posture is achieve.

3.4 Model-Free Second Order Sliding Mode Control

Consider the following nominal continuous control law:

$$\begin{aligned} u_q \;=\; & -K_d\left(Q(q)s_{vq}-\beta J_\varphi^T(q)s_{vF}\right) \\ & +J_\varphi^T(q)\left(-\lambda^d+S_F\right) \\ & +\gamma_2 J_\varphi^T(q)\left(\tanh(\mu S_{qF})+\eta\int \mathrm{sgn}(S_{qF})\right) \end{aligned} \tag{34}$$

with $\mu>0$ and $K_d=K_d^T>0,\in\mathrm{I\!R}^{n\times n}$. This nominal control, designed in the q-space can be mapped to the original coordinates model, expressed by the set (1)-(2), using the next relationship

$$u=J_\nu^{-T}(q)u_q \tag{35}$$

thus, the physical controller u is implemented in terms of a key inverse mapping J_ν^{-T}.

3.4.1 Closed-loop System

The open loop system (23) under the continuous model-free second order sliding mode control (34) yields to

$$\begin{aligned} M_q\dot{s} \;=\; & -\left[C_q+D_q+K_d\right]s-Y_r\Theta \\ & +J_\varphi^T\left(\tilde{\lambda}+\eta S_F\right)+\gamma_2 J_\varphi^T Z+\tau^* \\ & +\gamma_2 J_\varphi^T\left(sgn(S_{qF})+\eta\int sgn(S_{qF})\right) \end{aligned} \tag{36}$$

where $Z = tanh(\mu S_{qF}) - sgn(S_{qF})$ and $\tau^* \equiv 0$ is the virtual control input for the passivity analysis.

3.4.2 Stability Analysis

Theorem 1 *Consider a constrained submarine (13)-(16) under the continuous model-free second order sliding mode control (34). The Underwater system yields a second order sliding mode regime with local exponential convergence for the position, and force tracking errors.*

Proof. A passivity analysis $\langle S, \tau^* \rangle$ indicates that the following candidate Lyapunov function V qualifies as a Lyapunov function

$$V = \frac{1}{2}(s^T M_q s + \beta S_{vF}^T S_{vF}) \tag{37}$$

for a scalar $\beta > 0$. The time derivative of the Lyapunov candidate equation immediately leads to

$$\begin{aligned} \dot{V} &= -s^T(K_d + D_q)s - \beta\eta S_{vF}^T S_{vF} - s^T Y_r \Theta \\ &\quad + s^T \gamma_2 J_{\varphi}^T Z \qquad (38) \\ &\leq -s^T K_d s - \beta\eta S_{vF}^T S_{vF} + \|s\|\|Y_r\Theta\| \\ &\quad + \|s\|\|\gamma_2\|\|J_{\bar{\varphi}}\|\|Z\| \\ &\leq -s^T K_d s - \beta\eta S_{vF}^T S_{vF} + \|s\|\|\varepsilon\| \end{aligned}$$

where it has been used the skew symmetric property of $\dot{M} - 2C(q,\dot{q})$, the boundedness of both the feedback gains and submarine dynamic equation (there exists upper bounds for $M, C(q,\dot{q}), g(q), \dot{q}_r, \ddot{q}_r$), the smoothness of $\varphi(q)$ (so there exists upper bounds for J_φ and $Q(q)$), and finally the boundedness of Z. All these arguments establish the existence of the functional ε. Then, if K_d, β and η are large enough such that s converges into a neighborhood defined by ε centered in the equilibrium $s = 0$, namely

$$s \to \varepsilon \; as \; t \to \infty \tag{39}$$

This result stands for local stability of s provided that the state is near the desired trajectories for any initial condition. This boundedness in the $\mathcal{L}_\infty$ sense, leads to the existence of the constants $\varepsilon_3 > 0$ and $\varepsilon_4 > 0$ such that

$$|\dot{S}_{vp}| < \varepsilon_3, \tag{40}$$

$$|\dot{S}_{vF}| < \varepsilon_4 \tag{41}$$

An sketch of the local convergence of S_{vp} is as follows[1]. Locally, in the $n - r$ dimensional image of Q, we have that $S_{qp}^* = QS_{qp} \in \mathbb{R}^n$. Consider now that under an abuse of notation that $S_{qp} = S_{qp}^*$, such that for small initial conditions, if we multiply the derivative of S_{qp} in (32) by S_{qp}^T, we obtain

$$\begin{aligned} S_{qp}^T \dot{S}_{qp} &= -\gamma_1|S_{qp}| + S_{qp}^T \dot{S}_{vp} \qquad (42) \\ &\leq -\gamma_1|S_{qp}| + |S_{qp}||\dot{S}_{vp}| \end{aligned}$$

Substituting (40) into (42) yields

$$S_{qp}^T \dot{S}_{qp} \leq -(\gamma_1 - \varepsilon_3)|S_{qp}| \tag{43}$$

where γ_1 must be chosen such that $\gamma_1 > \varepsilon_3$. The equation (43) is precisely the condition for the existence of a sliding mode at $S_{qp}(t) = 0$. The sliding mode is established in a time $t \leq |S_{qp}(t_0)|/(\gamma_1 - \varepsilon_3)$, and according to the definition of S_{qp} (below (32)), $S_{qp}(t_0) = 0$, which simply means that $S_{qp}(t) = 0$ for all time. We see that if we multiply the derivative of (33) by S_{vf}^T, we obtain

$$\begin{aligned} S_{qF}^T \dot{S}_{qF} &= -\gamma_2|S_{qF}| + S_{qF}^T \dot{S}_{vF} \qquad (44) \\ &\leq -\gamma_2|S_{qF}| + |S_{qF}||\dot{S}_{vF}| \end{aligned}$$

substituting (41) into (44) yields

$$S_{qf}^T \dot{S}_{qf} \leq -(\gamma_2 - \varepsilon_4)|S_{qf}| \tag{45}$$

where γ_2 must be chosen such that $\gamma_2 > \varepsilon_4$. The equation (45) is precisely the condition for the existence of a sliding mode at $S_{qF}(t) = 0$. The sliding mode is established in a time $t \leq |S_{qF}(t_0)|/(\gamma_2 - \varepsilon_4)$ and, according to (33), $S_{qF}(t_0) = 0$, which simply means that $S_{qf}(t) = 0$ for all time, which simply implies that $\lambda \to \lambda_d$ exponentially fast.

4 SIMULATION RESULTS

Force control of submarine robots is in its infancy, therefore simulations study must be kept rather simple to enter into the intricacies of the numerical solutions. Later on, the complexity of the simulation study will be be increased such that underwater disturbances can include unmodelled submarine dynamics, such as lateral marine current with high Reynolds number, to mention one. Then, the simulation study has been made on a simplified platform of a real submarine. Data has been obtained from the Vortex system of IFREMER (IFREMER, 1992). Simulator presents only vertical planar results (only in the x-z plane), so the generalized coordinates for this case of study are:

$$q = \begin{pmatrix} x_v \\ z_v \\ \theta_v \end{pmatrix} \tag{46}$$

The vehicle velocities are

$$v = \begin{pmatrix} u_v \\ w_v \\ q_v \end{pmatrix} \tag{47}$$

[1] The strict analysis follows Liu, *et. al.*

where u_v and w_v are linear velocities (surge and heave) and q_v is the angular velocity in the x-z plane. The holonomic constraint is given by a vertical surface located at 2 meters from the origin. This is expressed as:

$$\varphi(q) \equiv x - 2 \tag{48}$$

Initial conditions were calculated to be at the contact surface with no force. Simulations with simple PD were also performed as a comparison tool. The model-free control parameters are as follows:

α	β	γ_1	γ_2
4	0.025	0.0025	10^{-3}
K_d	σ	η	μ
$200\hat{M}_v$	5	1000	1

where $\hat{M}_v$ is made by the diagonal values of the constant Inertia matrix when expressed in the vehicle's frame. For de PD the gains were defined as $K_p = 100\hat{M}_v$ and $K_d = 20\hat{M}_v$. The numerical dynamic induced in the constraint surface were performed with $P = 9x10^6$ and $D = 36x10^3$.

4.1 Set-Point Control Case

The task is very simple. It consist on a change of deep, from 1m to 1.5m, and orientation from 5 degrees to 0, while the submarine remains in contact with a constant contact force of 170N.

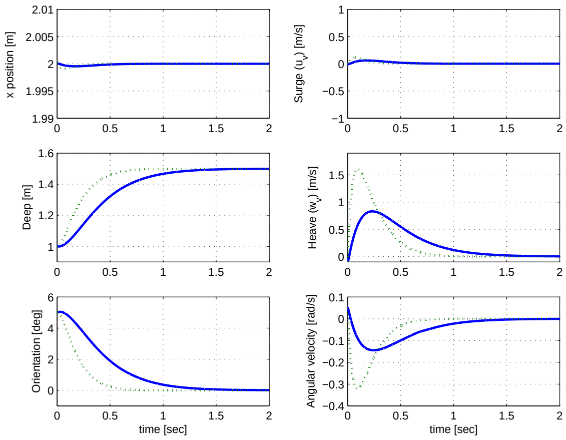


Figure 1: Generalized coordinates q and vehicle velocities ν, in vehicle's frame (continuous line for model-free second order sliding mode control, dotted line for PD control).

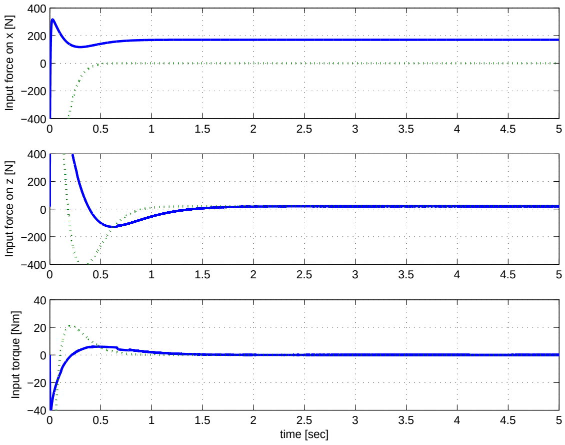


Figure 2: Inputs controlled forces u, in vehicle's frame (continuous line for model-free second order sliding mode control, dotted line for PD control).

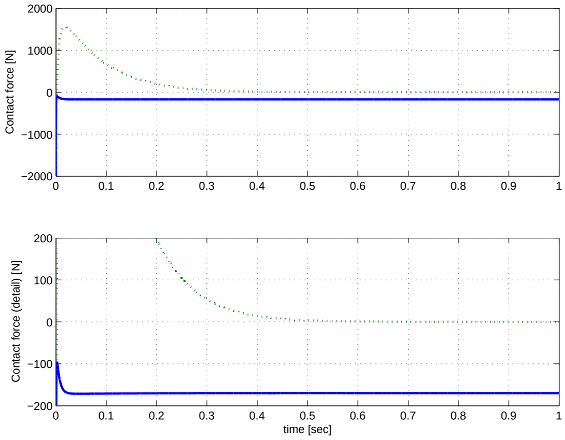


Figure 3: Contact force magnitude λ (continuous line for model-free second order sliding mode control, dotted line for PD control).

Figures 1 and 2 shows respectively position, velocity and force inputs. The difference in settling time has been explicitly computed in order to visualize the qualitative differences. In any case, this time interval can be modified with appropriate tuning of gain parameters. Notice that there is some transient and variability in the position of the contact point in the the direction normal to that force (the x coordinate). This is more visible in the velocity curve where velocity in the x direction should not occur if the constraint is infinitely rigid. The same transients are present in the force graphic where a noticeable difference between simple PD and model-free second order sliding mode control is present. The big difference is that although PD can regulate with some practical relative accuracy it is not capable to control force references, either constant nor variable. Figure 3 shows the transient of the contact force magnitude λ (only the 1st second is shown). The second window shows with more detail the transient in this contact force due to the numerical modification. In this graphic is is clear that no force is induced with the PD scheme. In the

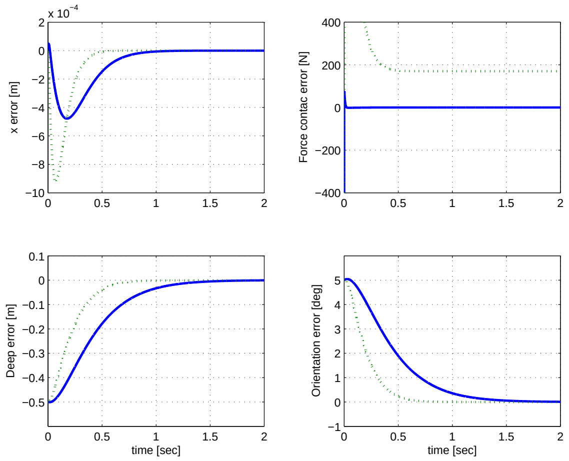


Figure 4: Set-point control errors: $\tilde{q} = q(t) - q^d$ and $\tilde{\lambda} = \lambda(t) - \lambda^d$ (continuous line for model-free second order sliding mode control, dotted line for PD control).

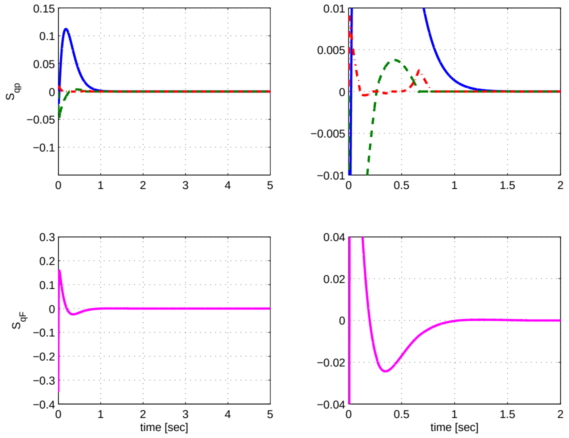


Figure 5: Control variables S_{qp} and S_{qF} for the model-free second order sliding mode control.

case of the model-free 2nd order sliding mode, force set-point control is achieved very fast indeed. Figure 4 shows the set-point control error for position (3 curves) and force. Is again evident the substantial difference in schemes for force control. Figure 5 shows the convergence of the extended position and force manifolds. They do converge to zero and induce there after the sliding mode dynamics.

4.2 Tracking Control Case

This task differs from the set-point control case in that the deep is a periodic desired trajectory center at 1 meter deep with 1 meter amplitude (pick to pick) and a 5 sec period. Contact desired force remains constant at 170N.

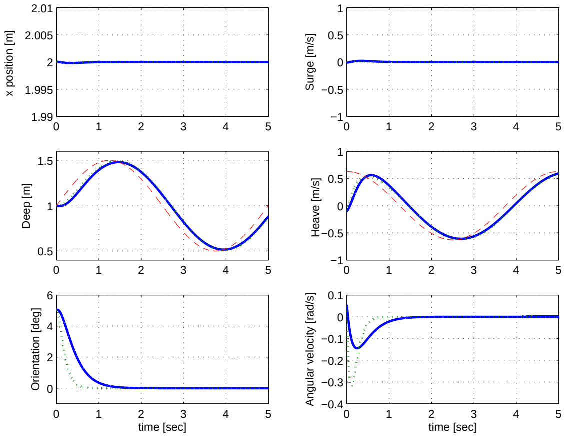


Figure 6: Generalized coordinates q and vehicle velocities ν, in vehicle's frame for the tracking case (continuous line for model-free second order sliding mode control, dotted line for PD control, dashed line for the tracking reference).

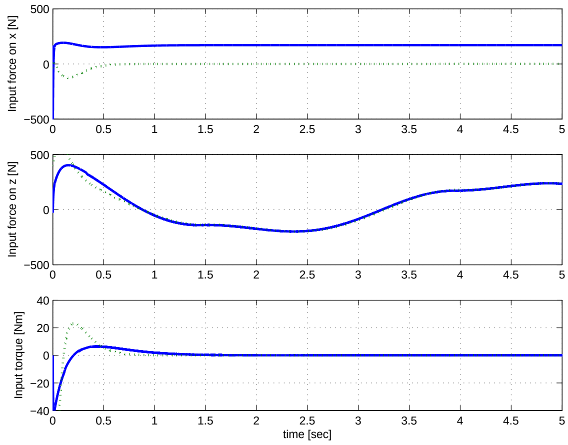


Figure 7: Inputs controlled forces u, in vehicle's frame (continuous line for model-free second order sliding mode control, dotted line for PD control).

As for the set-point control case, the position error and velocities in the constrained direction remain very small as can be see in figure 6. Also in this figure, the deep response has a lag from the tracking reference. As for the input control forces u the steady state resemble to those of the set-point control. For the deep coordinate, the control input tends to to be the same after some relative small time. Again, the PD does not provide any force in the constraint direction. The last can be verified in figure 8 where, for the PD case, there is no contact force although the position of the contact point is indeed in contact. For the model-free 2nd order sliding mode control, the contact force has the same shape as for the set-point control case. The same transient, due to the numerical consideration for

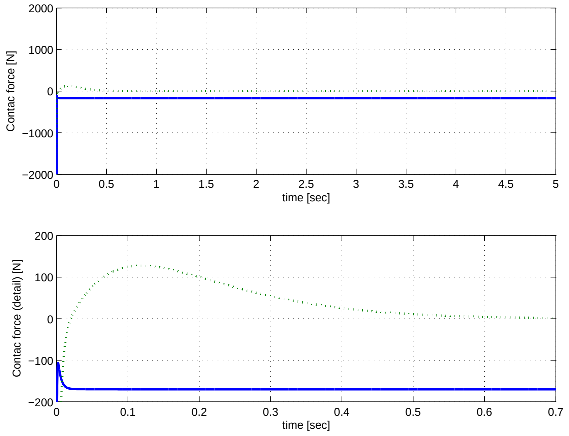


Figure 8: Contact force magnitude λ (continuous line for model-free second order sliding mode control, dotted line for PD control).

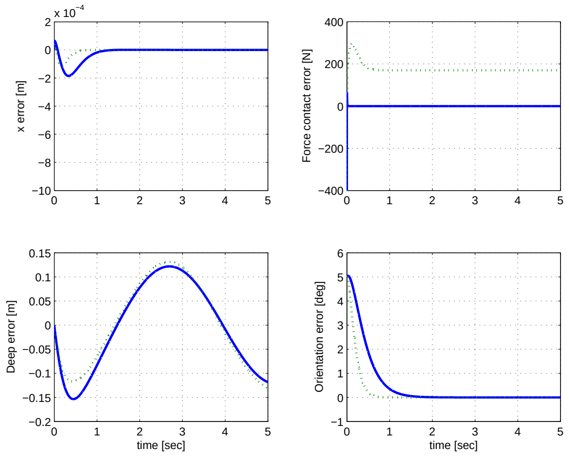


Figure 9: Set-point control errors: $\tilde{q} = q(t) - q^d$ and $\tilde{\lambda} = \lambda(t) - \lambda^d$ (continuous line for model-free second order sliding mode control, dotted line for PD control).

the almost rigid constraint surface, is present. The tracking errors for the x position and force has been reduced for the PD case certainly due to numerical initial conditions. In the tracking of deep, the model-free 2nd order sliding mode has apparently the same performance as the simpler PD controller. As it has been pointed out by other authors, for this kind of system, the PD controller remains as the best and simpler controller for position tracking. However, it does not track any force reference, which is the essence of the model-free 2nd order sliding mode control. The last figure 10 shows the convergence of the extended position and force manifolds to zero and induce there after the sliding mode dynamics.

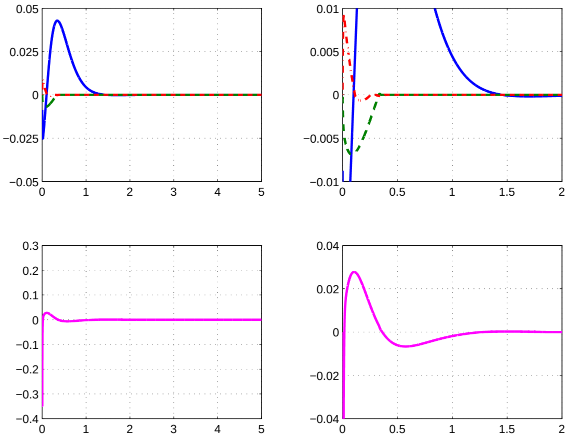


Figure 10: Control variables S_{qp} and S_{qF} for the model-free second order sliding mode control.

5 SOME DISCUSSIONS ON STRUCTURAL PROPERTIES

In the beginning of this study, we were expecting an involved control structure for submarine robots, in comparison to fixed-base robots, with additional control terms to compensate for hydrodynamic and buoyancy forces. However, surprisingly, no additional terms were required! Due to judicious treatment of the mathematical model, and careful mapping of velocities through jacobians in vehicle frame and generalized frame, it suffices only proper mapping of the gradient of contact forces. This results open the door to study similar force control structures developed for fixed-base robots, and so far, the constrained submarine robot control problem seems now at reach.

5.1 Properties of the Dynamics

As it as pointed out in Section 2, the model of a submarine robot can be expressed in either a *self space* (where the inertia matrix is constant for some conditions that in practice are not difficult to met) or in a generalized coordinate space (in which the inertial matrix is no longer constant but the model is expressed by only one equation, likewise the kinematic lagrangian chains). Both representations has all known properties of lagrangian systems such skew symmetry for the Coriolis/Inertia matrix, boundless of all the components, including buoyancy, and passivity preserved properties for the hydrodynamic added effects. The equivalences of these representation of the same model can be found by means of the kinematic equation. The last one is a linear operator that maps generalized coordinates time derivative with a generalized physical velocity. This relationship is specially important for the angular velocity of a free moving object due to the fact that time derivative of angular representations (such a roll-pitch-yaw) do not stand for the angular velocity. However, there is always a correspondence between these vectors. For external forces this mapping is indeed important. It relates a physical force/torque wrench to the generalized coordinates $q = (x_v, y_v, z_v, \phi_v, \theta_v, \psi_v)$ whose last 3 components does not represent a unique physical space. In this work, such mapping is given by J_q and appears in the contact force mapping by the normalized operator $J_{\bar{\varphi}}$. The operator J_q can be seen as the difference of a Analytical Jacobian and the Geometric Jacobian of a robot manipulator.

5.2 The Controller

Notice that the controller exhibits a PD structure plus a nonlinear I-tame control action, with nonlinear time-varying feedback gains for each orthogonal subspace. It is in fact a decentralized PID-like, coined as ”Sliding PD” controller. It is indeed surprising that similar control structures can be implemented seemingly for a robot in the surface or below water, with similar stability properties, simultaneous tracking of contact force and posture. Of course, this is possible under a proper mapping of jacobians, key to implement this controller.

5.3 Friction at the Contact Point

When friction at the contact point arises, which is expected for submarine tasks wherein the contact object is expected to exhibits a rough surface, with high friction coefficients, a tangential friction model should be added in the right hand side. Particular care must be employed to map the velocities. Complex friction compensators can be designed, similar to the case of force control for robot manipulators, therefore details are omitted.

6 CONCLUSIONS

Although a PD controller, recently validated by (Smallwood, 2004) has a much simpler structure and very good performance for underwater vehicles in position control schemes, it does not deal with the task of contact force control.

The desired contact force can be calculated via equation (19) directly in the generalized frame in which it occurs.

Structural properties of the open-loop dynamics of submarine robots are key to design passivity-based controllers. In this paper, an advanced, yet simple, controller is proposed to achieve simultaneously tracking of time-varying contact force and posture, without any knowledge of his dynamic parameters. This is a significant feature of this controller, since in particular for submarine robots the dynamics are very difficult to compute, let alone its parameters.

A simulation study provides additional insight into the closed-loop dynamic properties for set-point control and tracking case.

REFERENCES

Fossen, T. I. (1994). *Guidance and Control of Ocean Vehicles*. Chichester.

IFREMER (1992). Project vortex: Modélisation et simulation du comportement hidrodynamique d’un véhicule sous-marin asservi en position et vitesse. Technical report, IFREMER.

Liu, Y., Arimoto, S., Parra-Vega, V., and Kitagaki, K. (1997). Decentralized adaptive control of multiple manipulators in cooperations. *International Journal of Control*, 67(5):649–673.

Olguín-Díaz, E. (1999). *Modélisation et Commande d’un Système Véhicule/Manipulateur Sous-Marin*. PhD thesis, Laboratoire d’Automatique de Grenoble.

Parra-Vega, V. (2001). Second order sliding mode control for robot arms with time base generators for finite-time tracking. *Dynamics and Control*.

Parra-Vega, V. and Arimoto, S. (1996). A passivity-based adaptive sliding mode position-force control for robot manipulators. *International Journal of Adaptive Control and Signal Processing*, 10:365–377.

Sagatun, S.I.; Fossen, T. (1991). Lagrangian formulation of underwater vehicles’ dynamics. Decision Aiding for Complex Systems, IEEE.

Smallwood, D.A.; Whitcomb, L. (2001). Toward model based dynamic positioning of underwater robotic vehicles. volume 2. OCEANS,IEEE Conference and Exhibition.

Smallwood, D.A.; Whitcomb, L. (2004). Model-based dynamic positioning of underwater robotic vehicles: theory and experiment. *IEEE Journal of Oceanic Engineering*, 29(1).

Spong M.W., V. M. (1989). *Robot Dynamics and Control*.

Yoerger, D.; Slotine, J. (1985). Robust trajectory control of underwater vehicles. *Oceanic Engineering, IEEE Journal*, 10(4).

FUZZY-SYNTACTIC APPROACH TO PATTERN RECOGNITION AND SCENE ANALYSIS

Marzena Bielecka
Department of Geoinformatics and Applied Computer Science
Faculty of Geology, Geophysics and Environmental Protection
AGH University of Science and Technology, Kraków, Poland
bielecka@agh.edu.pl

Marek Skomorowski, Andrzej Bielecki
Institute of Computer Science, Jagiellonian University, Kraków, Poland
{skomorowski,bielecki}@softlab.ii.uj.edu.pl

Keywords: Syntactic pattern recognition, graph grammars, fuzzy graphs, parallel parsing, robot vision system.

Abstract: In syntactic pattern recognition an object is described by symbolic data. The problem of recognition is to determine whether the describing mathematical structure, for instance a graph, belongs to the language generated by a grammar describing the mentioned mathematical structures. So called ETPL(k) graph grammars are a known class of grammars used in pattern recognition. The approach in which ETPL(k) grammars are used was generalized by using probabilistic mechanisms in order to apply the method to recognize distorted patterns. In this paper the next step of the method generalization is proposed. The ETPL(k) grammars are improved by fuzzy sets theory. It turns out that the mentioned probabilistic approach can be regarded as a special case of the proposed one. Applications to robotics are considered as well.

1 INTRODUCTION

The fundamental idea in syntactic pattern recognition is using of symbolic data like strings, trees and graphs for representation of a class of recognized objects (Chen et al., 1991; Fu, 1982; Jakubowski, 1997; Jakubowski and Stąpor, 1999). The general scheme of syntactic pattern recognition and a scene analysis is following (Fu, 1982). After pre-processing the recognized object is segmented in order to recognize the primitives the pattern consists of and relations between them. Decision whether the analysed pattern representation belongs to the class of objects describing by a given grammar is made basing on the parsing algorithm. This classical approach can be applied in robotics, for instance in vision systems and in manufacturing for description and analysis of the production process (Chen et al., 1991; Yeh et al. 1993). It seems also be effective for applying in multi-agent systems, particularly in embodied cognitive ones because such agents should be equipped with symbolic and explicit representation of the surrounding world in order to analyse the scene they act on (Ferber, 1999; Scheier and Pfeifer, 1999). For instance in (Kok et al., 2005) so called coordination graphs are used for solving a behaviour management problem in a multi-robot system. In this graph a node represents an agent and an edge indices that a corresponding agents have to coordinate their actions.

The use of graph grammars for syntactic pattern recognition is relatively rare because of difficulties in building a syntax analyser of such grammars. Therefore every result in building efficient parser for graph grammars is valuable. An example of such result is a parser for, so called, *ETPL*(*k*) (*embedding transformation-preserving production-ordered k-left nodes unambiguous*) grammars introduced in (Flasiński, 1993 and 1998). An efficient parsing algorithm for *ETPL*(*k*) graph grammars, which the computational complexity is $O(n^2)$, has been constructed in (Flasiński, 1993). The so-called *IE* (*indexed edge-unambiguous*) graphs have been defined in (Flasiński, 1993) for a description of pattern (scenes) in syntactic pattern recognition. Nodes in an *IE* graph denote pattern primitives. Edges between two nodes in an *IE* graph represent spatial relations between pattern primitives. However, in practice, structural descriptions may contain pattern distortions. An idea of a probabilistic improvement of syntactic recognition of distorted patterns represented by graphs is described in (Flasiński and Skomorowski, 1998, Skomorowski 1998) and (Skomorowski, 1999). To take into account all variations of a distorted pattern under study, a probabilistic description of the pattern was introduced. A *random IE* graph approach (Flasiński and Skomorowski, 1998, Skomorowski, 1999, Skomorowski, 2000) is proposed for such a description and an efficient parsing algorithm for *IE* graphs is presented. Its computational complexity is $O(n^2)$ as well.

The purpose of this paper is to present an idea of approach to syntactic recognition of fuzzy patterns represented by fuzzy *IE* graphs, followed the example of random *IE* graphs used for distorted pattern description. It turns out that, in a way, the fuzzy approach is a generalization of the probabilistic one. Fuzziness allows us not only described distortions in analysed patterns but also give us possibility to describe in proper way patterns that can not be presented univocally. Furthermore there are a wide class of problems in which objects and/or spatial relations are described by fuzzy sets.

2 MOTIVATIONS

In this section a few example, in which the fuzzy-syntactic approach seems to be natural, are presented.

2.1 First Example

Assume that during a manufacturing process a robotic inspection system checks type of a hole in a making elements, for instance plates, and spatial relations between holes. Assume also that there are a few standard types of holes and circular and quadratic ones are among them – Fig.1a. Let, furthermore, the inspection system be based on a syntactic pattern recognition approach in which the holes are represented by nodes of graphs and spatial relations between holes by graph edges – see Fig.1b. A quadratic hole with rounded vertices can be regarded as a fuzzy object with partial membership to classes of both circular and quadratic holes – see Fig.1. In this example nodes description as fuzzy sets is a natural approach. In this case membership functions describing fuzzy sets can be define basing on axiomatic method (Bielecka, 2006).

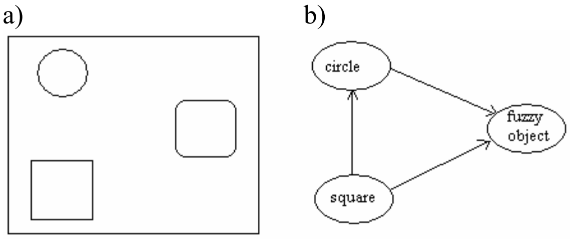


Figure 1: Holes in plate and their graph representation.

2.2 Second Example

Considering the previous example assume that robotic inspection of technological process is based on statistical distribution of inaccuracy frequencies (Flasiński and Skomorowski, 1998, Noori and Radford, 1995). If a hole is made in a sufficient accuracy it is accept by the system. Not only the hole shape but also its location should be taken into consider. Since inaccuracies of holes location influence each other, the simple statistical analysis can be insufficient to make a decision. In such a case a fuzzy inference can be applied. Then, holes and their locations can be represented by fuzzy sets and membership functions can be calculated using the statistical distribution according to the methodology described in (Bielecka, 2006). Let, like in the first example, the inspection system is based on a syntactic pattern recognition in which the holes are represented by nodes of graphs and spatial relations between holes by graph edges. In this example both the graph nodes and its edges would be described as fuzzy sets. Automatic focusing vision system for inspection of size and shape and positions of small holes in the context of precision engineering, described in (Han and Han, 1999), is an example of a system performing such type of task.

2.3 Third Example

Consider an autonomous mobile agent. Assume that it has to navigated in an unchanging environment. A helicopter flying autonomously in a textured urban environment is an example of such agent (Muratet et al., 2005). As it has been already mentioned it should be equipped with symbolic representation of the surrounding world in order to analyse the scene they act on (Ferber, 1999; Scheier and Pfeifer, 1999). Let its vision system be a syntactic one based on graph representation of the spatial relationships between obstacles the agent should navigate among. Let according to, for instance, the optimization requirements, the system prefers one direction but admits also another ones allowing to navigate without collision. In such a case the scene would be represented by a classical (i.e. not fuzzy) *IE* graph but directions the agent can choice would be represented by fuzzy sets – see Fig.2. The decision making system would be based on fuzzy inference.

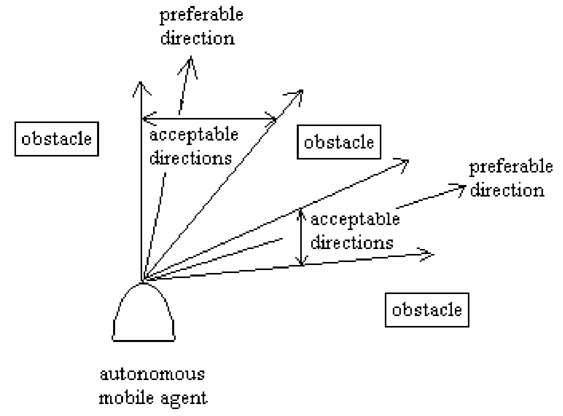


Figure 2: Detection of possible directions of motion.

2.4 Fourth Example

Let us consider computer-aided analysis and recognition of pathological wrist bone lesions (Tadeusiewicz and Ogiela, 2005; Ogiela et al., 2006). This method consists on analysis of the structure of the said bones based on palm radiological images. During pre-processing operations in the examined X-ray images the bones contours were separated and a graph representing bones and spatial relation between them was spanned. In the beginning, spatial relationships given by the graph edges were represented by single directions (Tadeusiewicz and Ogiela, 2005) but later each basic spatial relationship was represented as angular interval (Ogiela et al., 2006). The second approach can be interpreted in such a way that every basic spatial relationship is described as a fuzzy set for which its membership function has positive values on the specified angular interval and is equal to zero outside this interval. It should be mentioned that in (Ogiela et al., 2006) such interpretation was not considered.

Recapitulating, four examples in which various aspects of possibility of improve syntactic approach by fuzzy sets has been discussed. The classical *IE* graphs can be generalized by including fuzzy sets to description their nodes (example 1), edges (example 4), both the nodes and edges (example 2) and fuzzy inference approach can be applied to classical graphs (i.e. non-fuzzy ones) – example 3.

3 FUZZY IE GRAPHS

Recall a definition of *IE* graph (Flasiński, 1993).

Definition 3.1

An *IE* graph is a quintuple H=(V, E, Σ, Γ, φ) where:
V is a finite, non-empty set of nodes of a graph with assigned indexes in univocally way,
Σ is a finite, non-empty set of node labels,
Γ is a finite, non-empty set of edge labels,
E is a set of graph edges represented by triplet (v, λ, w) where v, w∈V, λ∈Γ and an index of v is smaller than an index of w,
φ:V→Σ is a nodes labeling function.

Let us assume that, due to pattern fuzziness, possible *IE* graphs associated with a given example pattern (scene) may look like *IE* graphs shown in Fig.3.

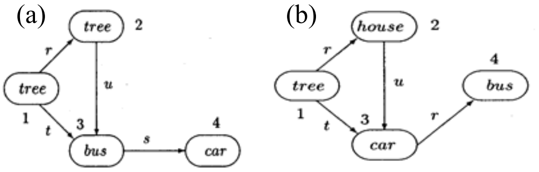


Figure 3: Possible IE graphs describing a given scene.

In the case of the *IE* graph *shown in* Fig.3a fuzziness concerns pattern primitives represented by the node 2 labeled by *tree* and the node 3 labeled by *bus*. In the case of the *IE* graph *shown in* Fig.3b fuzziness concerns a pattern primitive represented by the node 4 labeled by *bus* and a spatial relation between pattern primitives represented by the edge connecting the node 3 with the node 4. Assume that both labeled objects in nodes of a graph and spatial relations are represented by fuzzy sets of a first order (Zadeh L.A., 1965) with membership functions μ_i and ν_i respectively. Let, furthermore, the set of all objects Σ be *m*-elemental and the set of all spatial relations be *k*-elemental. Let us define, informally, a *fuzzy IE* graph as an *IE* graph in which nodes labels are replaced by a vector $\boldsymbol{\mu} = [\mu_1,...,\mu_m]$ of values of membership functions μ_i, $i\in\{1,...,m\}$ and edges labels are replaced by vector $\boldsymbol{\nu} = [\nu_1,...,\nu_k]$ of values of membership functions ν_j, $j\in\{1,...,k\}$.

Let propose a formal definition of a *fuzzy IE* graph

Definition 3.2

A *fuzzy IE* graph is a quintuple H=(V, E, Σ, Γ, Φ) where:

V is a finite, non-empty set of nodes of a graph with assigned indices in univocal way,

Σ is a finite, non-empty set of node labels, containing, say, *n* elements,

Γ is a finite, non-empty set of edge labels, containing, say, *k* elements,

E = V×Θ×V is a set of fuzzy graph edges represented by triplet (v, Θ^s, w) where v, w∈V and i(v) < i(w) i.e. an index of v is smaller than an index of w, $\Theta^s = [\Theta_1^s,...,\Theta_k^s]$ is represented by $[(\lambda_1, \nu_1^s),...,(\lambda_k, \nu_k^s\}]$ where ν_i^s is a value of a membership function of succeeding edge labels for a s-th edge,

Φ: V → Π_1×…×Π_m where for every nodes $\Phi(\nu_i) = (\Pi_1^i,...,\Pi_m^i)$ where $\Pi_k^i = (\sigma_k, \mu_k^i)$, $\sigma_k \in \Sigma$, μ_k^i is a value of a membership function of succeeding nodes labels for an i-th node.

The fuzzy measure of an outcome *IE* graph, obtained form a given *fuzzy IE* graph, is equal to the value of T-norm of the values components of the node and edge vectors. Recall axiomatic definition of T-norms which is given in, for instance, (Rutkowski, 2005) - definition 4.22, page 80.

Definition 3.3

T-norm is a function T:[0,1]×[0,1]→[0,1] satisfying the following axioms:

(i) T(a,b) = T(b,a),
(ii) T(T(a,b),c) = T(a,T(b,c)),
(iii) if a ≤ b and c ≤ d then T(a,b) ≤ T(c,d),
(iv) T(a,0) = 0 and T(a,1) = a.

Theorem

The functions T_m and T_u given by the formulae

$$T_m(a,b) = \min\{a,b\} \text{ and } T_u(a,b) = a\cdot b \qquad (1)$$

are T-norms. The function T_w given by the formulae

$$T_w(a,1) = a,\ T_w(a,b) = 0 \text{ for } a\neq 1 \text{ and } b\neq 1 \qquad (2)$$

is a T-norm as well. Furthermore, for every a,b∈[0,1] if a function T is a T-norm then

$$T_w(a,b) \leq T(a,b) \leq T_m(a,b) \qquad (3)$$

Thanks to the property (ii) in Definition 3.3 T-norm being a function of *n* variables can be introduced:

$$T(a_1,...,a_n) = \underset{i=1}{\overset{n}{T}}(a_i) = T(\underset{i=1}{\overset{n-1}{T}}(a_i), a_n) \qquad (4)$$

Having a *fuzzy IE* graph *R* the fuzzy measure of an outcome graph *r* is calculated as

$$\lambda(r) = T(\underset{p=1}{\overset{P}{T}}(\mu^{\alpha}_{f(\alpha)}), \underset{s=1}{\overset{S}{T}}(\nu^{\beta}_{g(\beta)})) \qquad (5)$$

where α is a number of a regarded node, β is a number of an edge, f(α) - is a chosen component number of a vector μ^{α} whereas g(β) is a number of component of a vector ν^{β}. If a product is used as a T-norm then the presented parsing (see section 4) is identical as the random parsing described in (Skomorowski, 1998). In calculations presented in the next section the minimum T_m-norm is used.

4 PARALLEL PARSING

Given an unknown pattern represented by a *fuzzy IE* graph *R*, the problem of recognition of a pattern under study is to determine if an outcome *IE* graph *r*, obtained from the *fuzzy IE* graph *R*, belongs to a graph language *L*(*G*) generated by an *ETPL*(*k*) graph grammar *G*. In the proposed parallel and cut-off strategy of *fuzzy IE* graph parsing for an efficient, that is with the computational complexity *O*(n^2), analysis of fuzzy patterns (scenes) a number of simultaneously derived graphs is equal to a certain number *limit*. In this case, derived graphs spread through the search tree, but only the best, that is with maximum measure value, *limit* graphs are expanded. Let us consider a graph shown in Fig.4a and a production shown in Fig.4b. Suppose that the embedding transformation for the production shown in Fig.4b is *C*(*r*, *input*) = {(*d*, *b*, *r*, *input*)} and *C*(*u*, *output*) = {(*e*, *B*, *r*, *input*)}. During a derivation, a non-terminal *A* in the node 2 of a graph shown in Fig.4a is removed and the graph of the production shown in Fig.4b is put in the place of the removed non-terminal *A*. The first item of the embedding transformation for the production: *C*(*r*,*input*) = {(*d*, *b*, *r*, *input*)} means that the edge *r* of the graph shown in Fig.4a should connect the node *d* of the production graph with the node *b* of the graph shown in Fig.4a. The second item of the embedding transformation for the production: *C*(*u*, *output*) = {(*e*, *B*, *r*, *input*)} means that the edge *u* of the graph shown in Fig. 4a should be replaced by the edge *r* connecting the node *e* of the production graph with the node B of the graph shown in Fig.4a. Thus, after the application of the production shown in Fig.4b to the node indexed by 2 of the graph shown in Fig.4a we obtain a graph shown in Fig.4c.

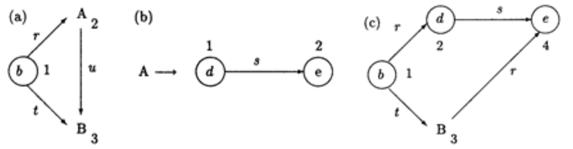


Figure 4: An example derivation step in an *ETPL*(*k*) graph grammar.

Suppose that we analyze an unknown fuzzy pattern represented by a fuzzy IE graph shown in Fig.5. (for clarity, only non-zero membership functions vectors components are specified).

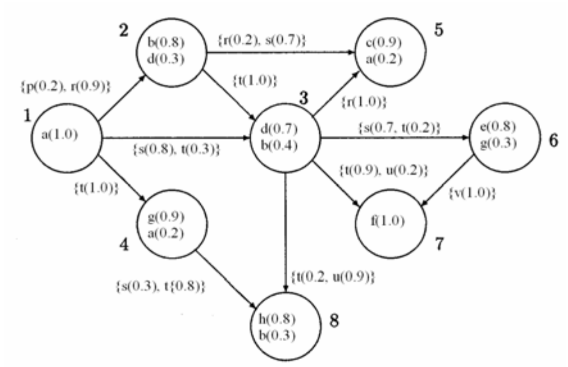


Figure 5: An example *fuzzy IE* graph representing an unknown distorted pattern.

Let us assume that a number of simultaneously derived graphs is equal to 2 (that is *limit* = 2). Furthermore let us assume that we are given an *ETPL*(*k*) graph grammar *G* with a starting graph *Z* shown in Fig.6 and a set of productions shown in Fig.7.

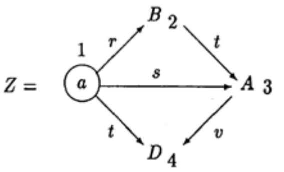
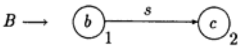
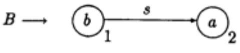


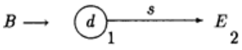
Figure 6: A starting graph *Z* of an example *ETPL*(*k*) graph grammar *G*.

(1) 

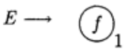
C(r, input) = {(b, a, r, input)}
C(t, output) = {(b, A, t, output), (c, A, r, input)}

(2) 

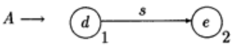
C(r, input) = {(b, a, r, input)}
C(t, output) = {(b, A, t, output), (a, A, r, input)}

(3) 

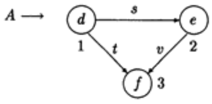
C(r, input) = {(d, a, r, input)}
C(t, output) = {(d, A, t, output), (E, A, r, input)}

(4) 

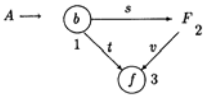
C(s, input) = {(d, a, s, input)}

(5) 

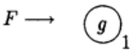
C(s, input) = {(d, a, s, input)}, C(t, input) = {(d, b, t, input)}
C(r, output) = {(d, c ,r, output)}, C(v, output) = {(d, D ,v, output)}

(6) 

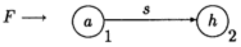
C(s, input) = {(d, a, s, input)}, C(t, input) = {(d, b, t, input)}
C(r, output) = {(d, c ,r, output), (d, a, r, output)}
C(v, output) ={(d, D ,v, output)}

(7) 

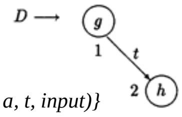
C(s, input) = {(b, a, s, input)}, C(t, input) = {(b, b, t, input)}
C(r, output) = {(b, c, r, output),(b, a, r, output)}
C(v, output) = {(b, D , v, output)}

(8) 

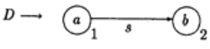
C(s, input) = {(g, b, s, input)}
C(v, output) = {(g, f, v, output)}

(9) 

C(s, input) = {(a, b, s, input)}, C(v, output) = {(a, f, v, output)}

(10) 

C(t, input) = {(g, a, t, input)}
C(v, input) = {(h, d ,u, input), (h, b, u, input)}

(11) 

C(t, input) = {(a, a, t, input)}
C(v, input) = {(b, d, u, input), (b, b, u, input)}

Figure 7: A set of productions of an *ETPL*(*k*) graph. grammar *G*.

In the first step of the derivation, after the application of the production (1), shown in Fig.7, to the node indexed by 2 of the starting graph *Z*, shown in Fig.6, we obtain a graph q_1 shown in Fig.8a. Similarly, after the application of the production (2) to the node indexed by 2 of the starting graph *Z* we obtain a graph q_2 shown in Fig.8b. The graphs q_1 and q_2 (Fig.8) are admissible for further derivation, that is they can be outcome graphs obtained from the *fuzzy IE* graph shown in Fig.5. The application of the production (3) to the node indexed by 2 of the starting graph *Z* does not lead to a graph which can be an outcome graph obtained from the *fuzzy IE* graph shown in Fig.5. Thus, a graph obtained after the application of the production (3) to the node indexed by 2 of the starting graph *Z* is not admissible for further derivation. As in the analyzed example a number of simultaneously derived graphs is equal to 2 we expand the graphs q_1 and q_2 in the second step of derivation.

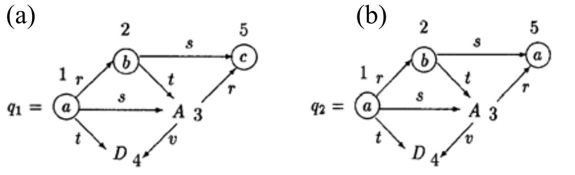


Figure 8: Derived graphs q_1 and q_2.

In the second step of derivation, after the application of the productions (6) and (7) (Fig.7) to the node indexed by 3 of the graph q_1 (Fig.8a) we obtain graphs $q_{1,6}$ and $q_{1,7}$ shown in Fig.9. Similarly, after the application of the productions (6) and (7) to the node indexed by 3 of the graph q_2 (Fig.8b) we obtain graphs $q_{2,6}$ and $q_{2,7}$ shown in Fig.10. The application of the production (5) to the node indexed by 3 of the graphs q_1 and q_2 (Fig.8) leads to graphs which can not be outcome graphs obtained from the *fuzzy IE* graph shown in Fig.5, as they miss the node indexed by 7 and labeled by *f* of the *fuzzy IE* graph shown in Fig.5. Thus, graphs obtained after the application of the production (5) to the nodes indexed by 3 of the graphs q_1 and q_2 (Fig.8) are not admissible for further derivation. The graphs $q_{1,6}$, $q_{1,7}$ (Fig.9) and $q_{2,6}$, $q_{2,7}$ (Fig.10) are admissible for further derivation, that is they can be outcome graphs obtained from the *fuzzy IE* graph shown in Fig.5.

Because in the analyzed example a number of simultaneously derived graphs is equal to 2 we should choose only two graphs from among the graphs $q_{1,6}$, $q_{1,7}$ (Fig.9) and $q_{2,6}$, $q_{2,7}$ (Fig.8) for further derivation. In order to do it, compute the following values: $\lambda(q_{1,6}) = 0.7$ and $\lambda(q_{1,7}) = 0.3$. The contributions of nodes indexed by 6 of the graphs $q_{1,6}$ and $q_{1,7}$ are not taken into account in this case as the node indexed by 6 and labeled by *F* in the graph $q_{1,7}$ is not a terminal one. Consequently, the contribution of the edge connecting nodes indexed by 3 and 6 as well as the contribution of the edge connecting nodes indexed by 6 and 7 in the graphs $q_{1,6}$ and $q_{1,7}$ are not taken into account. Similarly, we compute the following values: $\lambda(q_{2,6}) = 0.2$ and $\lambda(q_{2,7}) = 0.2$. As $\lambda(q_{1,6}) > \lambda(q_{1,7}) > \lambda(q_{2,6}) = \lambda(q_{2,7})$ we choose the graphs $q_{1,6}$ and $q_{1,7}$ for further derivation, that is we choose two graphs with maximum value (*limit* = 2). Similarly, in two next steps of derivation the final outcome *IE* graph is obtained – see Fig.11. The derived graph $q_{1,7,10,8}$ is also an outcome *IE* graph obtained from the parsed *fuzzy IE* graph shown in Fig.5.

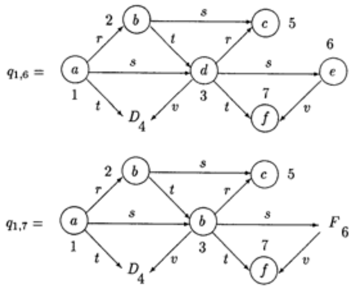


Figure 9: Derived graphs $q_{1,6}$ and $q_{1,7}$.

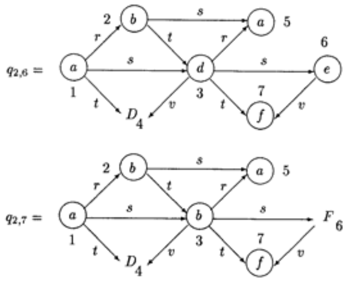


Figure 10: Derived graphs $q_{2,6}$ and $q_{2,7}$.

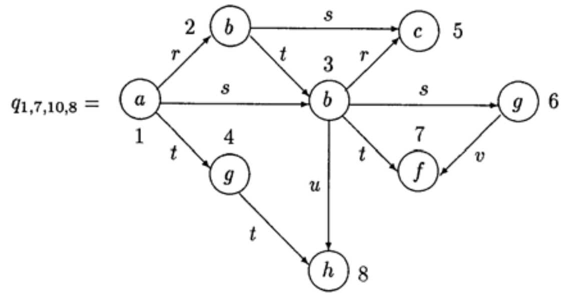


Figure 11: A derived graph $q_{1,7,10,8}$.

5 CONCLUSIONS

In this paper we have proposed an idea of a new approach to recognition of fuzzy patterns represented by graphs. The problem has been considered in the context of pattern recognition and scene analysis with references to robotics (Han and Han, 1999; Kok et al., 2005; Muratet et al., 2004; Petterson, 2005) and applications in medicine (Tadeusiewicz and Ogiela, 2005, Ogiela et al., 2006). To take into account variations of a fuzzy pattern under study, a description of the analysed pattern based on fuzzy sets of the first order was introduced. The *fuzzy IE* graph has been proposed here for such a description. The idea of an efficient, that is with the computational complexity $O(n^2)$, parsing algorithm presented in (Flasiński, 1993) is extended, so that fuzzy patterns, represented by *fuzzy IE* graphs, can be recognized. In the algorithm a T-norm is used for calculation of value of membership measure of output graphs. Such solution makes that the algorithm is very flexible. In particular if arithmetic product is used as a T-norm, the algorithm is the same as the random one described in (Skomorowski, 1998).

REFERENCES

Bielecka M, 2006. A method of membership function construction in fuzzy systems. In: Cader A., Rutkowski L., Tadeusiewicz R., Żurada J. (eds): Challenging Problems of Science – Computer Science, Academic Publishing House EXIT, Warszawa, 111-117.

Chen S.L., Chen Z., Li R.K., 1991. A DGR method for extracting the topology of an upper-half profile of a turned part from CAD data, Computer Integrated Manufacturing, 4, 45-56.

Ferber J., 1999. *Multi-Agent Systems. An Introducing to Distributed Artificial Intelligence*. Addison-Wesley, Harlow.

Flasiński M., 1993. On the parsing of deterministic graph languages for syntactic pattern recognition. Pattern Recognition 26, 1-16.

Flasiński M., 1998. Properties of *NLC* graph grammars with a polynomial membership problem. Theoretical Computer Science 201**,** 189-231.

Flasiński M., Skomorowski M, 1998. Parsing of random graph languages for automated inspection in statistical-based quality assurance system. Machine Graphics and Vision 7, 565-623.

Fu, K.S., 1982. *Syntactic pattern recognition and applications*, Prentice-Hall, New York.

Han M., Han H., 1999. Automatic focusing vision system for inspection of size and shape of small hole, Journal of the Korean Society of Precision Engineering, 16, no.10, 80-86.

Jakubowski R., 1997. Structural approach to modeling and analysis of 2D-shapes, Bulletin of the Polish Academy of Sciences – Technical Sciences, 45, 373-387.

Jakubowski J., Stąpor K., 1999. Structural method in perceptual organization of 2D-curve detection, representation and analysis, Bulletin of the Polish Academy of Sciences – Technical Sciences, 47, 309-324.

Kok J.R., Spaan M.T.J., Vlassis N., 2005. Non-communicative multi-robot coordination in dynamic environments, Robotic and Autonomous Systems, 50, 99-114.

Muratet L., Doncieux S., Briere Y., Meyer J.A., 2005. A contribution to vision-based autonomous helicopter flight in urban environments, Robotics and Autonomous Systems, 50, 195-229.

Noori, H., Radford R., 1995. Production and Operation Management. Tostal Quality and Responsiveness, McGraw-Hill.

Ogiela M.R., Tadeusiewicz R., Ogiela L., 2006. Image languages in intelligent radiological palm diagnostics, Pattern Recognition, 39, 2157-2165.

Petterson Ola, 2005. Execution monitoring in robotics: A survey, Robotics and Autonomous Systems, 53, 73-88.

Rutkowski L., 2005. *Artificial Intelligence Techniques and Methods*, PWN, Warszawa (in Polish).

Scheier C., Pfeifer R., 1999. The embodied cognitive science approach. In: Tschacher W., Dauwalde J.P. (eds): Dynamics, Synergetics, Autonomous Agents, Studies of Nonlinear Phenomena and Life Science Vol.8., World Scientific, Singapore, New Jersey, London, Hong Kong, 159-179.

Skomorowski M., 1998. Parsing of random graphs for scene analysis, Machine Graphics and Vision 7**,** 313-323.

Skomorowski M., 1999. Use of random graph parsing for scene labeling by probabilistic relaxation, Pattern Recognition Letters 20**,** 949-956.

Skomorowski M., 2006. Syntactic recognition of syntactic patterns by means of random graph parsing, Pattern Recognition Letters 28**,** 572-581.

Tadeusiewicz R., Ogiela M.R., 2005. Picture languages in automatic radiological palm interpretation, International Journal of Applied Mathematics and Computer Science, 15, 305-312.

Yeh S., Kamran M., Nnaji B., 1993. CAD-based automatic object recognition, Journal of Design and Manufacturing, 3, 57-73.

Zadeh L.A., 1965. Fuzzy sets, Information and Control, 8, 338-353.

VISUAL ALIGNMENT ROBOT SYSTEM: KINEMATICS, PATTERN RECOGNITION, AND CONTROL

SangJoo Kwon and Chansik Park
School of Aerospace and Mechanical Engineering, Korea Aerospace University, Goyang-city, 412-791, Korea
sjkwon@hau.ac.kr, creepworld@hau.ac.kr

Keywords: Visual alignment, robotics, parallel mechanism, precision control, pattern recognition.

Abstract: The visual alignment robot system for display and semiconductor fabrication process largely consists of multi-axes precision stage and vision peripherals. One of the central issues in a display or semiconductor mass production line is how to reduce the overall tact time by making a progress in the alignment technology between the mask and panel. In this paper, we suggest the kinematics of the 4PPR parallel alignment mechanism with four limbs unlike usual three limb cases and an effective pattern recognition algorithm for alignment mark recognition. The inverse kinematic solution determines the moving distances of joint actuators for an identified mask-panel misalignment. Also, the proposed alignment mark detection method enables considerable reduction in computation time compared with well-known pattern matching algorithms.

1 INTRODUCTION

In the flat panel display and semiconductor industry, the alignment process between mask and panel is considered as a core technology which determines the quality of products and the productivity of a manufacturing line. As the sizes of panel and mask in the next generation products increases but the dot pitch becomes smaller, the alignment systems must fulfill more strict requirements in load capacity and control precision. The alignment system largely has two subsystems. One is the multi-axes robotic stage to move the mask in a desired location with acceptable alignment errors and the other one is the vision system to recognize the alignment marks printed in mask and panel surfaces. In a display or semiconductor production line, the alignment systems are laid out in series as subsystems of pre-processing and post-processing equipments such as evaporation, lithography, and joining processes.

The alignment stage has at least three active joints to determine planar three degrees of freedom alignment motions. It usually adopts a parallel mechanism, specifically when it is used for display panel alignment, since it has the advantage of high stiffness and high load capacity. In this paper, we are to discuss the inverse kinematics of the parallel stage which has four prismatic-prismatic-revolute (4PPR) limbs where all the base revolute joints are active ones. For the same-sized moving platform, the four-limb mechanism brings higher stiffness and load capacity comparing with normal three-limb stages but inevitably it leads to a more difficult control problem. Although a commercial alignment stage with four driving axes was announced in (Hephaist Ltd., 2004), reports on the kinematics and control can be rarely found.

The next issue is the vision algorithm to extract the position of alignment marks. In many machine vision systems, the normalized gray-scale correlation (NGC) method (Manickam et al., 2000) has been used as a representative template matching algorithm. However, it requires long computation time since all pixels in the template image are compared in the matching process. An alternative to reduce the computation time is the point correlation (PC) algorithm (Krattenthaler et al., 1994) but it is still weak to the rotation of object and the change of illumination condition. As an another, the edge-based point correlation (Kang and Lho, 2003) was proposed to mitigate the effect of illumination change. In fact, commercial vision libraries, e.g., (Cognex Ltd., 2004), are adopted in many visual alignment systems considering the stability of the vision algorithm. However, they are computationally inefficient from the point of view that they have overspec for the monotonous vision environment of alignment systems (e.g., simple mark shape, fine light, and uniform background). In this paper, by incorporating the binarization and labeling algorithm (Gonzalez and Wood, 2002) together and designing a geometric template matching scheme instead of the general methods: NGC (Manickam et al., 2000) and PC (Krattenthaler et al., 1994), an efficient pattern recognition algorithm for alignment marks is suggested, which can greatly reduce vision processing time comparing with commercial products.

Related to the autonomous alignment system, several articles can be found. A two step alignment algorithm was suggested for wafer dicing process (Kim et al., 2004) and a self-alignment method for wafers was investigated using capillary forces (Martin, 2001). A visionless alignment method was also studied (Kanjilal, 1995). As a trial to improve the alignment speed, an additional sensor was integrated (Umminger and Sodini, 1995) and a modified template matching algorithm was presented (Lai and Fang, 2002).

2 VISUAL ALIGNMENT SYSTEM

2.1 System Configuration

Figure 1 shows the schematic of a PC-based experimental setup for the control of visual alignment system. Broadly speaking, it consists of the vision system to detect alignment marks on mask and panel and the stage control system to compensate misalignments. The vision system has normally two CCD cameras and optical lenses, illumination equipment, and frame grabber board to capture mark images. In real production lines, prior to the visual alignment, the pre-alignment process puts the marks of mask and panel into the field-of-view of CCD Cameras. Then, the vision system determines the mark positions in the global frame and the misalignment distance in camera coordinate can be converted into the moving distances of joint actuators through the inverse kinematic solution of alignment stage.

As denoted in Fig 2, the feedback control system in the alignment stage has the hierarchical feedback loops, where the outer visual servoing loop determines the misalignment quantity between mask and panel and the inner joint control loop actively compensates it. Due to system uncertainties such as friction and backlash, the alignment process is usually not completed by one visual feedback but a few cycles are repeated.

2.2 Parallel Alignment Stage

As the size of flat panel displays including TFT/LCD, PDP, and OLED becomes larger and larger, the alignment stage is required to have higher load capacity and wider moving platform. In this regard, the motion control performance of alignment stage is directly related to the productivity of the manufacturing process.

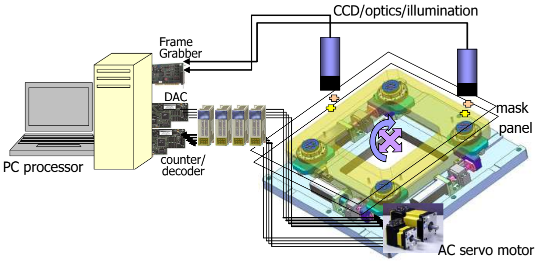


Figure 1: Schematic of visual alignment control system.

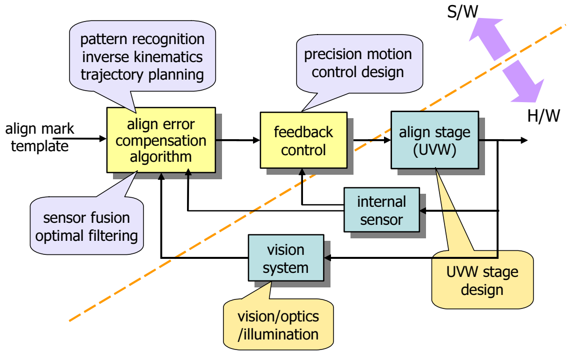

S/W
H/W
pattern recognition inverse kinematics trajectory planning
precision motion control design
align mark template
align error compensation algorithm
feedback control
align stage (UVW)
internal sensor
sensor fusion optimal filtering
vision system
UVW stage design
vision/optics /illumination

Figure 2: S/W and H/W components of visual alignment system.

The advantages of parallel manipulation mechanisms comparing with serial ones are i) high payload-to-weight ratio, ii) high structural rigidity since the payload is carried by several limbs, iii) high manipulation accuracy since active joints are distributed in parallel and the joint errors are non-cumulative, and iv) simple inverse kinematic solution. But, they suffer from smaller workspace and singular configurations. Although the movable range of the parallel stage is very small (usually a few *mms*), it is actually enough to compensate misalignments between mask and panel.

To position the moving platform (in which the mask is mounted) in a planar location, the task space of alignment stage must have at least three degrees of freedom and then it requires at least three active joints. Hence, it is common for the parallel alignment stage to have three active limbs. However, if an extra driving limb is added to support the moving platform, the stiffness and load capacity can be much increased. The visual alignment testbed with four driving axes is shown in Fig. 3, where the motion of the rotary motor in each limb is transferred to the moving platform through a ballscrew, cross-roller guide (for planar translational motion) and cross-roller ring(for rotational motion).

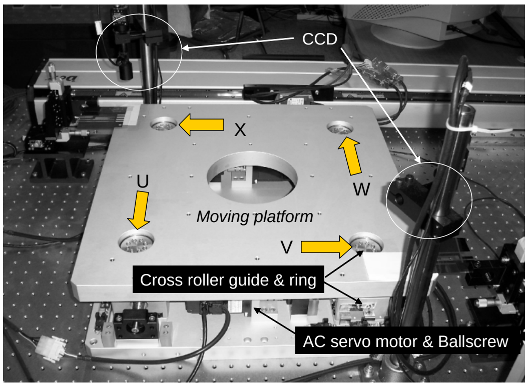


Figure 3: Visual alignment stage with four active joints.

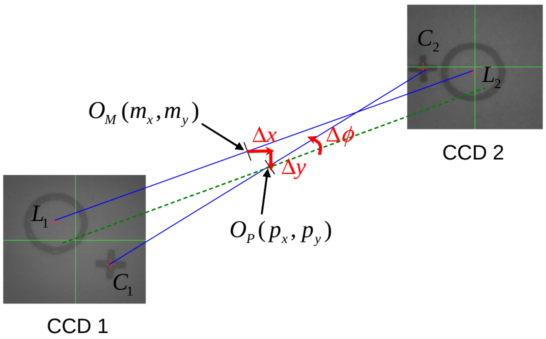


Figure 4: Determination of misaligned posture between mask and panel (circle: mask marks, cross: panel marks).

3 KINEMATICS OF 4PPR PARALLEL MECHANISM

3.1 Determination of Misalignments

When the centroids of alignment marks of panel and mask have been obtained through the image processing, it is trivial to get the misaligned posture between panel and mask. In Fig. 4, let the two align marks in the panel from the respective CCD image have the centroids of $C_1 = (x_{C_1}, y_{C_1})$ and $C_2 = (x_{C_2}, y_{C_2})$ and those in the mask have $L_1 = (x_{L_1}, y_{L_1})$ and $L_2 = (x_{L_2}, y_{L_2})$ in the global frame. Then, the center of line connecting two centroids can be written as

$$p_x = (x_{C_1} + x_{C_2})/2, \; p_y = (y_{C_1} + y_{C_2})/2 \qquad (1)$$

for the panel marks and also

$$m_x = (x_{L_1} + x_{L_2})/2, \; m_y = (y_{L_1} + y_{L_2})/2 \qquad (2)$$

for the mask ones. If the mask is to be aligned to the panel, the misaligned distance between mask and panel is given by

$$\Delta x = p_x - m_x = \frac{x_{C_1} - x_{L_1} + x_{C_2} - x_{L_2}}{2} \qquad (3)$$

$$\Delta y = p_y - m_y = \frac{y_{C_1} - y_{L_1} + y_{C_2} - y_{L_2}}{2} \qquad (4)$$

$$\Delta\phi = \tan^{-1}\left[\frac{y_{C_2} - y_{C_1}}{x_{C_2} - x_{C_1}}\right] - \tan^{-1}\left[\frac{y_{L_2} - y_{L_1}}{x_{L_2} - x_{L_1}}\right] \qquad (5)$$

in (x, y) directions and orientation, respectively.

3.2 Mobility of 4ppr Alignment Stage

The degrees of freedom of a mechanism can be found by the Grubler criterion (Tsai, 1999):

$$F = \lambda(n - j - 1) + \sum_i f_i - f_p, \qquad (6)$$

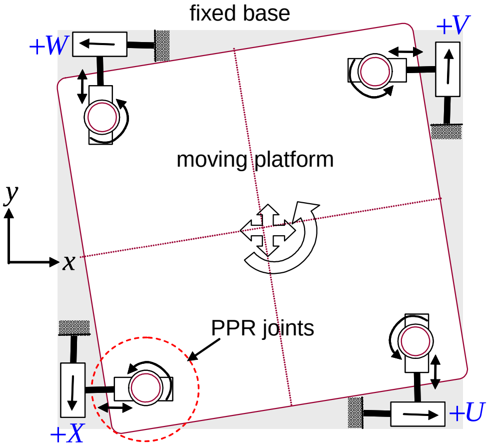


Figure 5: Planar 3-DOF, 4PPR parallel mechanim.

where λ is the dimension of the space in which the mechanism is intended to function, n the number of links, j the number of joints, f_i degrees of relative motion permitted by joint i, and f_p the passive degrees of freedom. In the case of a planar 4PPR parallel mechanism shown in Fig. 5, in which the moving platform is supported by four limbs with prismatic-prismatic-revolute joints, we have $\lambda = 3$, $n = 2 \times 4 + 1 + 1$ including fixed base and moving platform, $j = 4 \times 3$, $\sum f_i = 12 \times 1$ and $f_p = 0$. Hence, the degrees of freedom of the 4PPR mechanism is $F = 3$ as is expected.

In parallel manipulators, every limb forms closed-loop chain and the number of active limbs is typically equal to the number of degrees of freedom of the moving platform. Moreover, each limb has the constraint of having more joints than the degrees of freedom. Hence, at least three limbs must have an active joint (the first P in this case) to achieve the 3-DOF motion of moving platform and the remaining limb becomes passive. On the other hand, if all the four limbs are actuated to increase the rigidity of motion, the actuation redundancy problem is present and a sophisticated control logic is required to avoid mechanical singularity.

3.3 Inverse Kinematics

The inverse kinematic problem is to seek the moving ranges of input joints which correspond to the end-effector movement of a mechanism, i.e., moving platform in this case. In Fig. 6, the square of a fixed base is defined by the four fixed points (P,Q,R,S) and the moving platform is connected to the limbs at the four points (A,B,C,D). We assume that the two squares have the same side length of h and the global coordinates system is located at the center (O_1) of the fixed base.

Then, the positions of fixed points (P,Q,R,S) are given by $(x_P,y_P) = (-\frac{1}{2}h,-\frac{1}{2}h)$, $(x_Q,y_Q) = (\frac{1}{2}h,-\frac{1}{2}h)$, $(x_R,y_R) = (\frac{1}{2}h,\frac{1}{2}h)$, and $(x_S,y_S) = (-\frac{1}{2}h,\frac{1}{2}h)$. Moreover, assuming that the position of point A in the moving platform is known, the positions of the other three connecting points can be expressed as

$$
\begin{aligned}
(x_B,y_B) &= (x_A+h\cos\Delta\phi, y_A+h\sin\Delta\phi) && (7)\\
(x_C,y_C) &= (x_B-h\sin\Delta\phi, y_B+h\cos\Delta\phi) && (8)\\
(x_D,y_D) &= (x_A-h\sin\Delta\phi, y_A+h\cos\Delta\phi) && (9)
\end{aligned}
$$

In Fig. 6, we have $(x_A,y_A) = \overline{O_1A} = \overline{O_1O_2} + \overline{O_2A}$ where $\overline{O_1O_2} = (\Delta x, \Delta y)$ and

$$
\overline{O_2A} = R(\Delta\phi)\overline{O_1P} = \begin{bmatrix} \cos\Delta\phi & -\sin\Delta\phi \\ \sin\Delta\phi & \cos\Delta\phi \end{bmatrix} \begin{bmatrix} -\frac{1}{2}h \\ -\frac{1}{2}h \end{bmatrix} \qquad (10)
$$

Hence, the position of point A can be written by the misalignment variables as

$$
\begin{aligned}
x_A &= \Delta x - h(\cos\Delta\phi - \sin\Delta\phi)/2 && (11)\\
y_A &= \Delta y - h(\sin\Delta\phi + \cos\Delta\phi)/2 && (12)
\end{aligned}
$$

As denoted in Fig. 6, the moving distances of input prismatic joints have the following relationships (note the positive directions defined in Fig. 5): $U = x_B - x_Q$, $V = y_C - y_R$, $W = x_S - x_D$, and $X = y_P - y_A$. Now, by substituting (7)–(12) into the above expressions, we finally have

$$
\begin{aligned}
U &= \Delta x + h(\cos\Delta\phi + \sin\Delta\phi - 1)/2 && (13)\\
V &= \Delta y + h(\cos\Delta\phi + \sin\Delta\phi - 1)/2 && (14)\\
W &= -\Delta x + h(\cos\Delta\phi + \sin\Delta\phi - 1)/2 && (15)\\
X &= -\Delta y + h(\cos\Delta\phi + \sin\Delta\phi - 1)/2 && (16)
\end{aligned}
$$

3.4 Forward Kinematics

In serial mechanisms, the inverse kinematic solution is generally hard to find, but the direct kinematics is straightforward. However, the situation is reversed in parallel mechanisms, where the inverse kinematics is rather simple as shown in the former section but the direct kinematics is more complicated. The inverse kinematics is important for the control purpose while the direct kinematics is also required for the kinematic analysis of end-point.

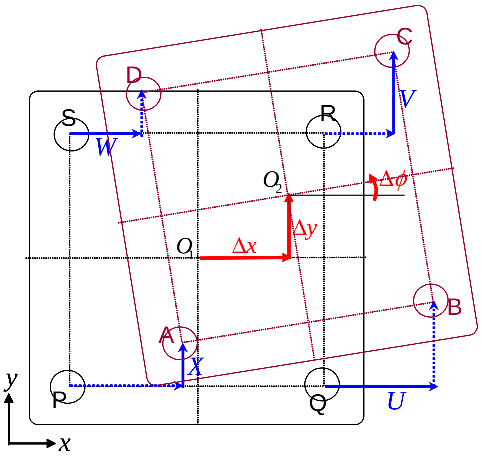


Figure 6: Moving distances of active (solid line) and passive (dotted line) joints for a misalignment posture.

Although finding out the direct kinematics in position level could be a long-time procedure, it becomes an easy task in velocity level. By differentiating (13)-(16) with respect to the time, the following Jacobian relationship between joint space velocity and task space one is given.

$$
\begin{bmatrix} \dot U \\ \dot V \\ \dot W \\ \dot X \end{bmatrix} = \begin{bmatrix} 1 & 0 & \frac{1}{2}h(\cos\Delta\phi - \sin\Delta\phi) \\ 0 & 1 & \frac{1}{2}h(\cos\Delta\phi - \sin\Delta\phi) \\ -1 & 0 & \frac{1}{2}h(\cos\Delta\phi - \sin\Delta\phi) \\ 0 & -1 & \frac{1}{2}h(\cos\Delta\phi - \sin\Delta\phi) \end{bmatrix} \begin{bmatrix} \dot x \\ \dot y \\ \dot\phi \end{bmatrix} \qquad (17)
$$

which can be simply represented by

$$
\dot q(t) = J(p)\dot p(t) \qquad (18)
$$

Considering the Jacobian $J \in \Re^{4\times3}$ in (18), the columns are linearly independent (rank(J)= 3). Hence, $J^TJ \in \Re^{3\times3}$ is invertible and there exists a left-inverse $J^+ \in \Re^{3\times4}$ such that $J^+ = (J^TJ)^{-1}J^T$ and the linear system (18) has an unique solution $\dot p$ for every $\dot q$ (Strang, 1988). In the sequel, the direct kinematic solution at current time can be determined by

$$
\begin{aligned}
& \dot p(t) = J^+\dot q(t) && (19)\\
\rightarrow\quad & p(t) = \int_{t_0}^{t} (J^TJ)^{-1}J^T\dot q(t)dt && (20)
\end{aligned}
$$

4 PATTERN RECOGNITION

A typical visual alignment system provides a structured vision environment. For example, CCD camera and illumination is under static condition and the objects to be recognized are just two kinds of alignment mark for mask and panel. Moreover, the shape of marks are usually symmetric ones such as circle, cross, rectangle, and diamond, in which case the feature points of an object can be easily determined. We consider just circle and cross shaped marks because they are most common in display and semiconductor industry. In this section, an efficient alignment mark recognition algorithm in Fig 10 is suggested by combining the conventional labeling technique and a geometric template matching method which is designed by analyzing the characteristics of alignment marks, . Basically, it is assumed that the alignment marks are in the field-of-view (FOV) of cameras after the pre-alignment process in the industrial visual alignment system.

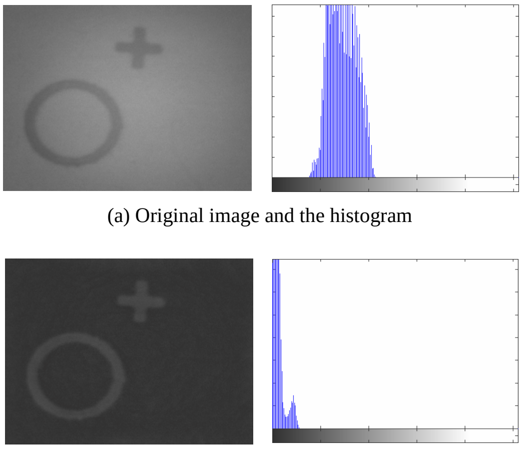
4.1 Preprocessing

Bottom-Hat Transform: Before applying the labeling algorithm, the gray image from the CCD camera is converted into the binary image. However, a proper binarization is impossible when the illumination is nonlinear or non-steady due to the limit of light source or other unexpected light disturbances. In order to eliminate these effects, we can use the bottom-hat transform which is a morphology method (Gonzalez and Wood, 2002), where the transformed image h can be calculated by

$$h = (f \cdot b) - f \tag{21}$$

where f represents the original image, b the circular morphology, and $(f \cdot b)$ the closing operation between f and b. By the closing operation, objects smaller than the size of b are eliminated and the rest of background is extracted. The size of b is usually determined experimentally. The figure 7 shows an example.

Dynamic Thresholding and Noise Filtering: To segregate marks from the background in the Bottom-Hat transformed image, we apply the binarzation algorithm. Among several methods to determine best threshold value, the following repetition method can be used: 1) Set the initial threshold T, which is the average between the maximum and the minimum of brightness from the binary image. 2) Divide the image into class G_1 and G_2 by T and calculate the averages m_1 and m_2 of them. 3) Calculate $T = (m_1 + m_2)/2$. 4) Repeat step 2 and step 3 until m_1 and m_2 are not changed.


(a) Original image and the histogram

(b) Bottom-Hat transformed image and the histogram

Figure 7: Comparison of original and transformed image and the histograms (b = 20 pixels).

Small noise patterns in the binary image can be recognized as independent areas during the labeling process. They can be eliminated by the opening and closing method:

$$\dot{f} = (f \circ b) \cdot b \tag{22}$$

where the morphology b is a small (e.g., 3 by 3) square matrix whose components are all 1, $(f \circ b)$ an opening operator, and $\dot{f}$ the filtered image.

Labeling: If the objects in a captured image are not overlapped, an easily accessible algorithm to extract them is the labeling technique. To separate marks from the noise-filtered image, we first apply the labeling algorithm to the area chained by eight-connectivity. Once the labeling process is finished, a number of areas including the marks will be labeled. As explained in Fig 10, if the labeled area is not counted, a new image should be captured by repeating the pre-alignment process. If the labeled area is only one, it is highly possible that two marks are overlapped and each mark should be extracted by applying another pattern recognition scheme. If there are 2 labeled areas, the marks of mask and panel are not overlapped. Then, the centroids of marks can be simply calculated by the center of area method:

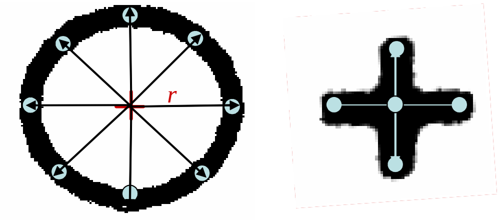
$$X_c = \frac{1}{n}\sum_{i=0}^{n-1} X_i,\ Y_c = \frac{1}{n}\sum_{i=0}^{n-1} Y_i \tag{23}$$

where X_c and Y_c represent the central point of labeled image and X_i and Y_i the horizontal and vertical positions of the pixels. Finally, if there are more than 3 labeled areas, as denoted in Fig 10, the areas which have less pixels than a specified number must be eliminated through an extra filtering process.

4.2 Geometric Template Matching

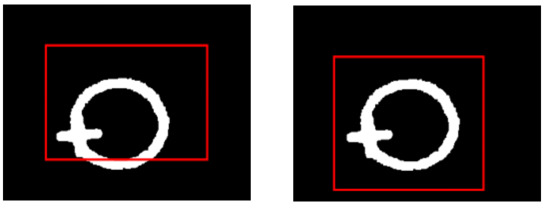
When the two marks of mask and panel are overlapped, i.e., when the number of labeled area is only one in Fig 10, the labeling algorithm alone is not enough but the marks can be separated in terms of any pattern matching method.

Since the alignment marks used in the display or semiconductor masks are very simple, their templates can be readily characterized by a few feature points. First, for the circular mark in Fig. 8(a) where the radius (r) is a unique trait, for example, the eight pixels along the circumference can be selected as the feature points. All the pixels in the matching area can be scanned by assuming them as the center of circle. Then, the centroid of circular mark can be found when all the feature pixels have the same brightness of 255 (white) in the binarized image. In reality, since the actual circular mark has a thickness, every pixel that fulfills this condition must be stored in the memory stack and the final centroid can be calculated by averaging the coordinate values. Similarly, five feature points can be chosen for the cross mark with length l as in Fig. 8(b). However, differently from the circular mark, we have to consider the rotation factor. Hence, the matching process should be performed by rotating the template from 1 to 90 degrees for all pixels in the matching area.



(a) Circular mark (b) Cross mark

Figure 8: Feature pixels of circular mark (radius = 2 mm) and cross mark (length = 1 mm).



(a) (b)

Figure 9: Matching areas for (a) circular mark and (b) cross mark.

The matching area for the circular mark is given by the rectangle in Fig. 9(a) with the size of $(M-2r)\times(N-2r)$ pixels in the $M\times N$ pixel image. Under the assumption that the exact central point of the circular mark has been found, the matching area for the cross mark is equal to $2(r+l)\times 2(r+l)$ pixels as shown in Fig. 9(b) regarding the farthest positions of the cross mark in the overlapped image.

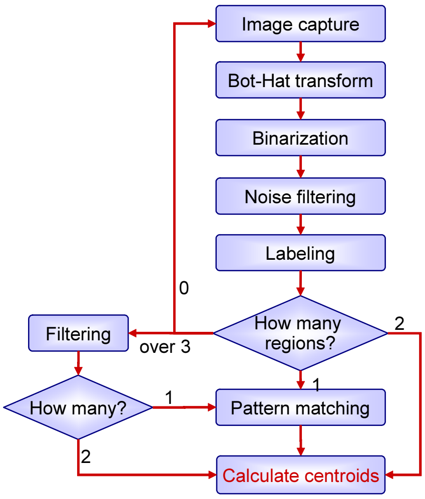


Figure 10: Overall flow of alignment mark recognition.

The normalized correlation (NC) (Manickam et al., 2000) and the point correlation (PC) algorithm (Krattenthaler et al., 1994) are most widely used template matching methods in machine vision systems. However, since the NC requires a vector operation for all pixels in the image and template to determine the correlation coefficient, it is too time consuming. Although the PC algorithm reduces the dimension of computation time, it is still not appropriate for real-time applications because it includes feature points extraction process. On the contrary, in our matching algorithm, the feature points of alignment marks are geometrically extracted based on the analysis of mark shape and the individual feature points and the image pixels at the same coordinates are directly compared without any vector operations.

The overall sequence of the suggested algorithm for alignment mark recognition is described in Fig 10, where the feature pixels of marks can be determined in advance of labeling and the computation will be finished at the labeling process when the marks are not overlapped. As far as the mark shape is geometrically simple, as is the case in semiconductor and display industry, the combined algorithm of labeling and geometric pattern matching can be considered as a reasonable way to reduce the overall tact time.

5 CONTROL

The first step in the visual alignment process is to detect the centroids of alignment marks from the raw images. The visual processing can be divided into pre-processing and image analysis. As described in the former section 4, the pre-processing includes binarization, edge detection, and noise filtering etc. In the image analysis procedure, a few object recognition techniques can be applied such as labeling and template matching. Once the centroids of marks are determined, the misalignment distance and angle between mask and panel, which is usually less than hundreds of microns, can be readily determined using geometric relationships. Given the misalignment quantity for the current image, the driving distances of joint actuators can be produced by the inverse kinematic solution for a specific parallel mechanism as shown in Section 3. Finally, the misalignment can be compensated by the joint controller as in Fig. 11, where the outer visual feedback loop should be repeated until it meets permissible alignment errors.

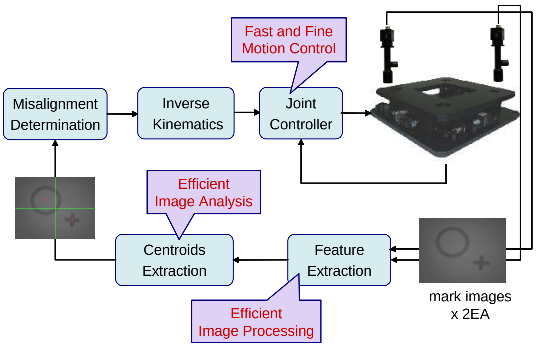


Figure 11: Vision-based look and move motion control.

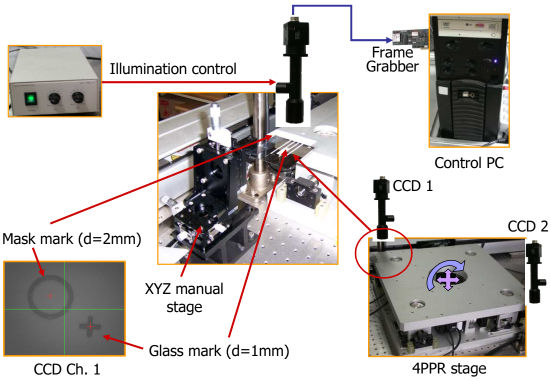
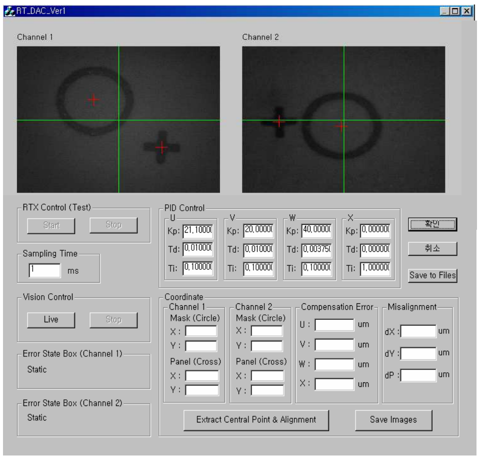


Figure 12: Experimental setup for visual alignment.

Figure 12 shows the experimental setup for the mask and panel alignment, where the CCD image has the size of 640 × 480 pixels and the field of view (FOV) is 6.4 × 4.8 *mm* and the rate of capture in the frame grabber is 15 fps. Circular marks have a diameter of 2 *mm* and thickness of 200 μm and the cross marks have the length of 1 *mm*. The graphic user interface shown Fig. 13 was developed to integrate all the functions to achieve the autonomous visual alignment.



Sampling Time: 1 ms

PID Control

	U	V	W	X
Kp:	21,10000(	20,00000(	40,00000(	0,000000(
Td:	0,010000(	0,010000(	0,00375(	0,000000(
Ti:	0,100000(	0,100000(	0,100000(	1,000000(

Error State Box (Channel 1): Static

Error State Box (Channel 2): Static

Coordinate

Channel 1: Mask (Circle) X: Y: Panel (Cross) X: Y:

Channel 2: Mask (Circle) X: Y: Panel (Cross) X: Y:

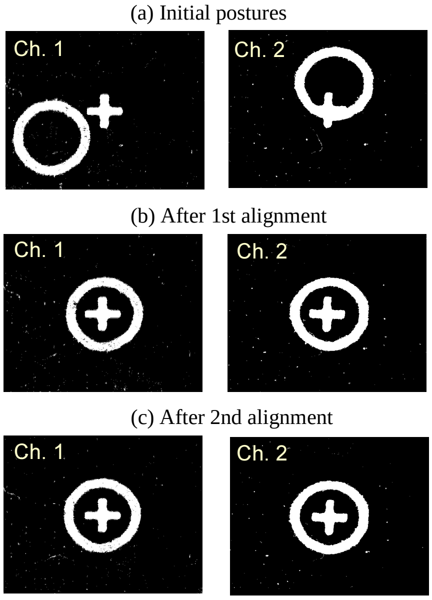
Compensation Error: U: um V: um W: um X: um

Misalignment: dX: um dY: um dP: um

Figure 13: Graphic User Interface of operating software.

First of all, we compared the performance of the alignment mark recognition algorithm in Fig. 10 with the popular NC (Manickam et al., 2000) and PC method (Krattenthaler et al., 1994). For the captured images in Fig. 13, the proposed method takes 37 *msec* on average to find the centroids of all marks including pre-processing time, while the NC and PC required 661 *msec* and 197 *msec*, respectively. As a result, the proposed method reduces one order of recognition time. As explained before, this is mainly because in the geometric template matching the vector operations for all pixels are not necessary unlike the NC and the feature points of objects are selected in advance unlike the PC. Although the geometric pattern matching is confined to simple objects, it is actually enough to extract alignment marks.

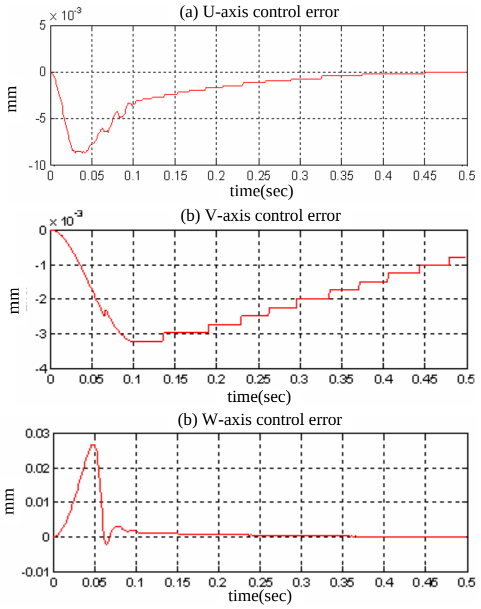
Figure 14 shows the visual alignment process for a given misaligned posture between mask and panel. As explained in Fig. 11, the inverse kinematic solutions are cast into the joint controllers as a reference input. To avoid excessive chattering, at every alignment cycle, we have applied the polynomial trajectory with rise time 0.6 *sec* for the reference values. In Fig. 14, the mask and panel were almost aligned after the 1st cycle and the 2nd cycle was activated since the misalignments in U-axis and W-axis are still over the tolerance. Considering the joint control error in Fig. 15 for the 2nd cycle, the controlled motion of V-axis is not smooth. When the reference input is too small, a stick-slip motion may occur at low velocities and this makes precision control very difficult.


(a) Initial postures

(b) After 1st alignment

(c) After 2nd alignment

Figure 14: Visual alignment experiment: the inverse kinematic solution is $(U,V,W) = (132.4, -592.6, -1367.6)\mu m$ at initial posture and $(U,V,W) = (-73.4, -3.5, 66.5)\mu m$ after the 1st alignment.


(a) U-axis control error

(b) V-axis control error

(b) W-axis control error

Figure 15: Joint control errors during the 2nd alignment (the 4th X-axis was not active).

6 CONCLUSION

In this paper, we investigated the visual alignment problem which is considered as a core requirement in flat panel display and semiconductor fabrication process. The kinematics of the 4PPR parallel mechanism was given and an efficient vision algorithm in terms of geometric template matching was developed for fast recognition of alignment marks. Through the control experiment, the proposed method was proven to be very effective.

REFERENCES

Cognex Ltd. (2004). In *http://www.cognex.com/ products/VisionTools/PatMax.asp*.

Gonzalez, R. C. and Wood, R. E. (2002). *Digital Image Processing*. Prentice Hall.

Hephaist Ltd. (2004). In *http://www.hephaist.co.jp/ e/pro/n_4stage.html*.

Kang, D. J. and Lho, T. J. (2003). Development of an edge-based point correlation algorithm for fast and stable visual inspection system. In *Journal of Control, Automation and System Engineering*. Vol. 9, No. 2, August, 2003 (in Korean).

Kanjilal, A. K. (1995). Automatic mask alignment without a microscope. In *IEEE Trans. on Instrumentation and Measurement*. Vol. 44, pp. 806-809, 1995.

Kim, H. T., Song, C. S., and Yang, H. J. (2004). 2-step algorithm of automatic alignment in wafer dicing process. In *Microelectronics Relaiability*. Vol. 44, pp. 1165-1179, 2004.

Krattenthaler, W., Mayer, K. J., and Zeiller, M. (1994). Point correlation: a reduced-cost template matching technique. In *IEEE Int. Conf. on Image Processing*. pp. 208-212, 1994.

Lai, S.-H. and Fang, M. (2002). A hybrid image alignment system for fast and precise pattern localization. In *Real-Time Imaging*. Vol. 8, pp. 23-33, 2002.

Manickam, S., Roth, S. D., and Bushman, T. (2000). Intelligent and optimal normalized correlation for high-speed pattern matching. In *Datacube Technical Paper*. Datacube Incorpolation.

Martin, B. R. (2001). Self-alignment of patterned wafers using capillary forces at a wafer-air interface. In *Advanced Functional Materials*. Vol. 11, pp.381-386, 2001.

Strang, G. (1988). *Linear Algebra and Its Applications*. College Publishers, 3rd edition.

Tsai, L.-W. (1999). *Robot Analysis: The mechanics of serial and parallel manipulators*. Wiley-Interscience.

Umminger, C. B. and Sodini, C. G. (1995). An integrated analog sensor for automatic alignment. In *IEEE Trans. on Solid-State Circuits*. Vol. 30, pp. 1382-1390, 1995.

DISTURBANCE FEED FORWARD CONTROL OF A HANDHELD PARALLEL ROBOT

Achim Wagner, Matthias Nübel, Essam Badreddin
Automation Laboratory
University of Mannheim, D-68131 Mannheim, Germany
a.wagner@ti.uni-mannheim.de

Peter P. Pott, Markus L. Schwarz
Laboratory for Biomechanics and experimental Orthopaedics, OUZ
Faculty of Medicine Mannheim, University of Heidelberg, Germany

Keywords: Robotic manipulators, Medical systems, Dynamic modelling, Disturbance rejection, Feedforward compensation.

Abstract: A model-based control approach for a surgical parallel robot is presented, which combines a local tool stabilization with a global disturbance feed forward control. The robot is held in the operator's hand during the manipulation of bones. For a precise processing the tool has to be decoupled from disturbances due to unintentional hand movements of the surgeon at the robot base. The base disturbances are transformed for a feed forward control using the inverse dynamics of the robot. Simulations show that disturbances can be reduced by many orders depending on sensor errors and delay.

1 INTRODUCTION

Parallel robots are widely used, where high stiffness, high dynamics or low error propagation over the kinematic chains is required, e.g. flight simulators (Koekebakker et al., 1998), processing machines (Tönshoff et al., 2002), positioning and stabilization platforms (Huynh, 2001), vibration isolation (Chen and McInroy, 2004) and medical manipulators (Pott et al., 2004; Wagner et al., 2004). The probably most famous Hexapod parallel kinematic structure is the Stewart-platform (Stewart, 1966) which has six degrees-of-freedom (DOF). An obvious advantage considering hand-held applications is, that a parallel robot has high potential for a lightweight construction. By fixing the most massive part of the actuators to base (Merlet, 2000), the actively positioned mass can be further decreased. This leads to a reduction of static and dynamic forces (Honegger et al., 1997; Huynh, 2001). The forces of the actuators can be transferred to the tool platform using light struts. Within the hand-held surgical robot project "Intelligent Tool Drive" (ITD), a parallel robot is designed to align a milling tool relatively to a moving bone of the patient (workpiece) and to decouple the tool from unintentional hand movements at the base. The standard procedure to control a parallel manipulator is transforming the desired tool motion into the desired leg motion with the inverse kinematics and controlling the leg motion separately on the axes level (Tönshoff et al., 2002). However, if high dynamics is required this simple kinematics approach does not lead to a high precision tool pose control, especially if the robot's base is freely movable. Therefore, full dynamic models of special parallel robots are introduced to improve the quality of a fast platform control (Riebe and Ulbrich, 2003; Honegger, 1999). The nonlinear inverse dynamic large-signal model for a parallel robot with two movable platforms is introduced in (Wagner et al., 2006). This model is used here to control the pose of the tool, while it is decoupled from base disturbances. Since the robot is movable freely in space adequate co-ordinate systems must be defined. The local control of the tool is performed in the base instantaneously coincident reference frame using the inverse kinematics description of the robot. The same reference frame can be used to achieve a dynamic feed forward compensation. However, for the compensation of the gravity influences and for a absolute position referencing a world coordinate system must be defined. In the following sections the control structure and simulations are presented to show the advantages and drawbacks of the approach.

2 SYSTEM DESCRIPTION

The goal of the ITD robot project is to adjust and to stabilize a drilling or milling tool (mounted at the tool platform) with respect to a moving bone of the patient (workpiece), while the complete machine is held in the surgeon's hand. Therefore it is necessary to isolate the tool from disturbances at the base produced by the operator's unintional hand movements. The amplitude and frequency of the human arm disturbances is available from literature (Takanokura and Sakamoto, 2001). Corresponding to the milling process with different tools a 6 DOF control of the tool is required. Assuming an additional safety margin the selected workspace of the robot ranges from -20 mm to +20 mm in the three Cartesian axes and respectively from -20^o to +20^o in the three rotational axes.

The mechanical device designed is a parallel robot with six base-fixed actuators. The CAD model of the surgical robot ITDII (Intelligent Tool Drive II) is shown in Fig. 1. The robot base has six fixed linear motors, i.e. the electrical part of the motors. Furthermore, there are guide rails, housing, and handles for the hand-held operation, building together the robot base. The sliders of the linear motors are connected to the tool platform via six lightweight struts. At the tool platform and at the sliders the struts are mounted with spherical joints.

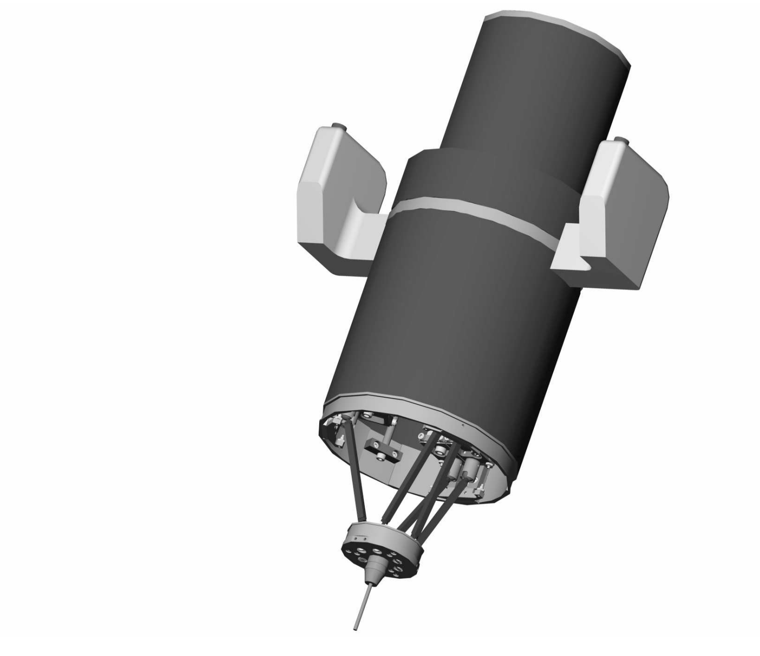


Figure 1: Hand-held surgical robot ITDII - CAD model.

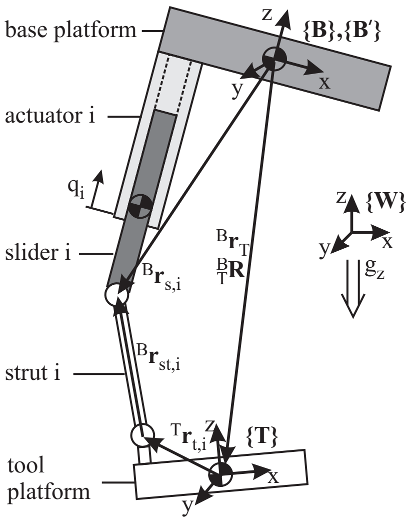


Figure 2: Topology of the parallel robot with fixed actuators.

Because the base of the robot can be moved freely in space, adequate coordinate systems have to be defined. For the calculation of the kinematic equations four coordinate frames are considered (Fig. 2): (1) the world (earth) frame $\{W\}$, (2) the tool frame $\{T\}$ in the tool's centre of gravity, (3) the movable base frame $\{B\}$ in the base's center of gravity, and (4) the instantaneously coincident base frame $\{B'\}$, which is fixed in $\{W\}$. The frame $\{B'\}$ and the frame $\{B\}$ are aligned, if the base stands still. Without loss of generality the motion of both platforms is represented in the $\{B'\}$ frame, which reduces the complexity of the kinematic description. The choice of the coordinates has no influence on the value of the inertial forces. For simplicity the gravity is calculated in the world frame. Afterwards, the resulting force can be transformed into the $\{B'\}$ frame with little effort.

The pose of the tool is defined by the position vector ${}^B\mathbf{r}_T$ and the orientation matrix ${}^B_T\mathbf{R}$ in the $\{B\}$ frame, which build together the pose vector ${}^B\mathbf{X}_T = ({}^B\mathbf{r}_T, {}^B_T\mathbf{R})$. From the matrix ${}^B_T\mathbf{R}$ the fixed zyx-Euler angles ϕ, θ, and ψ can be derived. The positions of the tool joints ${}^Tr_{t,i}$ in the tool frame and the initial positions of the slider joints ${}^Br_{s_0,i}$ in the base frame for each actuator i are known from the geometry of the construction. The sliders actual positions ${}^Br_{s,i}$ move in z-direction of the base according to the guidance, if the tool changes its position. Therefore, the initial positions ${}^Br_{s,i,x} = {}^Br_{s_0,i,x}$ and ${}^Br_{s,i,y} = {}^Br_{s_0,i,y}$ in the x-direction respectively the y-direction are not alternated. The struts have the lengths l_i.

3 MODELLING

For the model-based control of the robot, the nonlinear inverse dynamic description is required, especially if both platforms can be moved freely in the entire workspace.

3.1 Inverse Kinematics

The inverse kinematics of the robot describes the nonlinear relationship between the relative tool-base pose and the slider positions, which states the operating point of the robot in the instantaneously coincident base frame {B'}. Furthermore, the velocities and accelerations of the sliders are given from the relative motion of the tool and the base in {B'}.

3.1.1 Pose

Since the slider positions are constrained by the six struts $i = 1..6$, the joint x and y distances between slider and strut joints are given by

$$^{B}r_{st,i,x} \;=\; ^{B}r_{s,i,x} - \, ^{B}r_{t,i,x} \tag{1}$$

and

$$^{B}r_{st,i,y} \;=\; ^{B}r_{s,i,y} - \, ^{B}r_{t,i,y}. \tag{2}$$

With the assumption of constant strut lengths l_i the z-component of the joint distance vectors can be calculated as

$$^{B}r_{st,i,z} \;=\; \sqrt{l_i^2 - (^{B}r_{st,i,x})^2 - (^{B}r_{st,i,y})^2} \tag{3}$$

and the slider joint positions yield

$$^{B}r_{s,i,z} \;=\; ^{B}r_{t,i,z} + \, ^{B}r_{st,i,z}. \tag{4}$$

The required shift of the actuator positions with respect of the starting positions $^{B}r_{s_0,i,z}$ is

$$q_i \;=\; ^{B}r_{s,i,z} - \, ^{B}r_{s_0,i,z}, \tag{5}$$

which is measured by local positioning sensors.

3.1.2 Velocity

The generalized tool and base velocities $^{B'}\dot{\mathbf{X}}_T = (^{B'}\mathbf{v}_T, \, ^{B'}\omega_T)$ and $^{B'}\dot{\mathbf{X}}_B = (^{B'}\mathbf{v}_B, \, ^{B'}\omega_B)$ embrace the translational velocities $^{B'}\mathbf{v}_T$ respectively $^{B'}\mathbf{v}_B$ and the angular velocities $^{B'}\omega_T$ and $^{B'}\omega_B$ of both rigid bodies. According to a rigid body motion the tool joint positions and the base-fixed initial slider joint positions can be determined from the generalized velocities. Due to the constant strut lengths, the z-components of the relative slider velocities

$$^{B'}v_{st,i,z} \;=\; -\frac{^{B}r_{st,i,x}}{^{B}r_{st,i,z}} \, ^{B'}v_{st,i,x} - \frac{^{B}r_{st,i,y}}{^{B}r_{st,i,z}} \, ^{B'}v_{st,i,y} \tag{6}$$

result from the constraint movement on a sphere with radius l_i. The relative velocities of the sliders, which can be measured and controlled on the actuator level, is

$$\dot{q}_i \;=\; ^{B'}v_{st,i,z} + \, ^{B'}v_{t,i,z} - \, ^{B'}v'_{s,i,z} \tag{7}$$

with the tool joint velocities $^{B'}v_{t,i,z}$ and the velocities of the initial joint positions $^{B'}v'_{s,i,z}$.

3.1.3 Acceleration

Corresponding to the velocity derivation, the generalized accelerations of the tool and the base are defined by $^{B'}\ddot{\mathbf{X}}_T = (^{B'}\mathbf{a}_T, \, ^{B'}\alpha_T)$ and $^{B'}\ddot{\mathbf{X}}_B = (^{B'}\mathbf{a}_B, \, ^{B'}\alpha_B)$ with the translational accelerations $^{B'}\mathbf{a}_T$ and $^{B'}\mathbf{a}_B$ and the angular accelerations $^{B'}\alpha_T$ and $^{B'}\alpha_B$. The accelerations of the joints are assemblies of three terms: (1) inertial acceleration a', 2) centripetal acceleration a'', and (3) Coriolis acceleration a'''. The inertial terms of the slider joints are

$$^{B'}\mathbf{a}'_{s,i} \;=\; ^{B'}\mathbf{a}_B + \, ^{B'}\alpha_B \times \, ^{B}\mathbf{r}_{s,i}, \tag{8}$$

and the centripetal accelerations are

$$^{B'}\mathbf{a}''_{s,i} \;=\; ^{B'}\omega_B \times \, ^{B'}\omega_B \times \, ^{B}\mathbf{r}_{s,i}. \tag{9}$$

The Coriolis acceleration terms are non-zero in x-direction

$$^{B'}a'''_{s,i,x} \;=\; 2 \cdot ^{B'}\omega_{B,y} \cdot \dot{q}_i \tag{10}$$

and in y-direction

$$^{B'}a'''_{s,i,y} \;=\; -2 \cdot ^{B'}\omega_{B,x} \cdot \dot{q}_i. \tag{11}$$

The z-component $^{B'}a'''_{s,i,z}$ is identical zero.

Summarizing the three terms, the slider joint accelerations in x and y direction result in

$$^{B'}a_{s,i,x} \;=\; ^{B'}a'_{s,i,x} + \, ^{B'}a''_{s,i,x} + \, ^{B'}a'''_{s,i,x} \tag{12}$$

$$^{B'}a_{s,i,y} \;=\; ^{B'}a'_{s,i,y} + \, ^{B'}a''_{s,i,y} + \, ^{B'}a'''_{s,i,y}. \tag{13}$$

The slider acceleration in z-direction is not a simple sum, because the slider motion is constrained by the struts according to the sphere equation

$$\begin{aligned} ^{B'}a_{st,i,z} \;=\; & -\frac{^{B}r_{st,i,x}}{^{B}r_{st,i,z}} \left(^{B'}a_{st,i,x} + ^{B'}a_{nst,i,x} \right) \\ & -\frac{^{B}r_{st,i,y}}{^{B}r_{st,i,z}} \left(^{B'}a_{st,i,y} + ^{B'}a_{nst,i,y} \right) \\ & - \left(^{B'}a_{nst,i,z} \right) \end{aligned} \tag{14}$$

with the normal acceleration

$$^{B'}\mathbf{a}_{nst,i} \;=\; -\left(^{B}\mathbf{r}_{st,i} \times ^{B'}\mathbf{v}_{st,i} \right) \times ^{B'}\mathbf{v}_{st,i} \, / \, \left(^{B}\mathbf{r}_{st,i} \right)^2 \tag{15}$$

Finally, the slider joint accelerations are

$$^{B'}a_{s,i,z} \;=\; ^{B'}a_{st,i,z} + \, ^{B'}a_{t,i,z}. \tag{16}$$

and the acceleration of a slider's center of gravity yields

$$ {}^{B'}a_{cg,i,z} \;=\; {}^{B'}a_{s,i,z} + {}^{B'}\omega_B \times {}^{B'}\omega_B \times {}^{B}\mathbf{r}_{scg,i} \qquad (17) $$

with the position offset ${}^{B}\mathbf{r}_{scg,i}$ between the slider joint position and the slider's centre of gravity.

3.2 Inverse Dynamics

The inverse dynamics of the robot describes the actuator forces needed to move the tool's centre of gravity with a desired velocity and acceleration. Additionally, the actuator forces are derived to stabilize the tool, if the base is disturbed. In this section a brief sketch of the ITDII robot's inverse dynamics is given. A more detailed description can be found in (Wagner et al., 2006).

The rigid body dynamics of a parallel robot can be described generally by the equation of motion

$$ {}^{B'}\Lambda_T = \mathbf{M}(\mathbf{q}) \cdot \ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \cdot \dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}). \qquad (18) $$

with the generalized force ${}^{B'}\Lambda_T$, which is necessary to move the tool platform in a fixed frame {B'}. Within this equation **M** is the generalized mass matrix, **C** is the centripedal and Coriolis component and **G** is the gravity component of the platform. These parameters depend on the operationg point of the robot. The generalized force is converted into actuator forces using the Jacobian matrix of the robot. Additionally, the inertia forces

$$ \mathbf{F}_{s,i} \;=\; m_{s,i} \left[{}^{B'}a_{cg,i,z} \;-\; \left({}^{B}_{W}\mathbf{R}\mathbf{g}\right)_z \right]. \qquad (19) $$

are taken into acount with the slider masses $m_{s,i}$. The gravitiy vector $\mathbf{g} = (0\;0\; -9,81\,m/s^2)^T$ can be transformed from the {W} frame into the {B} frame using the rotation matrix ${}^{B}_{W}\mathbf{R}$. Since the major part of the strut masses is concentrated in the joints and since the link beetween the joints is lightweight, the struts can be separated into two parts, which can be added to the slider and tool components (Honegger, 1999). Correspondingly the strut masses and inertias are not considered in the inverse dynamics explicitly. In a first attempt the friction forces are neglected as well. The latter can easily be extended on the actuator level, if necessary. The non-working reaction forces/torques of the actuators are perpendicular to the actuator motions and, therefore, not considered in the robot dynamics.

For the calculation of all force components the relative velocity and acceleration between the tool and the base as well as the absolute orientation of the robot in the world coordinates must be available. It should be mentioned, that the actuator forces are not simple vector sums of the tool and the base related forces, since the equation of motion is non-linear. However, if the tool is fixed, the ${}^{B'}\Lambda_T$ term vanishes and the remaining forces for the slider mass acceleration yields a quite simple form (19).

4 CONTROL STRUCTURE

The robot control must be suited to adjust the tool against a movable workpiece and to decouple the tool from disturbances at the base. Therefore, a control structure was designed, which combines a servo control on the local axis level with a disturbance feed-forward control in the global Cartesian space (Fig. 3).

For the controller design the adequate choice of co-ordinate systems is essential, since the complete robot can be moved freely in space and the component inertia forces are related to a earth fixed (respectively inertia) reference system. The local stabilization is based on the robot kinematics, which describes the relative pose between tool and base in relation to the actuator positions. Here, the actuator position are not dependent on the tool and base reference. In contrast, the feed-forward control uses the inverse dynamics block, which is definded in {B'}. The sensor signals for the base pose, velocity and acceleration are retrieved in the world frame. Furthermore, the desired tool motion is given in the world frame as well. Therefore, the tool and base motion signal are converted to {B'} co-ordinates before they are applied to the inverse kinematics. This geometric transformation can be done with little effort.

The local control is based on the slider position and velocity sensor signals **q** and $\dot{\mathbf{q}}$. The actuator signals are compared to the desired actuator signals $\mathbf{q_d}$ and $\dot{\mathbf{q}}_\mathbf{d}$. Using the generalized tool mass matrix **M** a PD-controller generates the force signals $\mathbf{F_{fb}}$ needed to stabilize the tool. The reference signals $\mathbf{q_d}$ and $\dot{\mathbf{q}}_\mathbf{d}$ result from the desired tool signals ${}^{W}\mathbf{X}_{Td}$, and ${}^{W}\dot{\mathbf{X}}_{Td}$ and from the actual base coordinates ${}^{W}\mathbf{X}_B$, and ${}^{W}\dot{\mathbf{X}}_B$ both in Cartesian world coordinates, using the inverse kinematics of the parallel robot.

The feed-forward control block consists of two inputs, the tool reference signals ${}^{W}\mathbf{X}_{Td}$, ${}^{W}\dot{\mathbf{X}}_{Td}$, and ${}^{W}\ddot{\mathbf{X}}_{Td}$ and the actual base signals ${}^{W}\mathbf{X}_B$, ${}^{W}\dot{\mathbf{X}}_B$, and ${}^{W}\ddot{\mathbf{X}}_B$. Using the inverse dynamic model the feed-forward forces $\mathbf{F_{ff}}$ are calculated which are required to move the tool as desired, while the disturbances from the base are canceled. If the inverse dynamic model and the sensor/actuator signals are assumed to be perfect, the local control error will be zero. However, the inverse model is not really complete due to the neglected strut masses, actuator friction and un-

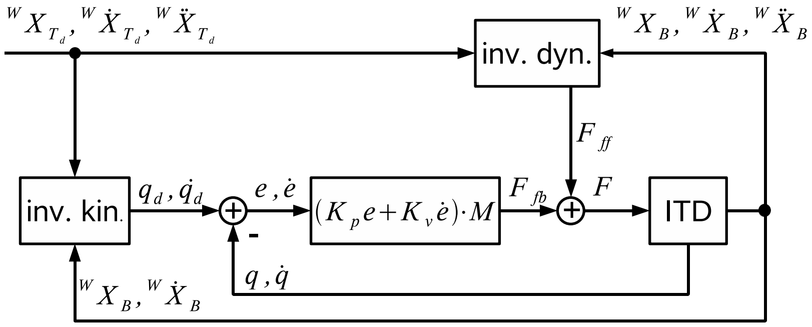

${}^{W}X_{T_d}, {}^{W}\dot{X}_{T_d}, {}^{W}\ddot{X}_{T_d}$
inv. dyn.
${}^{W}X_B, {}^{W}\dot{X}_B, {}^{W}\ddot{X}_B$
F_{ff}
inv. kin.
$q_d, \dot{q}_d$
$e, \dot{e}$
$(K_p e + K_v \dot{e}) \cdot M$
F_{fb}
F
ITD
-
$q, \dot{q}$
${}^{W}X_B, {}^{W}\dot{X}_B$

Figure 3: Control structure; local servo loop and global disturbance feed forward control.

measured disturbances. These model uncertainties are compensated by the local pose control. The presented structure is not a classical cascaded structure, because the base coordinates are not influenced immediately by the control action. The base motion results from the disturbance forces of the human operator. That means that no feed-back from the actuators to the base motion is assumed due to the large base mass and that the sensing of the base motion serves for the referencing of coordinates only.

In the realized system it is necessary to measure the pose, the velocity and the acceleration of the base. The tool coordinates are calculated from the desired tool trajectory. To take care of force feed-back during the processing of the workpiece, additional sensors measuring the tool motion or forces could be implemented in a feed-back control. This may make sense in special cases. However, backlash effects in the joints could lead to a destabilization of the controlled system. Therefore, and for the sake of simplicity we are content with the local stabilization.

As shown in literature (Riebe and Ulbrich, 2003), a friction compensation is essential in the real application, which can be added on the local axis level. To separate the kinetic effects from the friction effects, such a compensation is neglected in this paper. The control structure has been implemented in the simulation environment Matlab/Simulink using a fixed sampling time of 1 ms. As a consequence the minimum delay in the control loop is 1 ms as well. The forward dynamics of the robot ITD used in the closed loop simulations has been modelled with the SimMechanics toolbox.

5 SIMULATION

The simulations presented in this section mainly support the description of the disturbance decoupling ability of the system. Additionally, the overall trajectory tracking quality is described at the end using a standard circle test (ballbar-test).

5.1 Tool Stabilization

The first simulations describe the system disturbance response in the frequency domain. In the first simulations the control parameters are configured for a critically damped PD control loop with a system frequency $\Omega = 60Hz$. Figure 4(a) shows the tool motion after application of a 12 Hz, 1 mm sinusoidal signal at the base. The straight line represents the stimulus in the x-direction, which can be referenced to the left hand scale. The amplitude of the tool position response using a PD control (dotted line) and using a PD control with additional feed forward disturbance

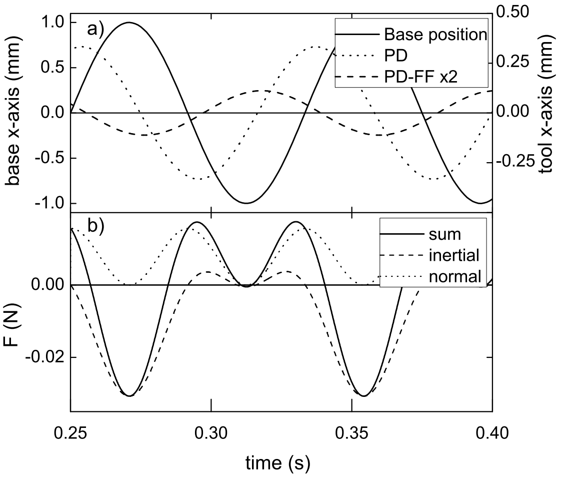


Figure 4: Response on a 12 Hz, 1 mm sinusoidal translational base disturbance in x-axis; (a) base position stimulus (straight line, left-hand scale), tool position response (PD control, dotted line, right-hand scale), tool position response multiplied by a factor of two (PD control with disturbance feed-forward, dotted line, right-hand scale); (b) Force needed for feed-forward compensation.

compensation (dashed line) can be extracted from the right hand scale. For a better readability the later is multiplied by a factor of two.

As a result the PD controlled tool is damped against the base disturbance with a transfer factor of 0.3 (-10dB) due to its limited bandwidth. Adding the feed-forward control the transfer factor decreases to 0.06 (-24dB).

The actuator force needed to cancel the disturbances is plotted in Fig. 4(b) for the first actuator. The resulting force F (straight line) is the total of all force components. The inertial component (dashed line) results from the tool and the slider acceleration (8) and the normal component (dotted line) is a consequence of the strut rotation velocity (6). No Coriolis or centripedal forces are generated for a translational tool movement. While the tool-pose reaction seams to be more or less a sinusoidal function, higher harmonic distortions are noticeable in the inertial component.

Now, the sinusoidal stimulus is shifted to 1 Hz and 20 mm amplitude (Fig. 5) compared to Fig. 4. Here, the feed-forward related position response is multiplied by a factor of 50. As a result of the lower frequency the PD-loop has an increased damping effect with a transfer factor of $6 \cdot 10^{-3}$ (-45dB) respectively $6.5 \cdot 10^{-5}$ (-84dB) with feed forward control (Fig. 5). The pose response signal in the large signal case is not a sinusoidal signal anymore due to the non-linear coupling between tool and base.

A sinusoidal stimulation of the base orientation around the x-axis and its response is shown in Fig. 6(a). The forces needed to stabilize the tool are plotted in Fig. 6(b). Here, the Coriolis (large dotted line) and centripedal (large dashed line) forces are in the range of the inertia forces. Unlike an intuitive estimation the Coriolis and centripedal components cannot be neglected in this special robot application.

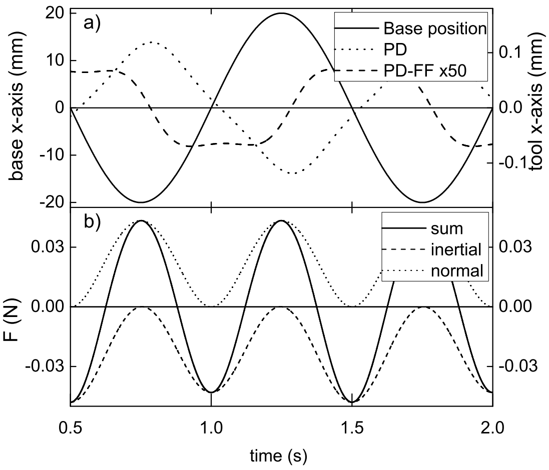


Figure 5: Response on a 1 Hz, 20 mm sinusoidal translational base disturbance in x-axis; (a) base position stimulus (straight line, left-hand scale), tool position response (PD control, dotted line, right-hand scale), tool position response multiplied by a factor of 50 (PD control with disturbance feed-forward, dotted line, right-hand scale); (b) Force needed for feed-forward compensation.

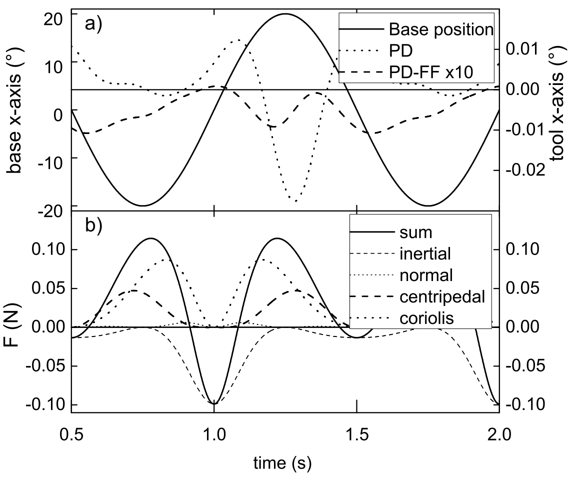


Figure 6: Response on a 1 Hz, 20^o sinusoidal rotational base disturbance around the x-axis; (a) base orientation stimulus (straight line, left-hand scale), tool orientation response (PD control, dotted line, right-hand scale), tool orientation response multiplied by a factor 10 (PD control with disturbance feed-forward, dotted line, right-hand scale); (b) Force needed for feed-forward compensation.

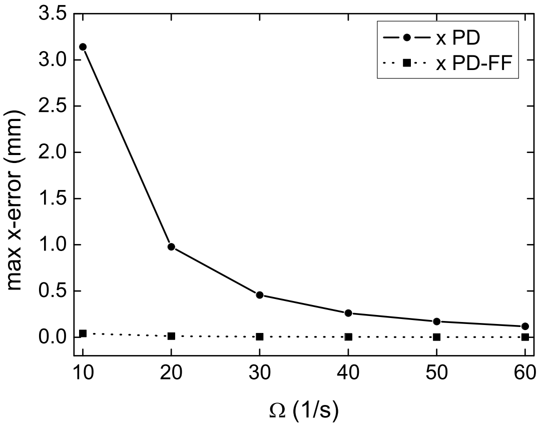


Figure 7: x- axis response on a 1 Hz, 20 mm sinusoidal translational base disturbance in dependence on the servo loop frequency.

The decoupling behaviour from the base motion strongly depends on the choice of the parameter Ω and the delay in the feed-back loop, which is at least one sampling period. This is shown in Fig.7 and Fig.8

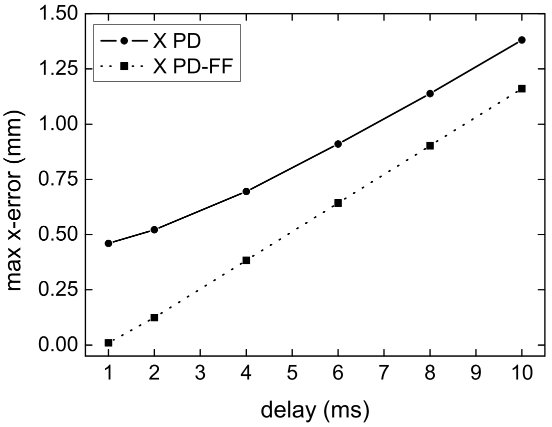


Figure 8: x- axis response on a 1 Hz, 20 mm sinusoidal translational base disturbance in dependence on the sensor delay.

for a base disturbance of 1 Hz and 1 mm.

The pose error amplitude and phase decrease with increasing control loop frequency. With disturbance feed forward control the error is generally smaller, however the quality strongly depends on the sensor delay. To estimate the limit time lack that can be allowed the pose error is plotted against the sensing delay. For instance, the delay should not exceed 3 ms, if a pose accuracy of 0.3 mm is required. (Certainly, additional error sources must be considered).

5.2 Tracking Control

To assess the possibilities of the disturbance feed forward control the standard circle (ballbar) test was simulated assuming an additional white Gaussian sensor noise with a translational standard deviation of $\sigma_{accel} = 0.17mm/s^2$ and a rotational standard deviation of $\sigma_{rot} = 0.17rad/s$ for all axes. Figure 9 shows the reference trajectory, which has to be followed with a velocity of 0.1 m/s, and the actual trajectories using a PD control and using a PD control with disturbance feed forward. The deviation from the reference trajectory is exaggerated by a factor of five. The mean pose error are 0.35 mm and 0.10 mm with respectively without feed forward control. The standard deviations are 0.051 mm and 0.052 mm. Also for the reference action the feed forward control diminishes the pose error remarkably. Because the signal noise from the base sensors is injected into the inverse kinematics block as well as into the inverse dynamics block, it influences the PD control loop once and the PD control with feed forward control twice. However, the persistent pose noise is not much larger with a feed forward control compared to the pure PD control approach.

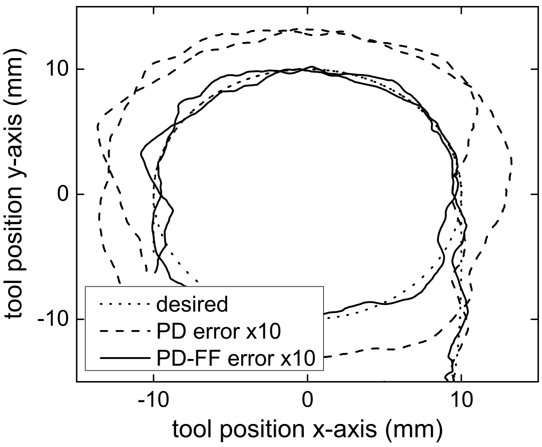


Figure 9: Standard circle test with white Gaussian sensor noise; $\sigma_{accel} = 0.17mm/s^2$ translational standard deviation, $\sigma_{rot} = 0.17rad/s$ rotational standard deviation, $v = 0.1m/s$ feed speed.

6 DISCUSSION

The simulations show, that if a complete model of the parallel robot and adequate sensor signals are available, a feed forward control can lead to a decoupling from base disturbances with high damping factors. For the quality of the decoupling the sensor noise and the sensor delay are essential. The simulated damping value of $6.5 \cdot 10^{-5}$ in the 1 Hz case (Fig. 5) must be interpreted with care, because the quality of the decoupling is influenced by numerical errors in this range. Furthermore, model uncertainties and sensor errors do not allow such a high precision in real-world applications. Within the control approach the base pose, velocity and acceleration must be measured or observed accurately in the inertia system. This is a serious problem, because not only sensor latency and noise must be considered but also misalignment, bias, drift, etc..

For the simulations an ideal model is assumed without geometric or parametric errors and without considering the friction in the actuators and joints. An enhancement of the dynamic model can be done with acceptable effort, e.g. the modelling of friction on the actuator level. However, the quantity of friction and its influence on the dynamics depends on the special mechanical implementation and many parameters. Therefore a possible model extension must be considered if the robot is realized. In contrast to the intuitive impression that for the size, mass and dynamics of a hand-held robot the Coriolis and centripedal forces do not play any role, the simulations show that these force components make a remarkable contribution to the total force in the actuators. Thus, a simplification of the model by neglecting the velocity related force components is not suitable. Finally, the simulations show in which range of precision a feed forward control does improve the decoupling behaviour and in which range the feed forward control can be neglected.

7 CONCLUSION

A model-based approach is presented to control the tool pose of a handheld robot and to decouple the tool from base disturbances. An adequate definition of coordinate frames for the dynamics modelling and the controller design reduces the effort for the implementation. The feasibility of a feed-back control on the local axis level in combination with a disturbance feed forward control on the robot level in world coordinates is shown. The local control is able to stabilize the robot and to avoid huge errors due to model uncertainties and disturbances. The feed forward control ensures the free movement of the robot in space, while measured disturbances can be compensated for. Furthermore, the usage of a non-linear inverse dynamics model enables the precise disturbance feed-forward control under different operational conditions. For the feed-forward control sensor error and delay are crucial.

REFERENCES

Chen, Y. and McInroy, J. E. (2004). Decoupled control of flexure-jointed hexapods using estimated joint-space mass-inertia matrix. *IEEE Transactions on Control Systems Technology*, 12.

Honegger, M. (1999). *Konzept einer Steuerung mit adaptiver nichtlinearer Regelung für einen Parallelmanipulator*. Dissertation, ETH Zurich, Switzerland, http://www.e-collection.ethbib.ethz.ch.

Honegger, M., Codourey, A., and Burdet, E. (1997). Adaptive control of the hexaglide, a 6 dof parallel manipulator. *IEEE International Conference on Robotics and Automation, Albuquerque, USA*.

Huynh, P. (2001). Kinematic performance comparison of linar type parallel mechanisms, application to the design and control of a hexaslide. *5th International conference on mechatronics technology (ICMT2001), Singapore*.

Koekebakker, S., Teerhuis, P., and v.d. Weiden, A. (1998). Multiple level control of a hydraulically driven flight simulator motion system. *CESA Conference, Hammammet*.

Merlet, J. (2000). *Parallel robots*. Kluwer Academic Publisher, Dordrecht, Netherlands.

Pott, P., Wagner, A., Köpfle, A., Badreddin, E., Männer, R., Weiser, P., Scharf, H.-P., and Schwarz, M. (2004). A handheld surgical manipulator: Itd - design and first results. *CARS 2004 Computer Assisted Radiology and Surgery, Chicago, USA*.

Riebe, S. and Ulbrich, H. (2003). Modelling and online computation of the dynamics of a parallel kinematic with six degrees-of-freedom. *Archive of Applied Mechanics*, 72:817–829.

Stewart, D. (1965-1966). A platform with six degrees of freedom. *Proceedings of the Institute of Mechanical Engineering*, 180:371–386.

Takanokura, M. and Sakamoto, K. (2001). Physiological tremor of the upper limb segments. *Eur. J. Appl. Physiol.*, 85:214–225.

Tönshoff, H., Grendel, H., and Grotjahn, M. (2002). Modelling and control of a linear direct driven hexapod. *Proceedings of the 3rd Chemnitz Parallel Kinematics Seminar PKS 2002, 2002 Parallel Kinematic Machines Int.Conf.*

Wagner, A., Pott, P., Schwarz, M., Scharf, H.-P., Weiser, P., Köpfle, A., Männer, R., and Badreddin, E. (2004). Control of a handheld robot for orthopedic surgery. *3rd IFAC Symposium on Mechatronic Systems, September 6-8, Sydney, Australia*, page 499.

Wagner, A., Pott, P., Schwarz, M., Scharf, H.-P., Weiser, P., Köpfle, A., Männer, R., and Badreddin, E. (2006). Efficient inverse dynamics of a parallel robot with two movable platforms. *4rd IFAC Symposium on Mechatronic Systems, Heidelberg, Germany*.

THE TELE-ECHOGRAPHY MEDICAL ROBOT OTELO2

Teleoperated with a Multi Level Architecture using Trinomial Protocol

Gwenaël Charron[(1)], Aïcha Fonte[(1)], Pierre Vieyres[(1)]
Philippe Fraisse[(2)], Lama Al Bassit[(1)] and Cyril Novales[(1)]
[(1)]Laboratoire Vision & Robotique, University of Orleans, Bourges, France
{firstname.name}@bourges.univ-orleans.fr
[(2)]Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier, University of Montpellier II
Montpellier, France
philippe.fraisse@lirmm.fr

Keywords: Teleoperated system, Telerobotics, OTELO medical robot, Tele-echography, Multi level archecture.

Abstract: This paper presents a novel architecture applied to a mobile teleoperated medical robotic system: OTELO2 (MObile Tele-Echography using an Ultra-Light RObot); OTELO2 performs a tele-echography at a distance for the benefit of medically isolated sites. First, this paper presents an overview of the OTELO2 teleoperated system. Then, it describes the modular control architecture used and the integration of the teleoperated layer on this multi level architecture. Finally, it presents the communication links used to control this system, as well as some experimental results.

1 INTRODUCTION

Telerobotics and teleoperation have currently a very important role to play in the medical field especially in non invasive medical application (i.e. tele-echography) needed by the patients living in isolated sites with reduced medical facilities. The aim of the OTELO2 system is to provide people with the best medical examination conditions and thus to have the best diagnostic as possible.

Based on the concept of a mechanical probe holder, OTELO2 is a teleoperated robotic manipulator arm. Teleoperated systems are exposed to possible instability due to the transmission delay of the communication link and to the need of remote safety and maintenance of the robot. The goal of the proposed combined approach, i.e. to use a specific protocol to reduce data loss and to implement a modular architecture to enhance the tele-echography system, is to favor the overall remote medical act for the benefit of the patients.

To control teleoperated system, lots of architectures have been proposed. The "Subsumption Architecture" (Brooks, 1986) is composed of parallel different levels which process information supplied by the sensors in order to determine the control to be sent to actuators. The LAAS architecture (Alami, 1998) is made up of three levels: decisional, executive and functional. Its goal is to homogenize the whole mobile robotics developments and to be able to re-use already designed modules. The AuRA architecture (Arkin, 1998) is made up of two parts (reactive and deliberate), each using distinct method to solve problems. The reactive part is based on sensors and the deliberate part uses artificial intelligence method contains a mission planner, a spatial reasoner and plan sequencer. The OTELO2 architecture relies on the concept of levels initially developed by Brooks and which appear in architectures proposed by AuRA or LAAS. The originality of this architecture is to decompose the control in multi-levels which allows to decompose in clear way the different functions realized and also to decompose each level in several blocks which allow to retail and to separate the connections with the sensors and the actuators.

A description of the teleoperated robotic system, including the "expert" and the "patient" station, is given in the first section. The second section presents the architecture developed to control the robot. The next section presents the communication links between the two stations composing the system and the protocol used to control the robot. The last section presents some experimental result obtained during a teleoperation between Bourges (France) and Montpellier (France).

2 THE OTELO2 SYSTEM

OTELO2 is a teleoperated robotic prototype system composed of an “expert” station and a “patient” station. A communication network (e.g. terrestrial or satellite) allows data transmission (i.e. ultrasound images, robot controls, haptic feedback and ambient images) between the two stations (Figure 1).

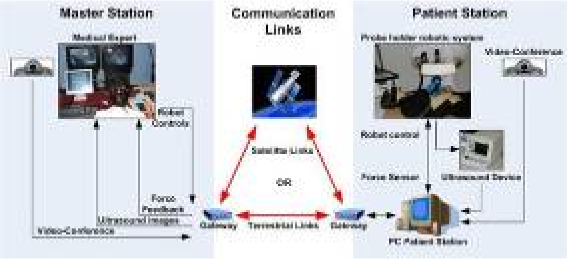


Figure 1: The OTELO2 teleoperation chain.

2.1 The “Patient” Station

The “patient” station is located near the patient at a medically isolated site or in a secondary hospital. It includes a portable echograph device allowing ultrasound frames acquisition and the hardware system for the control of the probe holder robot.

The results of a previous study (Al Bassit, 2003), on the medical gesture performed by a specialist during an abdominal echography, gave the mechanical constraints and characteristics of the robot work space with respect to the tele-echography medical application. The probe must have a spherical displacement around a contact point of the probe with the patient’s skin. Displacement amplitudes (Figure 2) are characterized by: a maximal probe inclination of 60° with respect to the normal of the skin plan. (larger inclination is considered useless by the medical expert); a minimal inclination of 35° is necessary, as well as a full rotation of 360° around the probe symmetric axis are needed to fulfill users’ requirements. Finally, a translation along the probe axis is necessary to obtain quality ultrasound images and maintain a continuous contact between the probe and the patient’s skin. For safety functioning, patient comfort and force control, this displacement amplitude is limited to the interval [-30mm, 10mm]. Hence, the maximal admissible force of the probe on the skin does not exceed 20 Newton.

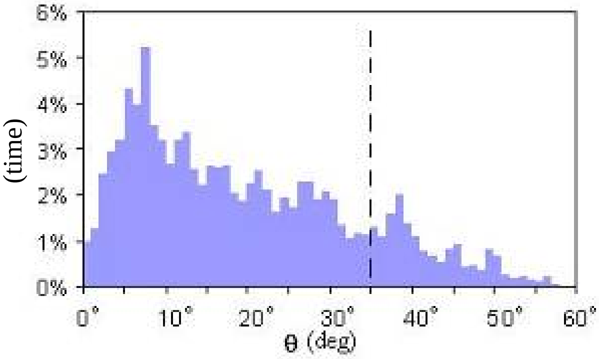


Figure 2: Histogram of inclination angle θ of the probe axis during an abdominal ultrasound examination.

The OTELO2 robot prototype was developed by the Laboratory of Vision and Robotic (LVR) in collaboration with European project partners in order to answer the previously mentioned criteria. OTELO2 is a serial six DOF (Degree Of Freedom) probe holder system; it includes a positioning module (with two prismatic articulations, P1 and P2, with perpendicular axis), a spherical module with distant rotation center (with three revolute pairs R1, R2 and R3, the R1-R2 and R2-R3 angles are equal to α) and a translation along the probe axis (P3, the R3-P3 angle is equal to β) allowing, for a given orientation, to modify the probe/skin contact force (Figure 3).

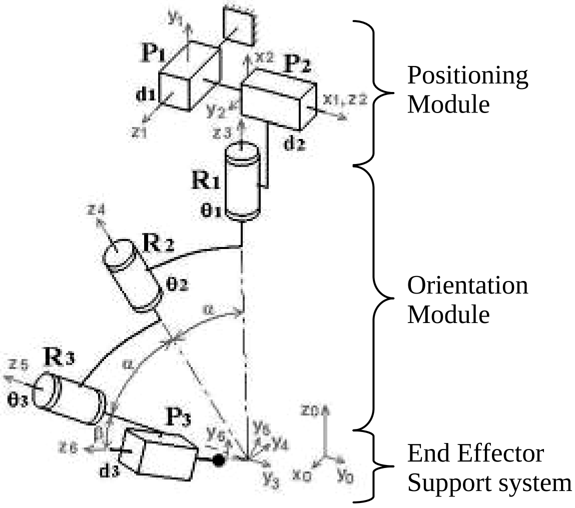


Figure 3: Kinematics diagram of OTELO2 robot prototype.

The α and β angles, respectively 27,5° and 10°, allow a probe maximal inclination of 65° which complies with the medical requirements. The positioning module allows a probe displacement with maximal amplitude of ±25mm for each axis and offers two DOF to search for an organ. The translation amplitude along the probe axis is about 40mm. This axis is coupled with a force sensor giving the force applied by the ultrasonic probe on the patient's skin and enabling its control. The force is transmitted back to the "expert" station in order to ensure the most realistic examination conditions for the medical expert. Finally, depending on the examination type (e.g. ObGyn, Abdominal), various types of manufactured probe can be attached to the end effector of the support system.

The end effector of the remote robot moves the ultrasound probe in order to reproduce the expert gestures which are being analyzed by a dedicated input device (Figure 4). Images coming from the ultrasound system are compressed and sent to the "expert" station, using the H263 protocol, and analyzed by the specialist.

2.2 The "Expert" Station

The "expert" station is located in a main hospital center and is operated by a medical expert. Based on the received ultrasound images, the expert uses a pseudo haptic input device (Figure 4) to control the positions and orientations of the remote ultrasound probe. A videoconferencing system between the two stations allows the medical expert to communicate with the patient, to give instruction to the assistant holding the robot, and to check the good positioning of the robot on the patient's body.

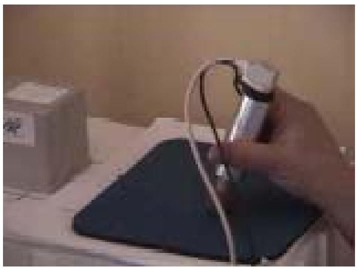


Figure 4: The pseudo haptic input device used to control the orientations and positions of the remote robot.

To control of the teleoperated echography robot was supervised under a novel multi layered and modular architecture. This hardware and software structure was added with specific communication protocols used to control the robot on Internet network. The following section presents the proposed control architecture and its layout. It is followed by the description of the protocol used for data transmission for the robot control.

3 THE OTELO2 SYSTEM CONTROL ARCHITECTURE

For the teleoperated robot and in order to integrate teleoperation layer, it was decided to set up a layered architecture (Novales, 2006). It is a multi level architecture where each level corresponds to a decision/perception loop.

In this section, we present the control architecture of the OTELO2 robot, and the global architecture of the OTELO2 system is described with the two MMI (Man Machine Interface) developed to control the system.

3.1 The "Patient" Architecture

The control architecture of the OTELO2 robot prototype is a three level architecture partitioned in two parts, namely the "Perception" and the "Decision" parts. Each one of these levels correspond to either a software layer or a hardware layer (Figure 5).

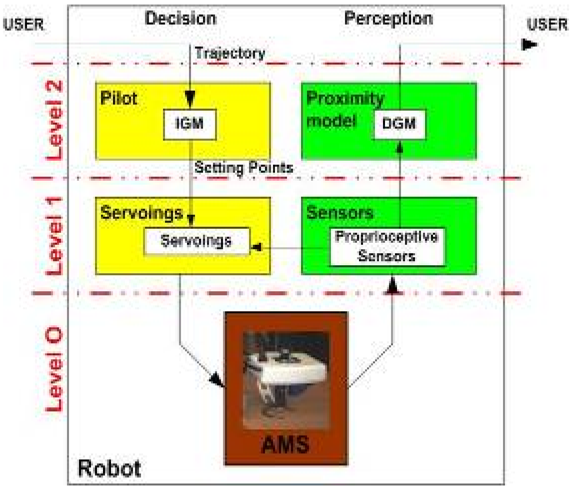


Figure 5: The layered control architecture of OTELO2 robot.

Level 0 represents the Articulated Mechanical System (AMS); it contains the input/output direct interface between the physic world and the robot. This level receives physical data necessary to its actuators and sends information to the sensors at level 1.

Level 1 of the decision part corresponds to the servoings level; it determines the physics data, to be addressed to level 0, from the setting points imposed directly by the upper level.

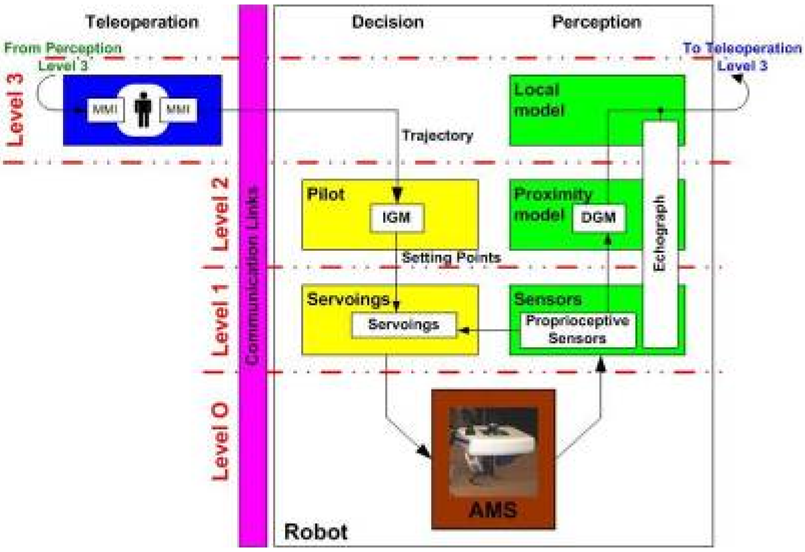


Figure 6: The global architecture of OTELO2 teleoperated system.

The level 1 perception part receives the information supplied by the sensors, and it translates this information to the upper level and to the servoings module. This level ensures the articular servoings with six modules in each part, corresponding to the six axes and associated sensors of the robot.

Finally, Level 2 decision part corresponds to the pilot level; it generates the articular setting points to the lower level from a trajectory (position and orientation of the probe) supplied by the user. The pilot block uses the IGM (Inverse Geometric Model) to generate the setting points taking into account the physical constraints of the system. The level 2 perception part presents a proximity model using a DGM (Direct Geometric Model) to transmit the robot current positions and orientations to the user. We can note, for our application, that there is not a direct feedback loop on this second level. The control loop is accounted for through the distant human teleoperator.

Perception and Decision parts constitute the so-called autonomous part of the robot. A third part, called the teleoperation, is added to the two previous one in the framework of a teleoperated system.

3.2 The Global Architecture

The global architecture includes the Perception, Decision and Teleoperation parts. Each level of the teleoperation part receives the data stemming off the level corresponding of the perception part and can a by-pass the corresponding level of the decision part in order to generate the controls for the lower level.

In the OTELO2 system global architecture (Figure 6), the teleoperation level is located at level 3; it corresponds to the navigation level. This part generates the trajectories which are executed by the robot and are sent to the pilot of the level 2 decision part. Moreover, the echograph device delivers information of a high level (ultrasound images) from its own sensors. Thus, this teleoperation level receives information from the level 3 perception part including the robot positions and orientations, and the ultrasound images coming from the ultrasound probe.

This global architecture offers the possibility of lower control level required for remote maintenance and testing of the teleoperated robot (Figure 7).

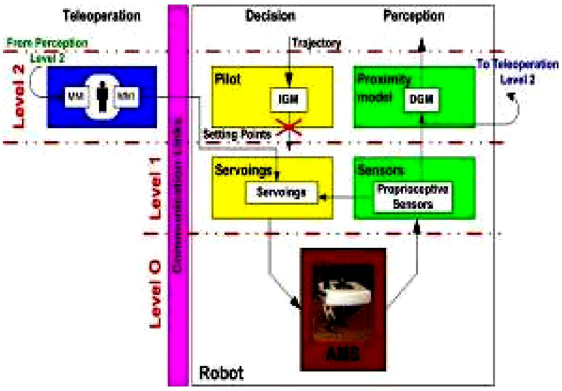


Figure 7: The level 2 teleoperation architecture.

This teleoperation level can control the robot directly, sending articulars setting points. Hence, the user is able to directly and remotely control the robot actuators and is able to detect which of the actuators has a malfunction.

These two teleoperation levels are associated with two MMI allowing an efficient and flexible utilisation of the remote robot.

3.3 Man Machine Interface

With the intention to support the medical expert and in order to ensure the best possible diagnostic, two MMI have been developed for the "expert" station.

The first MMI is a graphical interface provided to the medical expert to visualize the ultrasound images and to choose the appropriate teleoperation level. According to the teleoperation type, the medical expert can control the robot in two different ways. When the expert controls the robot with the high teleoperation level, he/she uses the pseudo haptic input device (second MMI). When the expert controls the robot at a lower teleoperation level, he/she uses a visual interface (Figure 8) to control each individual actuator.

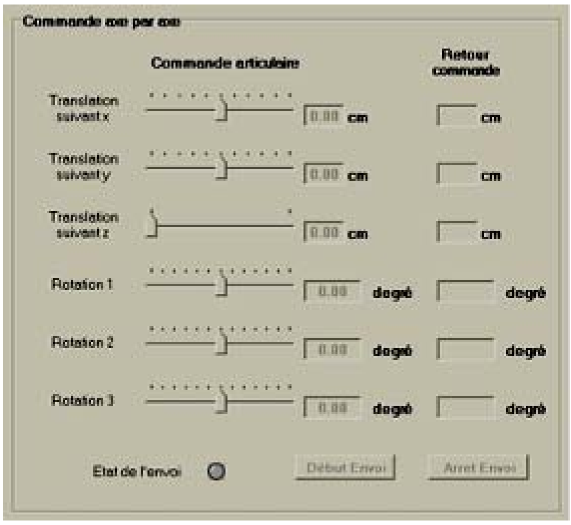


Figure 8: Level 2 teleoperation control interface.

The second MMI is a pseudo haptic interface that resembles an ultrasonic probe. The expert uses the pseudo haptic input device equipped with a six DOF localization magnetic sensor giving positions and orientations.

The pseudo haptic input device (Poisson, 2003) holds a constant stiffness coefficient which provides the medical expert with a rendering of the patients' body compliance and the force applied by the probe on the patients' body. The Figure 9 shows the design of the pseudo haptic input device prototype; it includes a force sensor to measure the force applied by the medical expert in accordance with principal axis of the probe.

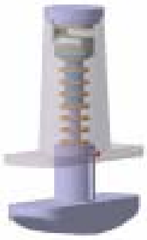


Figure 9: Design of the pseudo haptic input device prototype.

4 THE COMMUNICATION LINK AND DATA TRANSMISSION

The communication between the two stations can be carried out using different communication networks such as satellite, ISDN lines (Integrated Services Digital Network) or the Internet. To perform the robotic tele-echography, three communications protocols are used to transmit all data between the two stations (Figure 10).

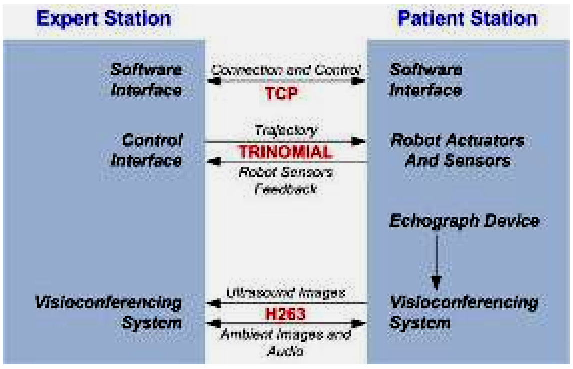


Figure 10: Data Flow representation between the expert and the patient station and selected protocol.

The TCP (Transmission Control Protocol) is an oriented connection protocol. It is used to establish the connection between the two stations, and allows a continuous control of the connection state between the two stations.

To transfer the robot controls, a reliable communication protocol with small transmission delays is needed. Two protocols were firstly considered: the TCP and UDP protocols. TCP ensures reliability in the exchange but can block the communication in case of data loss. UDP protocol (User Datagram Protocol) due to its simplicity generates small transmission delays. However it cannot adapt its flow to the network bandwidth and cannot confirm that data have arrived to the distant site. It was then decided to use a compromise between these two protocols: the trinomial (Xiaoping Liu, 2003). It allows the network not to remain blocked in case of data loss as there is no reemission of the lost data. However, contrary to UDP, trinomial takes into account the transmission delay (i.e. the received data acknowledgement) which allows a modulation of the flow and thus a limitation of the network saturation. In our case, the "expert" station sends the trajectories to the "patient" station using this protocol, and it receives the sensors feedback through the reception of the data acknowledgement.

Finally, a connection is established in order to transmit the ambient images or the ultrasound images to the "expert" station via the videoconferencing system. A bandwidth of 256-384 kbps is required depending on the quality of ultrasonic device to offer the best image quality to the "expert" station.

5 RESULTS

To validate the control architecture and to test the efficiency of our control transmission protocol, a set of tests was performed during a teleoperation between the LIRMM in Montpellier and the LVR in Bourges using the Ethernet RENATER-4 public network; this network provides a flow of about 30Mbit/s.

These results show Round-Trip Time (RTT) between Montpellier and Bourges (Figure 11 (a)), which corresponds to the delays measured between the data transmission and the acknowledgement reception of this data. The RTT varies between 7 and 11 ms proving the efficiency of the chosen data transmission protocol.

Moreover, these results show delays obtained between sending out the control data and the robot position feedback (Figure 11 (b) and (c)) (which include transmission and the time of the servoings). We can see that the system needs approximately 20 ms to reach the desired position.

These results are quite satisfactory and allow us to perform in real time examination in very good conditions without disturbing the rendering of the distant environment to the expert (Arbeille, 2004).

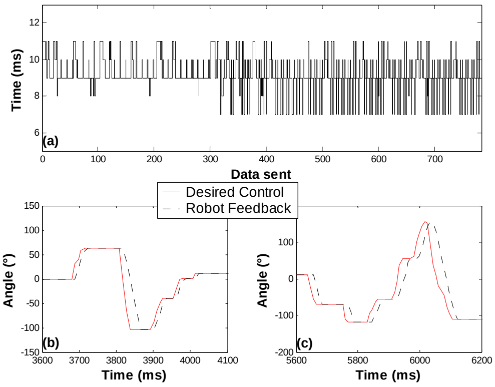


Figure 11: Result during test between Montpellier and Bourges (a) represents the data Round Trip Time, (b) and (c) represent respectively the second and the third axis servoing time delay.

6 PERSPECTIVES

Some improvements (transmission and architectural) have to be considered to provide the medical expert with better examination conditions thus ensuring the best diagnostic as possible.

The communication link (Internet, ISDN, satellite…) used to emit data from the expert station induces transmission delays that can provoke aperiodic data reception and even destabilize the closed loop global system. This can disturb the medical expert medical act when this delay varies too strongly. To avoid that, it is possible to use a FIFO regulator type (Lelevé, 2000) to synchronise the data reception, and thus to provide the expert with a more steady date flow and to adapt more easily his control of the distant robot to the transmission delays.

It is also possible to add an autonomous mode coupled with a level 4 of the teleoperation part, with the intention to realize a full echography of an organ allowing a 3D reconstruction. Thus, the medical expert would select the organ to be investigated and the robot would fallow all trajectories needed to supply the 3D reconstruction wanted by the medical expert. The virtual diagnosis by the expert is made from the 3D reconstruction.

7 CONCLUSION

From a mechanical view point, the OTELO2 prototype robot (Figure 12) corresponds to the criteria imposed by the medical gesture study and experts' requirement; it thus ensures identical examination conditions than to the standard ultrasound one. The modular architecture developed to control it permits easy insertion of new control modules whenever upgrade of the control architecture is needed. Finally, the communication protocol used to control the robot allows small transmission delays and offers moreover adaptability to the network condition.

The experimental results collected during the teleoperation between Montpellier and Bourges show the viability of the tele-echograph system and provided good clinical results.

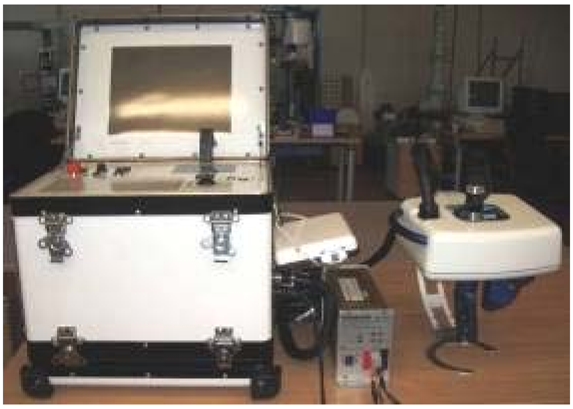


Figure 12: The OTELO2 prototype system.

REFERENCES

Alami, R., Chatila, R., Fleury, S., Ghallab M., and Ingrand F., 1998. *An architecture for autonomy*. The International Journal of Robotics Research, Special Issue on Integrated Architectures for Robot Control and Programming, vol. 17, no 4, pp. 315-337.

Al Bassit, L., Poisson, G., Vieyres, P., 2003. *Kinematics of a Dedicated 6DOF Robot for Tele-Echography,* Proceedings of the 11th International Conference on Advanced Robotics, ICAR 2003, pp. 906-910, Portugal.

Arbeille, Ph., Ayoub, J., Vieyres, P., Porcher, M., Boulay, J., Moreau, V., Poisson, G., 2004. *Echographic diagnostic on a patient far from the hospital using a teleoperated robotic arm and the satellite communications*. Conference on Tele-Health and Satellites, 8-9 July. Rabat, Marocco.

Arkin, R.C, 1998. *Behavior-based Robotics*. MIT Press.

Brooks, R.A., 1986. *A robust layered control system for a mobile robot*. IEEE Journal of Robotics and Automation, vol. 2, n°. 1, pp. 14–23.

Camarinha-Matos, L. M., Castolo, O., Vieira, W., 2002. *A mobile agents approach to virtual laboratories and remote supervision.* Journal of Intelligent and Robotic Systems, no 35, pp. 1-22.

Lelevé, A., 2000. *Contribution à la téléopération de robots en présence de délais de transmission variables*. PhD thesis from Montpellier II University.

Mouriox, G., Novales, C., Smith-Guérin, N., Vieyres, P., Poisson, G., 2005. *A hands free haptic device for tele-echography*. 6TH International Workshop on Research and Education Mechatronics, Annecy, France.

Novales, C., Mourioux, G., Poisson, G., 2006. *A multi-level architecture controlling robots from autonomy to teleoperation*. First National Workshop on Control Architectures of Robots. April 6, 7 2006. Montpellier, France.

Poisson, G., Vieyres, P., Courreges, F., 2003. *Sonde fictive échographique* European patent n°03290168.8.

Xiaoping Liu, P., Max. Q.-H. Meng and Simon. X. Yang, 2003. *Data Communications for Internet Robots*. Autonous Robot Volume 15, pages 213 to 223. Novembre 2003. Kluwer Academic Publishers.

RPQ: ROBOTIC PROXIMITY QUERIES
Development and Applications

Albert Hernansanz, Xavier Giralt
Research Group On Intelligent Robotics and Systems, Technical University of Catalonia, 08028 Barcelona, Spain
albert.hernansanz@upc.edu, xavier.giralt@upc.edu

Alberto Rodriguez
Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA
albertor@cmu.edu

Josep Amat
Institute of Robotics and Industrial Informatics, Technical University of Catalonia, 08028 Barcelona, Spain
josep.amat@upc.edu

Keywords: Collision detection, Proximity queries, Surgical application.

Abstract: This paper presents a robotic proximity query package (RPQ) as an optimization of the general collision library PQP (Proximity Query Package) for the detection of collisions and distance computation between open kinematic chains such as robotic arms. The performance of the optimizations to non specific collision query packages are explained and evaluated. Finally, a robotic assisted surgical application is presented which has been used as a test bed for the proximity package.

1 INTRODUCTION

One of the most important problems to solve in robotics is the collision avoidance between a robot and its environment. A robot should perceive the risk and have a reactive behavior before an imminent collision occurs. Path planning is a hard computational problem, so having a fast tool to calculate collisions is a key factor to decrease the necessary time to generate safety trajectories. In applications where no path planning exists, for instance in manual guidance or teleoperation, a real time collision detector is needed so as to avoid collisions and to be able to interact with the environment, for example sliding over a surface.

The knowledge of minimum distances between robots or objects that share a workspace enables robots to behave in a predictive way. In the human-robot interaction field, virtual fixtures can be used both to prevent collisions and help the human operator by increasing his performance (Stanisic et al., 1996). In this kind of applications minimum distance and collision detection must be known in real time.

A new library: Robotic Proximity Queries (RPQ) package (Giralt and Hernansanz, 2006) has been developed to deal with these requirements, using PQP (UNC, 1999) as the a proximity query engine. The original package has been used to optimize the queries when working with open kinematic chains, like robots. These optimizations have been done with the aim of improving the time performance of the generic package and simplifying its use in robotic environments. A system composed of two robots has been used as a test bed to show the performance of the RPQ library.

Finally a robotic assited surgical application that benefits from RPQ performance is presented. The application consists of the execution of an assisted cut of a rigid tissue. The surgeon guides freely the driller held by a slave robotic arm that avoids undesired drillings by means of virtual protections. With this application not only proximity queries are shown, but also the graphical interface and the use of a virtual robot based on RPQ. More information and videos are available at *http://grins.upc.edu*

2 RELATED WORK

During the last years, great efforts have been devoted to the development of efficient collision detection algorithms due to their wide range of applications, such as CAD/CAM, manufacturing, robotics, simulation and computer animation. A wide study of the performance and applicability of such methods can be found in (B.Geiger, 2000). Proximity query algorithms vary in terms of their range of applicability and the type of queries they can solve, mainly collision detection, minimum distance computation and interpenetrations modelling. Although most algorithms allow as input triangulated meshes of 3D points, they differ in the way those points are pre-processed and represented internally in order to speed up specific queries.

There is a wide set of methods that rely on Lin-Canny or Gilbert-Johnson-Keiethi like algorithms for computing minimum distances between pairs of objects as I-Collide, Swift, Swift++, SOLID, DEEP..., but they are only applicable to convex polytopes (S.Ehmann and Lin, 2000; S.Ehmann and Lin, 2001; Kim et al., 2002; Bergen, 2002). This restriction makes them inappropriate for RPQ purposes, due to the need to deal with more general geometric models.

More general collision detection methods usually base their efficiency on pre-computed representations by means of hierarchies of bounding volumes. Their differences rely on the specific type of bounding volumes used, ranging from binary space decompositions, spheres trees to oriented bounding boxes (OBB).

Among this set of algorithms, RAPID and PQP turn to be those that have both, fewer restrictions in the range of allowable geometric models and an easier application programming interface (API). Both of them use oriented bounding boxes for performing collision tests, and have similar time performances. However, the fact that PQP offers a wider range of queries, including minimum distance computation and tolerance tests makes PQP the best option for the proximity queries engine of RPQ, the Robotics Query Package presented in this paper.

3 LIBRARY DESCRIPTION

The goal of the Robotic Proximity Queries (RPQ) library is to offer an easy, modular and fast proximity query package oriented to robotics. As explained above, the aim of the project was not the development of a new collision detector, but specialize an existing one into the robotics field.

As described in section 2, there is a wide set of general purpose proximity query packages. The criterions used to choose PQP as the best candidate for the development of RPQ are:

1. Types of proximity queries available.
2. High time performance on proximity queries.
3. Ability to use geometrical models based on triangulated meshes of points.
4. Lack off restrictions on possible geometrical models.

The PQP library has been developed by UNC Research Group on Modelling, Physically-Based Simulation and Applications and offers three different kind of queries:

- *Collision detection*: detecting whether two models overlap, and optionally, give the complete list of overlapping triangle pairs.
- *Distance computation*: computing the minimum distance between a pair of models.
- *Tolerance verification*: determining whether two models are closer or farther than a given tolerance distance.

RPQ has been implemented in C++ language and its graphical interface has been developed using OpenGL. The RPQ library can be easily integrated into any software application.

The library interface allows non expert programmers to use it in an easy manner. The graphical interface is a separate module, allowing the programmer to decide whether using it or not. Fig. 1 shows the integration of the library and its graphical interface into a generic application.

3.1 Class Description

The RPQ library is based on the Object Oriented paradigm. Focused on this paradigm, and based on robotic environments, three main classes have been developed: Scenario, Object and Robot.

3.1.1 Scenario

Scenario is the workspace where the objects cohabit. Concerning its implementation, Scenario is a class that contains all the objects (Robots and generic objects), a global reference frame, and all the methods necessary to generate the proximity query.

3.1.2 Object

An Object is the minimum entity that exists in a Scenario. There are two types of Objects: simple and complex. A simple Object is represented by a geometrical model composed of a set of triangles referred to a frame tied to the Object. The Object has also a transformation matrix to refer itself to the world reference frame. A complex Object is an Object composed of a set of geometrical models with joints (rotational or prismatic) between them. Thus, a complex Object is an open kinematic chain composed of sub objects. The transformation matrix M_i refers subobject$_i$ to subobject$_{i-1}$. The transformation matrix M_0 refers the object base (subobject$_0$) to the world. The object stores its own geometrical model. Concerning its implementation, an Object is a class containing the geometrical model, the transformation matrix and a set of methods to position and to orient itself in space. This class also contains methods to calculate the different detail representations of its geometrical model.

3.1.3 Robot

A Robot is a particularization of a complex Object where each of its links is represented by a simple Object. A Robot has a set of functions to make a complex Object as similar as possible to a real robot. For instance, the spatial relationship between links is described using the Denavit-Hartenberg notation. Direct and inverse kinematics can be calculated considering the robots own restrictions (joint limitations, configurations, etc). Concerning implementation, the class Robot is derived from the class Object. Robot adds all the functions that are necessary to control a robot. For instance joint positioning of the robot (direct kinematics), position and orientation of its tool center point (inverse kinematics), change of the robot configuration, joints overshoot . . . These added functions with respect an Object are very helpful when a new robot is created or used in robotic applications like simulators, path planners, etc.

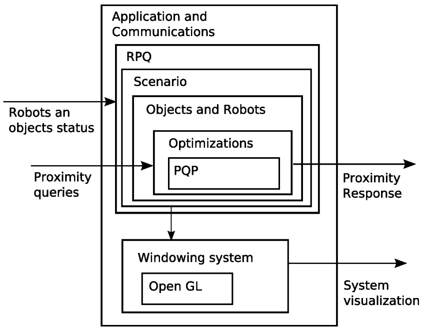


Figure 1: Schema of integration of RPQ into a generic application.

3.2 Optimizations

PQP is a generic package that does not use the knowledge of the object's kinematics. In contrast, RPQ is oriented to robotics, and the knowledge of robot's kinematics is the base of the optimizations that specialize it. RPQ is designed to answer proximity queries between two robots or between a robot and any kind of rigid object.

Three optimizations have been developed and tested to improve the performance offered by PQP:

- Different resolution levels of object's representation
- Collision queries sorted using a Weight Matrix
- Collision Matrix

3.2.1 Different Resolution Levels of Object's Representation

Objects can be represented in very different resolution levels. The idea of this optimization is to use the simplest representation models (minimum number of triangles) to discard collisions. The lower the number of triangles of the geometric model are, the faster the collision queries are executed.

Three resolution levels are used to represent robots and two for the rest of objects. The highest resolution level is the complete geometrical model. The second level is the oriented bounding box (OBB) of each sub object in which a complex object is divided. The lowest resolution level is the bounding box of the whole complex object. This level is only defined for complex objects with more than a sub object, as in robots with several links. There are other possible intermediate resolution levels that can be used, for instance the convex hull. It offers a good ratio between resolution and the quantity of triangles, although it does not reduce it as drastically as the low resolution levels chosen.

This optimization is useful in two different situations. First, in applications where no high precision is required, for instance when the precision of the OBB or the convex hull of each link is enough. The second situation occurs when the different resolution levels are used in a complementary manner.

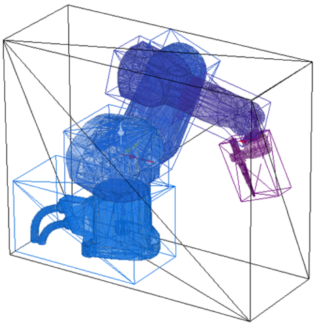


Figure 2: Robot with three resolution level representation: The geometrical model of each link (L3), the OBB of each link (L2) and the whole bounding box(L1).

When a collision query is performed, a low to high resolution level list of collision queries is generated.

Starting with the lowest resolution level, queries are generated until any collision can be completely discarded. For instance, if a possible collision between two 6-DOF robots is studied, the first query is done between the bounding boxes of each robot. If the collision can not be discarded, then the bounding box of each link is used. If at this level collisions still can not be discarded, the geometrical models of each link are checked. As shown in Fig. 2 a Robot with its three resolution levels of representation.

A test has been developed for the evaluation of the performance of the optimizations. It consists on a couple of virtual robotic arms (Staubli RX60B) that are placed one close to the other in a scenario. The geometrical models used for the test are high resolution (composed of 23012 triangles each). The robots are placed at a variable distance between them and the scenario is checked for collisions for an equidistributed set of joint positions in their 6 dof.

This test allows us to study the dependency on the performance of the proposed improvements in terms of the probability of collision. This is because, as shown in Table 1, in the designed test, the closer the robots are, the greater the number of joint configurations that result in collision.

Table 1: Dependence of the amount of colliding configurations on the distance between robots.

Dist. robots(m)	Joint config. checked	Collis.	Not Collis.
0,4	9216	4864	4352
0,5	9216	3530	5686
0,6	9216	2500	6716
0,7	9216	1121	8095
0,8	9216	140	9076

Fig. 3 shows the consequences of using different resolution levels. When the distance between the robots increases, the queries solved with the bounding box of the robot increases, and consequently the time to solve a collision query between the robot decreases. If the distance decreases, the best combination is using levels two and three or only level three.

3.2.2 Collision Queries Sorted using a Weight Matrix

This optimization is based on two assumptions. The first one is that the goal of the query is just to know whether there is a collision or not, but not the number of them. The second assumption is that the kinematics and the morphology of the robots are well known.

Given these assumptions, the objective is to find quickly whether there is collision or not, by means of minimizing the number of partial collision queries.

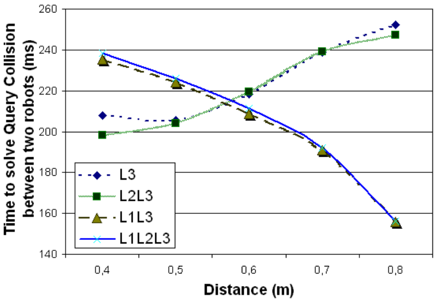


Figure 3: Time to solve a collision query between two Staubli RX60B robots using different resolution levels. L3: Geometrical model of each link. L2: The OBB of each link. L1: Bounding box of the robot.

The knowledge of the kinematics and the morphology of the robots gives us the possibility of assigning an *a priori* collision probability to each link of the robot with respect to the rest of robots and objects present in the same workspace. During execution time, these probabilities are automatically updated depending on the result of the collision queries (Probability increases in case of detecting a collision and decreases otherwise). Therefore, a weight matrix C is generated combining the probability of collision between each pair of objects in the workspace. Each component $c_{ij} \in C$ verifies $c_{ij} = P_i + P_j$ where P_i and P_j are the assigned probability of collision of Object$_i$ and Object$_j$ respectively. These weights determine the order of the collision queries, that is if $c_{ij} > c_{kt}$ the collision query between Object$_i$ and Object$_j$ will be generated before Object$_k$ and Object$_t$.

A simple way to assign a collision probability to the links of a robot is to assign higher probability to those links that are farther in the kinematic chain, with respect to the base of the robot.

3.2.3 Collision Matrix

The Collision Matrix is a binary matrix that reflects the possibility of collision between two objects. If the collision matrix indicates that a collision between two objects is impossible, its correspondent collision query is not performed. Of course, a matrix query is much less expensive than a collision query in computational terms.

This optimization improves the performance of the system when a high number of collision queries are discarded by the Collision Matrix. Computationally, this condition can be expressed as in equation (1):

$$n \cdot QC > m \cdot QM + k \cdot (QC + QM) \quad (1)$$
$$\text{with} \qquad n = m + k$$

where:

n Total number of queries.

m Queries resolved with the Collision Matrix.

k Queries resolved with the Query Collision.

QC Average time to solve a Query Collision.

QM Time to solve a query with the Col. Matrix.

The performance of the Collision Matrix has been studied using the same test designed for testing the use of different levels of representation. The results are shown in Fig. 4. The farther the robots are, the lower is the number of links that can collide, as seen in Table 2. Therefore, the higher the number of queries that are solved with the Collision Matrix. As it is shown in the figure, using the Collision Matrix the number of collision queries decreases, so does the time to solve the query.

Table 2: Dependence of the percentage of collision queries solved by the collision matrix on the distance between robots.

Dist. between robots (m)	Pairs solved with CM
0,4	4,94
0,6	9,88
0,9	24,69
1,2	64,20

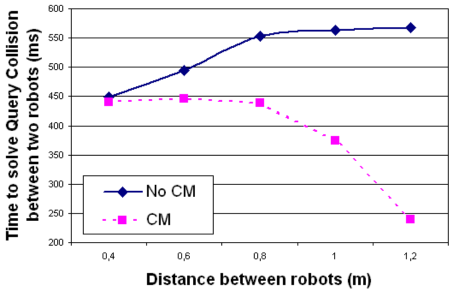


Figure 4: Time necessary to solve a collision query between two Staubli RX60B robots using the Collision Matrix (CM) or not (NoCM).

Each one of the optimizations proposed improves the performance of the original library, PQP. However, the combined use of all of them improves even more the global performance of the application.

The complete algorithm with all three optimizations is shown in Fig. 5. First of all, a query collision between the whole bounding box of both robots is performed. If at this level the collision cannot be solved then it is necessary to study collisions among the whole set of links of both robots. The order in which these queries must be performed is given by the Weight Matrix. The query finishes as soon as a collision appears between a pair of links either in the second or third level, or when all pairs have not reported any collision. For each pair of links, the second and third representation levels are studied consecutively, so if a collision is detected in the second level, the third level has to be studied as well.

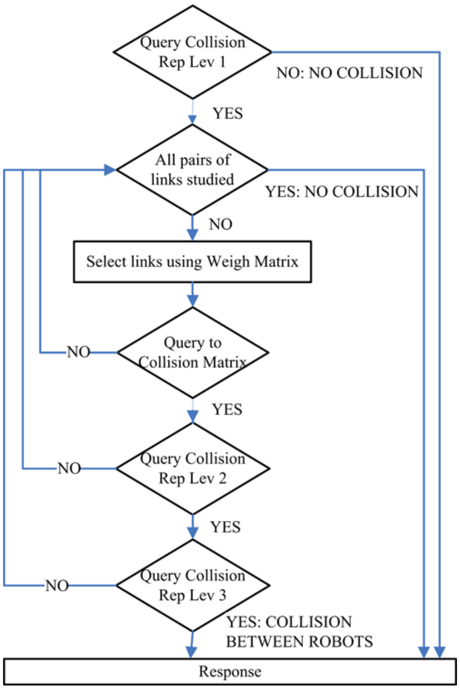

Query Collision Rep Lev 1
NO: NO COLLISION
YES
All pairs of links studied
YES: NO COLLISION
NO
Select links using Weigh Matrix
Query to Collision Matrix
NO
YES
Query Collision Rep Lev 2
NO
YES
Query Collision Rep Lev 3
NO
YES: COLLISION BETWEEN ROBOTS
Response

Figure 5: Algorithm for collision detection between two robots using all the optimizations.

4 APPLICATION

One of the advantages of making RPQ generic (although it is optimized for robots) is that this library can be applied in a wide set of applications. In this paper a robotic assisted surgical application is presented. This application has the aim of helping the surgeon to make a cut on a rigid tissue, for example in the cranium, avoiding undesired collisions between the patient and the driller. The surgeon guides freely the driller that is held by the robot acting in a passive mode as seen in Fig. 6, allowing all movements except those which produce undesired collisions.

The system is composed by a Staubli RX60B robotic arm and a driller. Between the robot and the

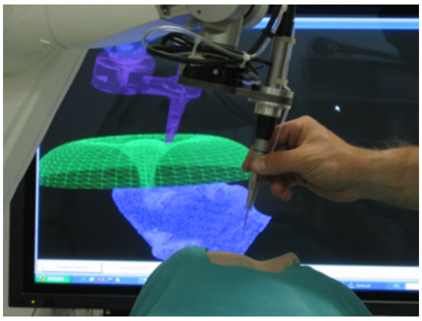


Figure 6: Robotic assisted surgical application.

driller there's a ATI Multi-Axis Force/Torque Sensor. The system transforms forces and torques generated in the driller by the surgeon in new destination points were the robot must go.

The geometric model of the patient's cranium is obtained transforming Computer Tomography data into a geometrical model composed of triangles. The surgeon can define the cutting area in the pre-operative phase and, as will be explained latter, virtual fixtures are automatically generated to protect the patient in the operative phase. The possibility of introducing virtual objects in the scene and the interaction between them is one of the key factors of RPQ.

The surgeon has a friendly and easy-use graphical interface that allows the navigation over the virtual system. This graphical interface helps the surgeon not only in the pre-operative phase but also during the surgical operation, providing an augmented reality visual feedback (collisions, minimal distance between the cranium and the tool, different points of view of the scene, virtual fixtures).

4.1 Surface Navigation

The library developed is useful not only to avoid collisions but also to navigate over an object's surface. For instance, the robot tool slides over the surface of the virtual shield described in section 4.2. This surface navigation allows the surgeon to feel smooth movements of the robot when the tool is in contact with the virtual fixtures. The navigation algorithm helps the surgeon not only avoiding the forbidden regions defined in the pre-operative phase but also guiding him to the desired cutting path.

The navigation algorithm modifies the position of the robot tool, but not its orientation. The algorithm is based on three steps: Knowing the new desired destination of the robot tool, the first step consists of detecting all collisions between the tool and the object's surface. When all collisions are detected, the second step consists of projecting the desired point to the plane of the collision triangle, Fig. 7.a. Finally the projected point that is closer to the desired one is selected. A problem can occur when the projected point falls outside the triangle region. In this situation it is not possible to ensure that this new projected point is always be outside the object Fig. 7.c. In this case the new point is projected to the perimeter of the triangle. To accomplish this, the outside region of the triangle is divided into six new regions ($R1, R12, R2, R23, R3, R31$), which are defined by the normals of the edges of the triangle applied to the vertices of the triangle. The point is then projected to the closest point of the triangle border of its region Fig. 7.b. Now, the new destination point is collision free Fig. 7.d.

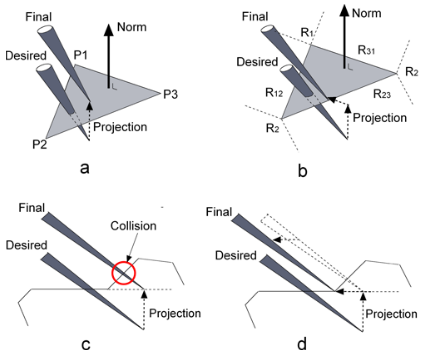


Figure 7: Different contact situations between the tool and a triangle or set of triangles.

4.2 Virtual Protections

Virtual Protections are constraints that rule the behaviour of the robot that are specifically designed to prevent motion into a forbidden region of the workspace. In this work, the surgeon guides freely a driller held by the slave robot. The main idea is to develop a system helpful for the surgeon that prevents undesired drillings.

4.2.1 Strategy Description

The main goal of the system is to give the robot a reactive behaviour by means of the definition of virtual objects in the PQP Scenario with two objectives:

- Protect the patient from the robot and the driller.
- Help the surgeon to locate the desired cutting area.

PQPs ability to check collisions in real time allows us not only to achieve these objectives but to operate in an efficient manner.

Throughout a simple interface, in the pre-operative phase, the surgeon specifies the exact location and shape of the cut that must be done. With that information, the system generates a shield that covers the patient while it adapts perfectly to the target area defined by the surgeon, as shown in Fig. 8.

The proposed behaviour is achieved by the navigation algorithm exposed in chapter 4.1. The system gives the surgeon the confidence of knowing that the robot will never let him approach the patient in a non-desired area. However, while the surgeon does not attempt to cross the shield, the reobot can be moved freely.

4.2.2 Shield Generation

The problem here is to generate a surface that connects the inner polygonal curve defined by the surgeon with an outer curve. The connection between them is done in a smooth way, as Fig. 8 illustrates.

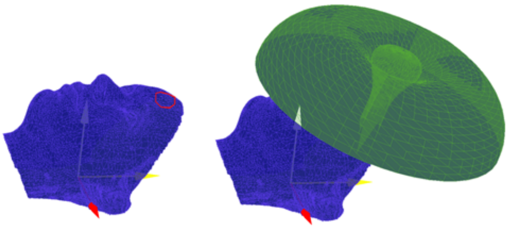


Figure 8: Lateral view of a face and the virtual shield.

The surface, $s(u,v)$, is parametrized as the upper part of a regular torus, but adapting its equators to the inner and outer curves, as in (4).

$$\text{Inn. curve} \quad i\left(t\right) = \left(i_x\left(t\right), i_y\left(t\right)\right) \quad t \in 0..2\pi \qquad (2)$$

$$\text{Out. curve} \quad o\left(t\right) = \left(o_x\left(t\right), o_y\left(t\right)\right) \quad t \in 0..2\pi \qquad (3)$$

$$\begin{aligned} s(u,v) \quad = \quad & \Big(\sin^2\left(\frac{v}{2}\right) i_x\left(u\right) + \cos^2\left(\frac{v}{2}\right) o_x\left(u\right), \\ & \sin^2\left(\frac{v}{2}\right) i_y\left(u\right) + \cos^2\left(\frac{v}{2}\right) o_y\left(u\right), \\ & H \sin\left(v\right)\Big) \quad u \in 0..2\pi \quad v \in 0..\pi \quad (4) \end{aligned}$$

There are peculiarities of the shield to consider:

- The shield should be smooth, in order to make navigation over it as comfortable as possible. Once the surface is generated it is transformed into a triangle model. By augmenting or decreasing the number of triangles, one can vary the smoothness of the surface.
- Reaching the inner curve from the surface it has to be easy and intuitive. This implies certain restrictions on the shield design. This property depends on the way inner points are connected to outer points. First of all inner and outer curves have to be parametrized, (2) and (3). There are some restrictions regarding them in order to make the surface easy and intuitive.
  - $i\left(t\right)$ and $o\left(t\right)$, must be parametrized radially, from an inner point, with constant radial velocity, in order to avoid self intersections of the surface, as in Fig. 9 a) and b).
  - The polygon defined by $i\left(t\right)$, must be star-shaped from the parametrization origin, in order to avoid self intersections of the surface, as in Fig. 9 c) and d).
  - If $i\left(t\right)$ defines a non star-shaped polygon a previous step must be done. An adaptation disc between the polygon and its convex hull is needed before constructing the shield, so that the latter can be parametrized radially from a suitable inner point, as in Fig. 10.

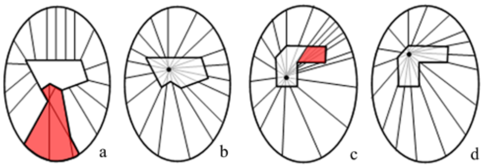


Figure 9: a) Non valid parametrization of curves (self intersection of the surface). b) Valid parametrization of curves. c) Non valid point for radial parametrization of curves. Polygon is not star-shaped from that point. d) Valid point for radial parametrization of curves.

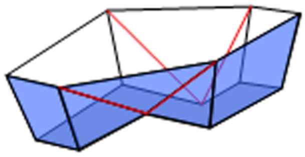


Figure 10: Adaptation disc between an non star-shaped polygon and its correspondent 2D Convex Hull.

- H, the height of the shield, must be chosen in order to protect the patient.

Once the required smoothness for the surface is defined, the parametrization for $i\left(t\right)$ and $o\left(t\right)$, solved the problem with non star-shaped polygons and chosen a desired height, the surface is generated by eq. (4).

Finally, the intersections of the discrete set of geodesics $u = \frac{2\pi}{N}n$ and $v = \frac{\pi}{N}n$ for $n \in 1 \ldots N$ is triangulated in order to obtain the geometrical model. The parameter N allows us to choose the number of points and triangles of the surface, and ultimately its smoothness.

4.2.3 Cutting Aid

Besides the shield generated, another virtual object is proposed in order to aid the surgeon to carry out the cut with great accuracy. It consists of an inner cover that prevents the driller from escaping the perimeter of the shape defined.

Since (2) has the shape of the cut to be done, we define $\widehat{i}\,(t)$ as an under scaled version of $i\,(t)$. $\widehat{i}\,(t)$ has the same shape as $i\,(t)$ but its sides are placed at a distance Δx from $i\,(t)$ sides, being Δx the desired width for the cut.

Therefore, a new shield is generated, $\widehat{s}(u,v)$, having a semisphere shape, with $\widehat{i}\,(t)$ as its equator, parametrized by (5).

$$\begin{aligned} &\widehat{s}(u,v) = \\ = \ &\Big(\big(\widehat{H}\cos^2(v)\cos(u) + \sin^2(v)\widehat{i}_x(u)\big)\sin(v), \\ &\big(\widehat{H}\cos^2(v)\sin(u) + \sin^2(v)\widehat{i}_y(u)\big)\sin(v), \\ &\widehat{H}\cos(v)\Big) \ \ u \ \in 0..2\pi \quad v \ \in 0..\frac{\pi}{2} \qquad (5) \end{aligned}$$

5 CONCLUSION

This paper presents the development of RPQ, a proximity queries library optimized for applications where robots share a common workspace and interact with objects. Due to the amount of collision detection packages, RPQ is built above a generic collision package, PQP. It is the generic collision package that better fits RPQ purposes.

The goal of developing RPQ was to fill an existing gap in computer tools for robotic applications, where robots interact with objects in their environment. Three optimizations have been performed to a generic collision library: working with different resolution levels of representation, the use of a weighted matrix for choosing the best order for collision checking and the definition of a binary matrix that determines the possibility of collision between objects. RQP has been validated in different applications such as multirobot collision avoidance, virtual robot controller and surface navigation.

As expected, optimizations improve the time performance of the system, although this improvement is highly application dependent.

The introduction of different levels of resolution in the geometric models of the objects and robots generally decreases the computational time for collision checking. The use of bounding boxes decreases drastically the number of high resolution queries needed. This is a really important point taking into account that they are much more time consuming. There are cases where low resolution queries do not solve the whole collision query. This increases the computation time. However, choosing a suitable order for checking collisions helps to find them in a quicker manner. The precomputation of impossibilities of collision between different objects (Collision Matrix) increases the performance of the system in case of having objects with restricted mobility in the workspace.

The combined use of optimizations generate good results in workspaces shared by at least two robots and objects.

RPQ has a wide range of applicability. RQP library is not only useful for proximity queries but has also proved to be a good tool for surface navigation and virtual representations, due to its ability to introduce virtual objects in the shared workspace. The virtual fixtures developed in the paper are examples of how RPQ can be used to modify robot's behaviour.

As proved in the application presented, RPQ is not only useful for developers of robotic applications, but also for users of robotic applications, i.e. surgeons that require new robotic tools for improving surgical procedures.

REFERENCES

Bergen, G. V. D. (2002). Solid collision detection library users guide.

B.Geiger (2000). Real-time collision detection and response for complex environments. In *International Conference on Computer Graphics*. IEEE Computer Society.

Giralt, X. and Hernansanz, A. (2006). Optimization of proximity queries in robotic environments. In *AVR - 2es Jornades UPC de Recerca en Automtica, Visio i Robotica (in catalan)*.

Kim, Y., Lin, M., and Manocha, D. (2002). Deep: Dual-space expansion for estimating penetration depth between convex polytopes. In *International Conference on Robotics and Automation*. IEEE.

S.Ehmann and Lin, M. (2000). Accelerated proximity queries between convex polyhedra by multilevel voronoi marching. Technical report, Department of Computer Science, University of North Carolina.

S.Ehmann and Lin, M. (2001). Accurate and fast proximity queries between polyhedra using convex surface decomposition. In *Eurographics*, volume 3.

Stanisic, Z., Jackson, E., and Payandeh, S. (1996). Virtual fixtures as an aid for teleoperation. In *9th Canadian Aeronautics and Space Institute Conference*.

UNC (1999). Pqp - a proximity query package by research group on modeling, physically-based simulation and applications.

EVOLUTIONARY PATH PLANNING FOR UNMANNED AERIAL VEHICLES COOPERATION

Ioannis K. Nikolos and Nikos Tsourveloudis
Intelligent Systems and RoboticsLaboratory, Technical University of Crete, Chania, Greece
jnikolo@dpem.tuc.gr, nikost@dpem.tuc.gr

Keywords: 3-D Path Planning, Evolutionary Algorithms, Navigation, UAV cooperation, B-Splines.

Abstract: We suggest an evolutionary based off-line/on-line path planner for cooperating Unmanned Aerial Vehicles (UAVs) that takes into account the environment characteristics and the flight envelope and mission constraints of the cooperating UAVs. The scenario under consideration is the following: a number of UAVs are launched from the same or different known initial locations. The main issue is to produce 3-D trajectories that ensure a collision free operation with respect to mission constraints. The path planner produces curved routes that are represented by 3-D B-Spline curves. Two types of planner are discussed: The off-line planner generates collision free paths in environments with known characteristics and flight restrictions. The on-line planner, which is based on the off-line one, generates collision free paths in unknown static environments, by using acquired information from the UAV's on-board sensors. This information is exchanged between the cooperating UAVs in order to maximize the knowledge of the environment. Both off-line and on-line path planning problems are formulated as optimization problems, with a Differential Evolution algorithm to serve as the optimizer.

1 INTRODUCTION

Path planning is the generation of a space path between an initial location and the desired destination that has an optimal or near-optimal performance under specific constraints (Gilmore, 1991). Searching for optimal paths is not a trivial task; in most cases results in excessive computation time and in some cases even the computation of just a feasible path is a very difficult problem. Therefore, in most cases we search for suboptimal or just feasible solutions.

Unmanned Aerial Vehicles (UAVs) path planning algorithms were initially developed for the solution of the single vehicle case. However, the continuously increasing interest for cooperating UAVs resulted in the development of algorithms that take into account the special issues and characteristics of such missions. Cooperation between UAVs has gained recently an increased interest as systems of multiple vehicles engaged in cooperative behavior show specific benefits compared to a single vehicle. Cooperative behavior may be defined as follows (Cao et al., 1999): Given some tasks specified by a designer, a multiple robot system displays cooperative behavior if, due to some underlying mechanism, i.e. the "mechanism of cooperation", there is an increase in the total utility of the system.

Path planning problems are computationally demanding multi-objective multi-constraint optimization problems (Mettler et al., 2003). The problem complexity increases when multiple UAVs should be used. Various approaches have been reported for UAVs coordinated route planning, such as Voronoi diagrams (Beard et al., 2002), mixed integer linear programming (Richards et al., 2002), (Schouwenaars et al., 2004) and dynamic programming formulations (Flint et al., 2002). Computational intelligence methods, such as Neural Networks (Ortega and Camacho, 1996), Fuzzy Logic (Kwon and Lee, 2000) and Evolutionary Algorithms (EAs), (Nikolos et al., 2003), (Zheng et al., 2005) have been successfully used to produce trajectories for guiding mobile robots in known, unknown or partially known environments.

EAs have been successfully used in the past for the solution of the path-finding problem in ground based or sea surface navigation (Michalewicz, 1999). A common practice was to model the path using straight line segments, connecting successive way points. In the case of partially known or dynamic environments a feasible and safe trajectory was planned off-line by the EA and the algorithm was used on-line whenever unexpected obstacles were sensed (Smierzchalski, 1999), (Smierzchalski and Michalewicz, 2000). EAs have been also used for solving the path-finding problem in a 3-D environment for underwater vehicles, assuming that the path is a sequence of cells in a 3-D grid (Sugihara and Yuh, 1997).

In (Nikolos et al., 2003) an EA based framework was utilized to design an off-line / on-line path planner for UAVs, which calculates a curved path line represented by B-Spline curves in a 3-D environment. The coordinates of the B-Spline control points serve as design variables. For both off-line and on-line planners, the problem is formulated as an optimization one. Constraints are formulated using penalty functions.

The work in this paper is the outgrowth of the one presented in (Nikolos et al., 2003) for the case of multiple and cooperating UAVs. The scenario considered here is the following: a number of UAVs are launched from the same or different known initial locations with predefined initial directions. The main issue is to produce 3-D trajectories, represented by 3-D B-Spline curves, which connect the initial locations with a single destination location and ensure a collision free operation with respect to mission constraints.

Two types of path planner are discussed: The *off-line planner* generates collision free paths in environments with known characteristics and flight restrictions. The *on-line planner*, being an extension of the off-line one and based on the ideas presented in (Nikolos et al., 2003) was developed to generate collision free paths in unknown environments. The knowledge of the environment is gradually acquired through the on-board sensors that scan the area within a certain range from each UAV. This information is exchanged between the cooperating UAVs in order to maximize the sensors effectiveness. The on-line planner rapidly generates a near optimum path for each vehicle that will guide the vehicle safely to an intermediate position within the known territory, taking into account the mission and cooperation objectives and constraints. The process is repeated until the corresponding final position is reached by an UAV. Then, each one of the remaining UAVs uses the acquired information about the environment and the off-line planner output to compute a path that connects its current position and the final destination. Both path planning problems are formulated as optimization (minimization) problems, where specially constructed functions take into account mission and cooperation objectives and constraints, with a differential evolution algorithm to serve as the optimizer.

The rest of the chapter is organized as follows: section 2 contains B-Spline and differential evolution algorithms fundamentals. The off-line path planner will be briefly discussed in section 3. Section 4 deals with the concept of cooperating UAV on-line path planning using differential evolution. The problem formulation is described, including assumptions, objectives, constraints, cost function definition and path modeling. Simulations results are presented in section 5, followed by discussion and conclusions in section 6.

2 DIFFERENTIAL EVOLUTION OF B-SPLINE PATHS

2.1 Path Modelling using B-Splines

Straight-line segments that connect a number of waypoints have been used in the past to model UAV paths in 2D or 3D space (Moitra, 2003), (Zheng et al., 2005). However, these simplified paths cannot be used for an accurate simulation of UAV's flight, unless a large number of waypoints is adopted. Furthermore, if an optimization procedure is used, the number of design variables explodes, especially if cooperating flying vehicles are considered. As a result, the computation time becomes impractical for real world applications.

B-Spline curves have been used in the past for trajectory representation in 2-D (Alfaro and Garcia, 1988) or in 3-D environments (Nikolos et al., 2003), (Nikolos et al., 2001). Their parametric construction provides the ability to produce non-monotonic curves, like the trajectories of moving objects. If the number of control points of the corresponding curve is $n+1$, with coordinates (x_0, y_0, z_0) ,…, (x_n, y_n, z_n), the coordinates of the B-Spline curve may be written as:

$$x(u) = \sum_{i=0}^{n} x_i \cdot N_{i,p}(u), \qquad (1)$$

$$y(u)=\sum_{i=0}^{n} y_i \cdot N_{i,p}(u), \tag{2}$$

$$z(u)=\sum_{i=0}^{n} z_i \cdot N_{i,p}(u), \tag{3}$$

where u is the free parameter of the curve, $N_{i,p}(u)$ are the blending functions of the curve and p is its degree, which is associated with curve's smoothness ($p+1$ being its order). Higher values of p correspond to smoother curves (Farin, 1988).

The blending functions are defined recursively in terms of a *knot* vector $U=\{u_0,\ldots, u_m\}$, which is a non-decreasing sequence of real numbers, with the most common form being the *uniform non-periodic* one, defined as:

$$u_i = \begin{cases} 0 & \mathit{if} \quad i < p+1 \\ i-p & \mathit{if} \quad p+1 \le i \le n \\ n-p+1 & \mathit{if} \quad n < i. \end{cases} \tag{4}$$

The blending functions $N_{i,p}$ are computed using the knot values defined above, as:

$$N_{i,0}(u) = \begin{cases} 1 & \mathit{if}\ u_i \le u < u_{i+1} \\ 0 & \mathit{otherwise}, \end{cases} \tag{5}$$

$$N_{i,p}(u)=\frac{u-u_i}{u_{i+p}-u_i}N_{i,p-1}(u)+\frac{u_{i+p+1}-u}{u_{i+p+1}-u_{i+1}}N_{i+1,p-1}(u). \tag{6}$$

If the denominator of the two fractions in (6) is zero, that fraction is defined to have zero value. Parameter u varies between 0 and $(n-p+1)$ with a constant step, providing the discrete points of the B-Spline curve. The sum of the values of the blending functions for any value of u is always 1.

2.2 Differential Evolution Algorithm

Differential Evolution (DE) (Price et al., 2005) is used in this work as the optimization tool. The classic DE algorithm evolves a fixed size population, which is randomly initialized. After initializing the population, an iterative process is started and at each generation G, a new population is produced until a stopping condition is satisfied. At each generation, each element of the population can be replaced with a new generated one. The new element is a linear combination between a randomly selected element and the difference between two other randomly selected elements. Below, a more analytical description of the algorithm's structure is presented.

Given a cost function $f(X):\mathbb{R}^{n_{param}} \to \mathbb{R}$, the optimization target is to minimize the value of this cost function by optimizing the values of its parameters (design variables), that is

$$X = \left(x_1, x_2, \ldots, x_{n_{param}}\right),\ x_j \in \mathbb{R}, \tag{7}$$

where X denotes the vector composed of n_{param} cost function parameters (design variables). These parameters take values between specific upper and lower bounds, as follows:

$$x_j^{(L)} \le x_j \le x_j^{(U)},\ j=1,\ldots,n_{param} \tag{8}$$

The DE algorithm implements real encoding for the values of the objective function's variables. In order to obtain a starting point for the algorithm, an initialization of the population takes place. The initialization (for $G=1$) is established by randomly assigning values to the parameters j of each i member of the population, within the given boundaries

$$x_{i,j}^{(1)} = r\cdot\left(x_j^{(U)}-x_j^{(L)}\right)+x_j^{(L)},\ i=1,\ldots,n_{pop},\ j=1,\ldots,n_{param} \tag{9}$$

where r is a uniformly distributed random value within range [0, 1]. DE's mutation operator is based on a triplet of randomly selected different individuals. For each i member of the population, a new parameter vector $V_i^{(G)}$ is generated by adding the weighted difference vector between the two members of the triplet to the third one (the donor). That is:

$$\begin{aligned} V_i^{(G)} &= X_{r3_i}^{(G)} + F\cdot\left(X_{r1_i}^{(G)} - X_{r2_i}^{(G)}\right), \\ V_i^{(G)} &= \left(v_{i,1}^{(G)}, v_{i,2}^{(G)}, \ldots, v_{i,n_{param}}^{(G)}\right), \end{aligned} \tag{10}$$

where $X_{r3_i}^{(G)}$ is called the "donor", G is the current generation, and

$$\begin{aligned} & i=1,\ldots,n_{pop},\ j=1,\ldots,n_{param} \\ & r1_i \in \left[1,\ldots,n_{pop}\right],\ r2_i \in \left[1,\ldots,n_{pop}\right], r3_i \in \left[1,\ldots,n_{pop}\right] \\ & r1_i \ne r2_i \ne r3_i \ne i \\ & F \in [0,1+],\ r \in [0,1] \end{aligned} \tag{11}$$

In this way a perturbed individual is generated. The perturbed individual $V_i^{(G)}$ and the initial population member $X_i^{(G)}$ are then subjected to a crossover operation that generates the final candidate solution $U_i^{(G+1)}$

$$u_{i,j}^{(G+1)} = \begin{cases} v_{i,j}^{(G)} & if\ \left(r \le C_r \vee j = k\right) \forall\ j = 1,\ldots,n_{param} \\ x_{i,j}^{(G)} & otherwise \end{cases} \qquad (12)$$
$$C_r \in [0,1]$$

where k is a random integer within [1, n_{param}], chosen once for all members of the population. The random number r is seeded for every gene of each chromosome. F and C_r are DE control parameters, which remain constant during the search process and affect the convergence behaviour and robustness of the algorithm. Their values also depend on the objective function, the characteristics of the problem and the population size.

The population for the next generation (G+1) is selected between the current population and the final candidates. If each candidate vector is better fitted than the corresponding current one, the new vector replaces the vector with which it was compared. The DE selection scheme is described as follows (for a minimization problem):

$$X_i^{(G+1)} = \begin{cases} U_i^{(G+1)} & if\ f\left(U_i^{(G+1)}\right) \le f\left(X_i^{(G)}\right) \\ X_i^{(G)} & otherwise \end{cases} \qquad (13)$$

A new scheme (Hui-Yuan et al., 2003) to determine the donor for mutation operation has been adopted for accelerating the convergence rate. In this scheme, donor is randomly selected (with uniform distribution) from the region within the "hyper-triangle", formed by the three members of the triplet presented below:

$$donor_i^{(G)} = \sum_{k=1}^{3}\left(\lambda_k \Big/ \sum_{m=1}^{3}\lambda_m\right) X_{rk_i}^{(G)}, \quad \lambda_m = rand\,[0,1] \qquad (14)$$

where *rand* [0, 1] denotes a uniformly distributed value within the range [0, 1]. With this scheme the donor comprises the local information of all members of the triplet, providing a better starting-point for the mutation operation that result in a better distribution of the trial-vectors.

The random number generation (with uniform probability) is based on the algorithm presented in (Marse and Roberts, 1983). In each different operation inside the DE algorithm that requires a random number generation, a different sequence of random numbers is produced, by using a different initial seed for each operation and a separate storage of the corresponding produced seeds.

3 OFF-LINE PATH PLANNER

The off-line planner generates collision free paths in environments with known characteristics and flight restrictions. The derived path lines are continuous 3-D B-Spline curves.

Each B-Spline control point is defined by its three Cartesian coordinates $x_{k,j}$, $y_{k,j}$, $z_{k,j}$ ($k=0,\ldots,n$, $j=1,\ldots,N$), N is the number of UAVs, while $n+1$ is the number of control points in each B-Spline curve (the same for all curves). The first ($k=0$) and last ($k=n$) control points of the control polygon are the initial and target points of the j^{th} UAV, which are predefined by the user. The second ($k=1$) control point is positioned in a pre-specified distance from the first one, in a given altitude, and in a given direction, in order to define the initial direction of the corresponding path.

For the case of a single UAV the optimization problem to be solved minimizes a set of five terms, connected to various objectives and constraints; they are associated with the feasibility of the curve, its length and a safety distance from the ground. The cost function to be minimized is defined as:

$$f = \sum_{i=1}^{5} w_i f_i \qquad (15)$$

Term f_1 penalizes the non-feasible curves that pass through the solid boundary. In order to compute this term, discrete points along each curve are computed, using B-Spline equations (Eq. 1 to 6) and a pre-specified step for B-Spline parameter u.

Term f_2 is the length of the curve (non-dimensional with the distance between the starting and destination points) and is used to provide shorter paths.

Term f_3 is designed to provide flight paths with a safety distance from solid boundaries.

Term f_4 is designed to provide B-Spline curves with control points inside the pre-specified space.

Term f_5 was designed to provide path lines within the known terrain. This characteristic is particularly useful when the off-line path planner is used together with the on-line one, as it will be explained later.

Weights w_i are experimentally determined, using as criterion the almost uniform effect of the last four terms in the objective function. Term $w_1 f_1$ has a dominant role in (15) providing feasible curves in few generations, since path feasibility is the main concern. The minimization of (15), through the DE procedure, results in a set of B-Spline control points, which actually represents the desired path.

4 ON-LINE PATH PLANNING

Having *N* UAVs launched from the same or different known initial locations, the issue is to produce *N* 3-D trajectories, aiming at reaching a predetermined target location, while ensuring collision avoidance with the environmental obstacles. Additionally, the produced flight paths should satisfy specific route constraints. Each vehicle is assumed to be a point, while its actual size is taken into account by equivalent obstacle – ground growing.

The general constraint of the problem is the collision avoidance between UAVs and the ground. The route constraints are:

(a) Predefined initial and target coordinates for all UAVs

(b) Predefined initial directions for all UAVs,

(c) Predefined minimum and maximum limits of allowed-to-fly space (minimum and maximum allowed Cartesian coordinates for all path points).

The cooperation objective is that all UAVs should reach the same target point.

The on-line planner uses acquired information from all UAV's on-board sensors (that scan the area within a certain range from the corresponding UAV). The planner rapidly generates a near optimum path, modeled as a 3-D B-Spline curve that will guide each vehicle safely to an intermediate position within the already scanned area. The information about the already scanned area by each UAV is passed to the rest cooperating UAVs, in order to maximize the knowledge of the environment. The process is repeated until the final position is reached by a UAV. Then the rest of UAVs turn to the off-line mode and a single B-Spline path for each UAV is computed to guide the corresponding vehicle from its current position, through the already scanned territory to the common final destination. As a result, each path line from the corresponding starting point to the final goal is a smooth, continuous 3-D line that consists of successive B-Spline curves, smoothly connected to each other (Figure 1).

As the terrain is completely unknown and the radars (or equivalent sensors) gradually scan the area, it is impossible to generate feasible paths that connect each starting point with the target one. Instead, at certain moments, each sensor scans a region around the corresponding moving UAV and this region is added to the already scanned regions by all cooperating UAVs. For the UAV under consideration a path line is generated that connects a temporary starting point with a temporary ending point. Each temporary ending point is also the next curve's starting point for the corresponding vehicle. Therefore, what is finally generated is a group of smooth curve segments connected to each other, eventually connecting the starting point to the final destination for each UAV.

In the on-line problem only four control points define each B-Spline curve, the first two of which are fixed and determine the direction of the current UAV path. The remaining two control points are allowed to take any position within the scanned by the radars known space, taking into consideration given constraints.

When the next path segment is to be generated, only the first control point of the B-Spline curve is known (it is the last control point of the previous B-Spline segment). The second control point is not random as it is used to make sure that at least first derivative continuity of the two connected curves is provided at their common point. Hence, the second control point of the next curve should lie on the line defined by the last two control points of the previous curve.

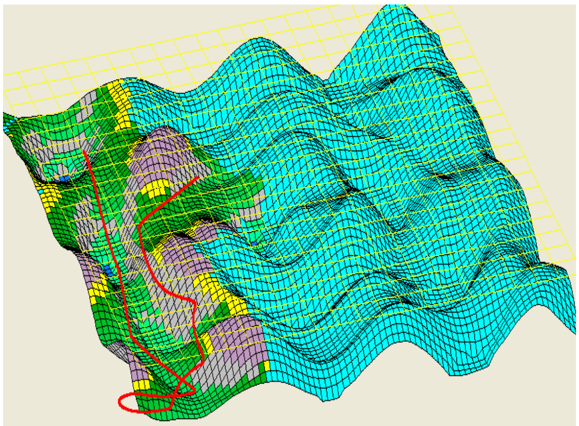


Figure 1: On-line path planning for a single UAV. The path is shown in an intermediate position of the UAV's flight. The already scanned area is presented in color.

The path-planning algorithm considers the scanned surface as a group of quadratic mesh nodes. All ground nodes are initially considered unknown. Radar information is used to produce the first path line segment for the corresponding UAV. As the vehicle is moving along its first segment and until it has travelled about 2/3 of its length, its radar scans the surrounding area, returning a new set of visible nodes, which are subsequently added to the common set of scanned nodes. The on-line planner, then, produces a new segment for each UAV, whose first point is the last point of the previous segment and whose last point lies somewhere in the already scanned area, its position being determined by the on-line procedure. The on-line process is repeated until the ending point of the current path line segment of one UAV lies close to the final destination. Then the rest UAVs turn into the off-line process, in order to reach the target using B-Spline curves that pass through the scanned terrain.

The computation of intermediate path segments for each UAV is formulated as a minimization problem. The cost function to be minimized is formulated as the weighted sum of eight different terms:

$$f = \sum_{i=1}^{8} w_i f_i \ , \qquad (16)$$

where w_i are the weights and f_i are the corresponding terms described below.

Terms f_1, f_2, and f_3 are the same with terms f_1, f_3, and f_4 respectively of the off-line procedure. Term f_1 penalizes the non-feasible curves that pass through the solid boundary. Term f_2 is designed to provide flight paths with a safety distance from solid boundaries. Only already scanned ground points are considered for this calculation. Term f_3 is designed to provide B-Spline curves with control points inside the pre-specified working space.

Term f_4 is designed to provide flight segments with their last control point having a safety distance from solid boundaries. This term was introduced to ensure that the next path segment that is going to be computed will not start very close to a solid boundary (which may lead to infeasible paths or paths with abrupt turns). The minimum distance D_{min} from the ground is calculated for the last control point of the current path segment. Only already scanned ground points are considered for this calculation. Term f_4 is defined as:

$$f_4 = \left(d_{safe} / D_{\min} \right)^2 \ , \qquad (17)$$

where d_{safe} represents a safety distance from the solid boundary.

The value of term f_5 depends on the potential field strength between the initial point of the UAVs path and the final target. This potential field between the two points is the main driving force for the gradual development of each path line in the on-line procedure. The potential is similar to the one between a source and a sink, defined as:

$$\Phi = \ln \frac{r_2 + c \cdot r_0}{r_1 + c \cdot r_0} \ , \qquad (18)$$

where r_1 is the distance between the last point of the current curve and the initial point for the corresponding UAV, r_2 is the distance between the last point of the current curve and the final destination, r_0 is the distance between the initial point for the corresponding UAV and the final destination and c is a constant. This potential allows for selecting curved paths that bypass obstacles lying between the starting and ending point of each B-Spline curve (Nikolos et al., 2003).

Term f_6 is similar to term f_5 but it corresponds to a potential field between the current starting point (of the corresponding path segment) and the final target.

Term f_7 is designed to prevent UAVs from being trapped in small regions and to force them move towards unexplored areas. In order to help the UAV leave this area, term f_7 repels it from the points of the already computed path lines (of all UAVs). Furthermore, if a UAV is wandering around to find a path that will guide it to its target, the UAV will be forced to move towards areas not visited before by this or other UAVs. This term has the form:

$$f_7 = \frac{1}{N_{po}} \sum_{k=1}^{N_{po}} \frac{1}{r_k} \ , \qquad (19)$$

where N_{po} is the number of the discrete curve points produced so far by all UAVs and r_k is their distance from the last point of the current curve segment.

Term f_8 represents another potential field, which is developed in a small area around the final target. When the UAV is away from the final target, the term is given a constant value. When the UAV is very close to the target the term's value decreases proportionally to the square of the distance between the last point of the current curve and the target.

Weights w_i in (16) are experimentally determined, using as criterion the almost uniform effect of all the terms, except the first one. Term $w_1 f_1$ has a dominant role, in order to provide feasible curve segments in a few generations, since path feasibility is the main concern.

5 SIMULATION RESULTS

The same artificial environment was used for all test cases considered here. The artificial environment is constructed within a rectangle of 20x20 (non-dimensional lengths). The (non-dimensional) radar's range for each UAV was set equal to 4. The safety distance from the ground was set equal to d_{safe}=0.25. The (experimentally optimized) settings of the DE algorithm during the on-line procedure were as follows: *population size* = 20, F = 0.6, C_r = 0.45, *number of generations* = 70. For the on-line procedure we have two free-to-move control points, resulting in 6 design variables. The corresponding settings during the off-line procedure were as follows: *population size* = 30, F = 0.6, C_r = 0.45, *number of generations* = 70. For the off-line procedure eight control points were used to construct each B-Spline curve (including the initial (k=0) and the final one (k=7). These correspond to five free-to-move control points, resulting in 15 design variables. All B-Spline curves have a degree p equal to 3. All experiments have been designed in order to search for path lines between "mountains". For this reason, an upper ceiling for flight height has been enforced in the optimization procedure, by explicitly providing an upper boundary for the z coordinates of all B-Spline control points.

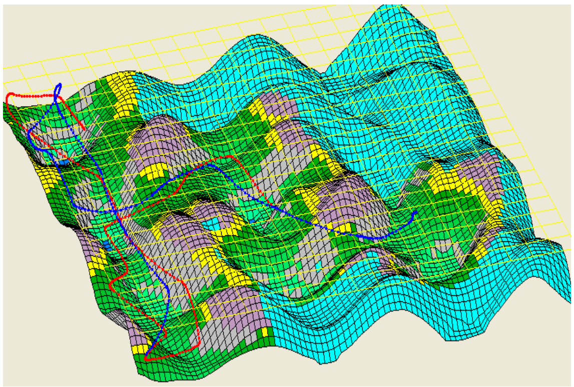


Figure 2: Test case 6 corresponds to the on-line path planning of two UAVs. When the first UAV (blue line) reaches the target the second one turns into the off-line mode. The starting point for the first UAV is close to the lower left corner of the terrain, while for the second one is close to the upper left corner.

Test case 1 corresponds to on-line path planning of two UAVs. Figure 2 shows the path lines when the first UAV (blue line) reaches the target. In that moment the second UAV (red line) turns into the off-line mode, in order to compute a feasible path line that connects its current position with the target, through the already scanned area. The final status is demonstrated in Figure 3. The starting point for the first UAV is near the lower left corner of the terrain, while for the second one is near the upper left corner.

Test case 2 corresponds to the on-line path planning of three UAVs. Figure 4 shows the status of the two path lines when the first UAV (blue line) reaches the target. In that moment the second UAV (red line) and the third one (black line) turn into the off-line mode, in order to compute feasible path lines that connect their positions with the target. The final paths for all three UAVs are demonstrated in Figure 5. The starting point for the first and second UAVs are the same as in case 1, while the third UAV is near the middle of the left side of the terrain.

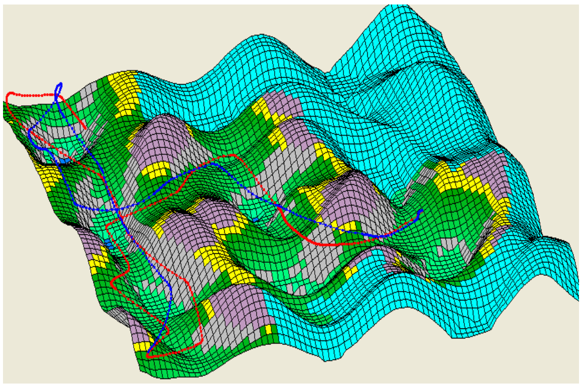


Figure 3: The final path lines for the test case 1.

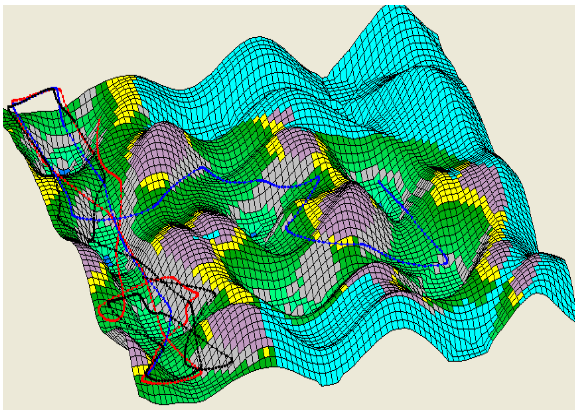


Figure 4: On-line path planning of three cooperating UAVs (test case 2). The picture shows the paths when the first UAV (blue line) reaches the target.

Test case 3 also corresponds to the on-line path planning of three UAVs but with distant starting points. Figure 6 shows the status of the two path lines when the first UAV (blue line) reaches the target. The final status is demonstrated in Figure 7. As the first UAV (blue line) is close to the target, it succeeds in reaching it using just one B-Spline segment. Then, the other two UAVs turn into off-line mode to reach the target.

In the *test case 4* three UAVs are launched from the centre of the working space but towards different directions. Figure 8 shows the status of the two path lines when the first UAV (blue line) reaches the target. The final status is demonstrated in Figure 9. When the final point of a curve segment is within a small distance from the final destination, the on-line procedure is terminated; this is the reason for the absence of coincidence between the final points of the first (blue line) and the rest path lines.

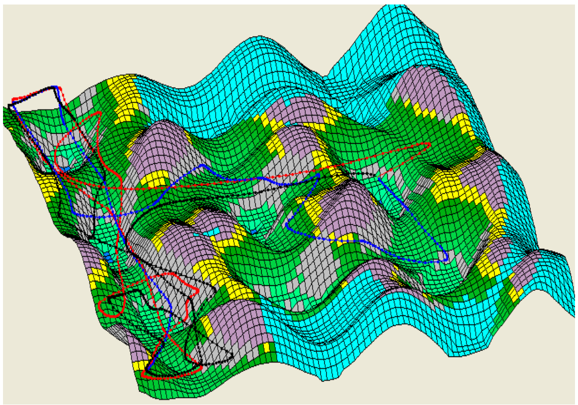


Figure 5: The final paths of the test case 2.

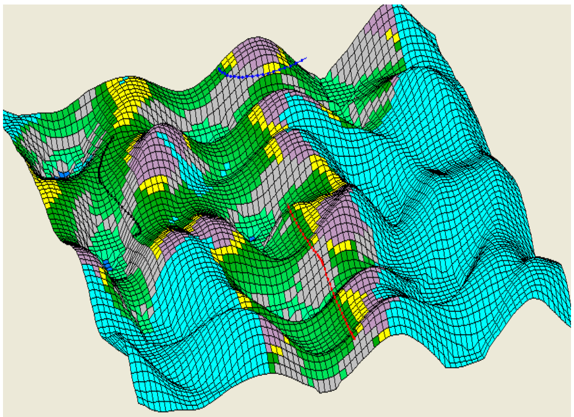


Figure 6: Path lines of three distant UAVs when the first one (blue line) reaches the target.

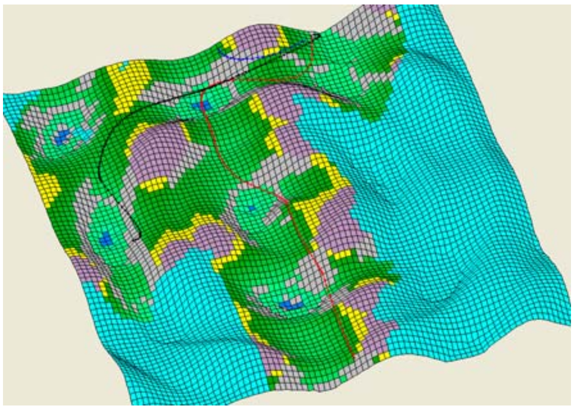


Figure 7: The final paths of test case 3.

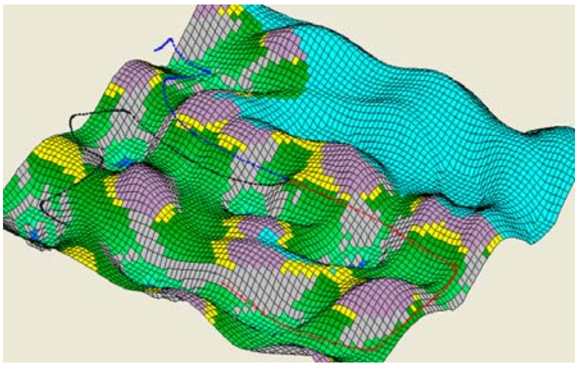


Figure 8: On-line path planning for three UAVs launched from the same point but different initial directions. The first UAV (blue line) just reached the target.

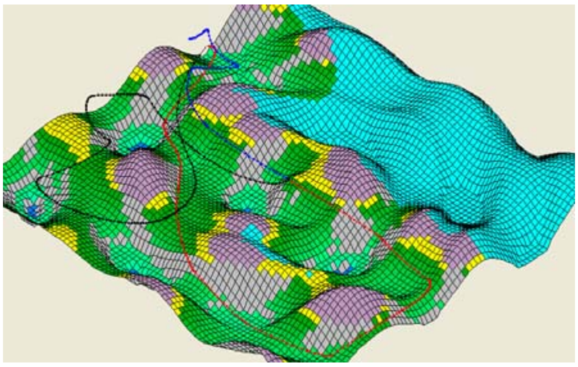


Figure 9: The final paths of test case 4.

6 CONCLUSIONS

A path planner for a group of cooperating UAVs was presented. The planner is capable of producing smooth path curves in known or unknown static environments. Two types of path planner were presented. The off-line one generates collision free paths in environments with known characteristics and flight restrictions. The on-line planner, which is based on the off-line one, generates collision free paths in unknown environments. The path line is gradually constructed by successive, smoothly connected B-Spline curves, within the gradually scanned environment. The knowledge of the environment is acquired through the UAV's on-board sensors that scan the area within a certain range from each UAV. This information is exchanged between the cooperating UAVs; as a result, each UAV utilizes the knowledge of a larger region than the one scanned by its own sensors. The on-line planner generates for each vehicle a smooth path segment that will guide the vehicle safely to an intermediate position within the known territory. The process is repeated for all UAVs until the corresponding final position is reached by an UAV. Then, the rest vehicles turn into the off-line mode in order to compute path lines consisting of a single B-Spline curve that connect their current positions with the final destination. These path lines are enforced to lie within the already scanned region. Both path planners are based on optimization procedures, and specially constructed functions are used to encounter the mission and cooperation objectives and constraints. A differential evolution algorithm is used as the optimizer for both planners. No provision is taken by the on-line planner for collision avoidance between the cooperating vehicles; this can be encountered by an on board controller for each vehicle.

REFERENCES

Gilmore, J.F., 1991. Autonomous vehicle planning analysis methodology. In *Proceedings of the Association of Unmanned Vehicles Systems Conference.* Washington, DC, 503–509.

Zheng, C., Li, L., Xu, F., Sun, F., Ding, M., 2005. Evolutionary Route Planner for Unmanned Air Vehicles. *IEEE Transactions on Robotics*, 21, 609-620.

Uny Cao, Y., Fukunaga, A.S., Kahng, A.B., 1997. Cooperative Mobile Robotics: Antecedents and Directions. *Autonomous Robots*, 4, 7-27.

Mettler, B., Schouwenaars, T., How, J., Paunicka, J., and Feron E., 2003. Autonomous UAV guidance build-up: Flight-test Demonstration and evaluation plan. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, AIAA-2003-5744.

Beard, R.W., McLain, T.W., Goodrich, M.A., Anderson, E.P., 2002. Coordinated target assignment and intercept for unmanned air vehicles. *IEEE Transactions on Robotics and Automation*, 18, 911-922.

Richards, A., Bellingham, J., Tillerson, M., and How., J., 2002. Coordination and control of UAVs. In *Proceedings of the AIAA Guidance, Navigation and Control Conference*, Monterey, CA.

Schouwenaars, T., How, J., and Feron, E., 2004. Decentralized Cooperative Trajectory Planning of multiple aircraft with hard safety guarantees. In *Proceedings of AIAA Guidance, Navigation, and Control Conference and Exhibit*, AIAA-2004-5141.

Flint, M., Polycarpou, M., and Fernandez-Gaucherand, E., 2002. Cooperative Control for Multiple Autonomous UAV's Searching for Targets. In *Proceedings of the 41st IEEE Conference on Decision and Control.*

Gomez Ortega, J., and Camacho, E.F., 1996. Mobile Robot navigation in a partially structured static environment, using neural predictive control. *Control Eng. Practice*, 4, 1669-1679.

Kwon, Y.D., and Lee, J.S., 2000. On-line evolutionary optimization of fuzzy control system based on decentralized population. *Intelligent Automation and Soft Computing*, 6, 135-146.

Nikolos, I.K., Valavanis, K.P., Tsourveloudis, N.C., Kostaras, A., 2003. Evolutionary Algorithm based offline / online path planner for UAV navigation. *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, 33, 898-912.

Michalewicz, Z., 1999. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer Publications.

Smierzchalski, R., 1999. Evolutionary trajectory planning of ships in navigation traffic areas. *Journal of Marine Science and Technology*, 4, 1-6.

Smierzchalski, R., and Michalewicz Z., 2000. Modeling of ship trajectory in collision situations by an evolutionary algorithm. *IEEE Transactions on Evolutionary Computation*, 4, 227-241.

Sugihara, K., and Yuh, J., 1997. GA-based motion planning for underwater robotic vehicles. *UUST-10*, Durham, NH.

Moitra, A., Mattheyses, R.M., Hoebel, L.J., Szczerba, R.J., Yamrom, B., 2003. Multivehicle reconnaissance route and sensor planning. *IEEE Transactions on Aerospace and Electronic Systems*, 37, 799-812.

Martinez-Alfaro H., and Gomez-Garcia, S. 1988. Mobile robot path planning and tracking using simulated annealing and fuzzy logic control. *Expert Systems with Applications*, 15, 421-429.

Nikolos, I.K., Tsourveloudis, N., and Valavanis, K.P., 2001. Evolutionary Algorithm Based 3-D Path Planner for UAV Navigation. In *Proceedings of the 9th Mediterranean Conference on Control and Automation*, Dubrovnik, Croatia.

Farin, G., 1988. *Curves and Surfaces for Computer Aided Geometric Design, A Practical Guide*. Academic Press.

Price, K.V., Storn, R.M., Lampinen, J.A., 2005. *Differential Evolution, a Practical Approach to Global Optimization.* Springer-Verlag, Berlin Heidelberg.

Marse, K. and Roberts, S.D., 1983. Implementing a portable FORTRAN uniform (0,1) generator. *Simulation*, 41-135.

A NOVEL STRATEGY FOR EXPLORATION WITH MULTIPLE ROBOTS

Jonathan Rogge and Dirk Aeyels
SYSTeMS Research Group, Ghent University, Ghent, Belgium
Jonathan.Rogge@UGent.be,Dirk.Aeyels@UGent.be

Keywords: Multi-robot systems, coverage, exploration, demining.

Abstract: The present paper develops a novel strategy for the exploration of an unknown environment with a multi-robot system. Contrary to most exploration problems, the topographical properties of the space need not be mapped. The algorithm we propose is inspired by methods used for complete coverage of an area, where all free space has to be physically covered by all robots. In the present paper it is required that the entire free space is covered by the *sensors* of the robots, with a certainty of 100%. This weaker requirement enables us to scan more space in less time, compared to complete coverage algorithms. Moreover the shape of the robot formation adjusts itself to situations where obstacles, narrow spaces, etc. have to be passed. Communication between the robots is restricted to line-of-sight and to a maximum interdistance between robots. A direct application of the algorithm is mine field clearance.

1 INTRODUCTION

The research domain of multi-agent robot systems can be divided into subdomains according to the task given to the robot group (Ota, 2006). At present well-studied subdomains are motion-planning (also called path-planning), formation-forming, region-sweeping, and combinations of the foregoing. The problem considered in the present paper belongs to the discipline comprising region-sweeping. In this discipline two different robot tasks are usually considered.

In the first task a group of robots receives the order to *explore/map* an unknown region. The goal is to obtain a detailed topography of the desired area. A typical approach to tackle the above problem with multiple robots assumes unlimited communication (Burgard et al., 2005): since exploration algorithms are already devised for a single robot it seems straightforward to divide the area to be explored into disjunct regions, each of which is assigned to a single robot. The robots communicate to each other the area they have explored so that no part of the free space will be explored twice unnecessarily. At no point during the task are the robots trying to form a fixed formation. Each robot explores a different part of the unknown region and sends its findings to a central device which combines the data received from the robots into one global map of the area.

Closely related to the exploring/mapping task is the second task, called *complete coverage*, where the robots have to move over all of the free surface in configuration space. Typical applications are mine field clearance, lawn mowing and snow cleaning. The coverage problem has been addressed in the literature both in a deterministic and a probabilistic setting. In the probabilistic approach the robots are considered as if they were fluid or gas molecules satisfying the appropriate physical laws of motion (Kerr et al., 2004), (Keymeulen and Decuyper, 1994). Just as a gas by diffusion fills an entire space, the robots will cover all free space when time tends to infinity. In the remainder of the paper we focus on the deterministic setting. In this setting the robot group typically forms (partial) formations to solve the task. Reference (Choset, 2001) gives a short overview of existing techniques for multi-robot coverage problems. Different approaches to the coverage problem are found in (Cortés et al., 2004), (Kurabayashi et al., 1996), (Wong and MacDonald, 2004), (Zheng et al., 2005) (Rekleitis et al., 2004) and (Kong et al., 2006).

The problem statement of the present paper does not differ that much from the common exploration/mapping task and the complete coverage problem, but is rather a combination of both. It is required that all of the free space is *sensed* by the robots, but not necessarily physically covered. However, unlike the common exploration case, the sensing of the area does not have as goal to map the topography of the free space and the location of the obstacles in it. Our aim is to locate several unknown targets within the free space. Moreover, similar to the complete coverage setting we demand a 100% certainty that *all free space* has been covered by the sensors at the end of the exploration procedure, implying that all targets have been found. Since the robots no longer have to cover all free space physically, the novel algorithm will yield a time gain compared to complete coverage strategies. The developed algorithm is inspired by complete coverage strategies as (Rekleitis et al., 2004), (Kong et al., 2006). In these strategies it is assumed that the space to be explored does not have a maze-like structure with many narrow corridors, but is an open space containing only convex obstacles sparsely spread throughout. In the present paper we adopt these assumptions, postponing more complex geometries to future work. Our algorithm is presented in Section 2 of the paper. A short comparison between the algorithm presented here and that of (Rekleitis et al., 2004) is given in Section 3.

A specific application we have in mind is mine field clearance using chemical vapor microsensors (Gage, 1995). Once a landmine is deployed, the environment near the mine becomes contaminated with explosives derived from the charge contained in the mine. The vapor microsensors are able to detect the chemical vapor signature of the explosives emanating from the landmines from a distance. This implies that complete coverage algorithms may be too restrictive with respect to the demining problem. Performing the algorithm of the present paper, with the weaker requirement of sensor coverage, will result in a gain of time.

The algorithm can also be used in problems where a robot formation has to cross a terrain containing sparsely spread obstacles. There is a natural trade-off between coherence of the formation and avoidance of the obstacles. The robot group is allowed to split in order to pass the obstacles, resulting in faster progress of the group across the terrain. The algorithm ensures that once the obstacle is passed, the robots regroup.

2 AN ALGORITHM FOR COMPLETE SENSOR COVERAGE

2.1 Setting

Consider a population of N identical robots. Each robot is equipped with two types of sensors. One type serves as a means to detect the goal targets to be found in the assigned area, e.g. landmines; the other type is used to detect and locate other robots and obstacles in the neighborhood of the robot[1]. Both sensors have a maximum detection range s_t and s_r respectively. It is assumed that targets which come within the radius of the corresponding sensor area s_t or s_r of the robot are always detected, and that if they are located farther away than the distance s_t, s_r they are never detected. The robot configuration allows limited communication. First, this is expressed by the maximum detection range s_r as described above. Second, line-of-sight communication is assumed: two robots can only sense each other if they are sufficiently close to each other and if there is no obstacle located on the straight line connecting both robots.

Two robots are called *connected* to each other when they sense each other. Every robot is assigned an index number. The initial state of the robot configuration is such that robot i is connected to robots $i-1$ and $i+1$, $\forall i \in \{2,\ldots,N-1\}$. (Robot 1 is only connected to robot 2 and robot N is only connected to robot $N-1$.) An example of such a configuration is depicted in Figure 1 for $N = 6$. The dashed circles have radius s_t and signify the sensed area for goal targets of each robot. It is assumed that

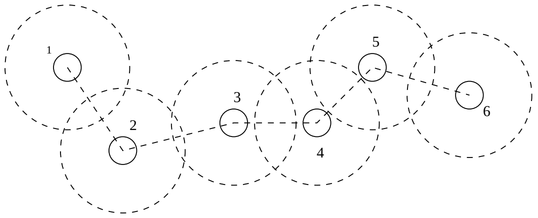


Figure 1: Overlapping sensor areas in a possible robot configuration.

$$\frac{s_r}{2} < s_t < s_r. \qquad (1)$$

[1] In practice the latter type consists of two distinct minimally interfering IR-sensors: one sensing obstacles and the other sensing robots. Since this is not relevant for the theoretical description of the algorithm, these sensors are considered as if they are one and the same.

The lower bound on s_t in (1) ensures that the areas sensed for goal targets of neighboring robots partially overlap, as illustrated by Figure 1.

In a second step, we impose a preferred formation on the robot group as follows: each robot keeps a constant distance $d < s_r$ with its neighbors and observes them at preferred angles with respect to its forward direction. With notation from Figure 2, these angles are defined by:

$$\alpha_i = \begin{cases} \pi/4, & i \text{ even}, \\ 3\pi/4, & i \text{ odd}, \end{cases}$$
$$\beta_i = \begin{cases} -\pi/4, & i \text{ even}, \\ -3\pi/4, & i \text{ odd}. \end{cases} \tag{2}$$

Furthermore each robot is equipped with a compass. Together with the above defined angles, the forward direction of each robot (the same for all robots) is imposed at the initialization of the algorithm. The above conditions imply a robot formation with zigzag shape, as shown in Figure 3. Once the preferred formation is attained the scanning algorithm is started.

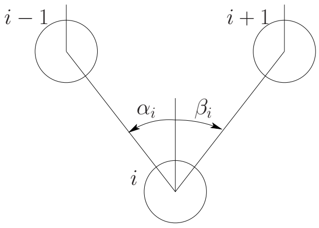


Figure 2: Defining the angles of the preferred robot configuration.

2.2 The Scanning Algorithm

Assume for simplicity that the area to be explored is a rectangular subset S of $\mathbb{R}^2$. All obstacles contained in S are assumed disjoint and convex. Divide the set S into parallel (scanning) strips of width $(\frac{N}{2}-1)d$. This choice of the value of the width will be motivated later, in Section 2.4. Furthermore assume that N and d are such that all obstacles in S have a diameter smaller than the width of a scanning strip. Fix the maximum allowed diameter at $(\frac{N}{2}-3)d$. The main idea of the algorithm is to let the group of robots sweep the area S strip after strip in a zigzag-like pattern. Clearly, when there is a sufficient number of robots available the set S can be regarded as one big strip, simplifying the algorithm since no transitions between consequent strips have to be performed.

In a first case we consider a strip where no objects are located on the boundary (see Figure 3). Robots 1 and N are allocated the task to follow the boundaries of the strip at a constant distance at the constant velocity v. They can be considered leaders of the robot group. These two leader robots do not try to stay in the preferred formation, i.e. the condition on the corresponding angles α_N, β_1 is removed, and they do not maintain a fixed interdistance d with their neighbors. The remaining robots, however, still maintain the preferred formation. When no obstacles are present in the strip, the robots scan the strip for goal targets in the above defined preferred (rigid) formation moving at a velocity v.

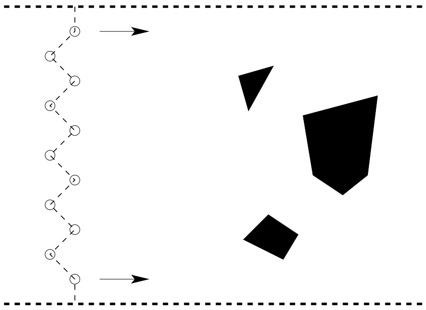


Figure 3: A depiction of the algorithm. The arrows indicate the constant velocity of both leader robots. The dashed lines represent the strip boundaries.

When an obstacle is encountered the algorithm aims to guide the robot group past it in a time-optimal way. The leader robots start driving at a preset velocity $v_0 < v$ (see Section 3); the group is split into two subgroups in order to move around the obstacle. The subgroups rejoin after passing the obstacle to resume the preferred formation structure.

Assume for reasons of simplicity that the number of robots N is even. Consider the situation where robot m encounters an object on its path such that it cannot stay in the preferred formation any longer. More precisely, the sensors of robot m measure

- an interdistance between the obstacle and the robot smaller than a preset distance $d_o < s_r$,
- the position of the obstacle at an angle with its forward direction inside the interval $(-\gamma, \gamma)$, with γ a fixed value inside the interval $(0, \frac{\pi}{4})$.

The presence of the obstacle is communicated to all the robots in the group. Each robot takes on a different role such that two subgroups will be formed. The robots with index $i \in S_1 := \{2, \ldots, N/2\}$ now follow the neighboring robot with corresponding index $i-1$. Similarly, robots with index $i \in S_2 := \{N/2+1, \ldots, N-1\}$ follow the neighboring robot with index $i+1$. More precisely, the robot with in-

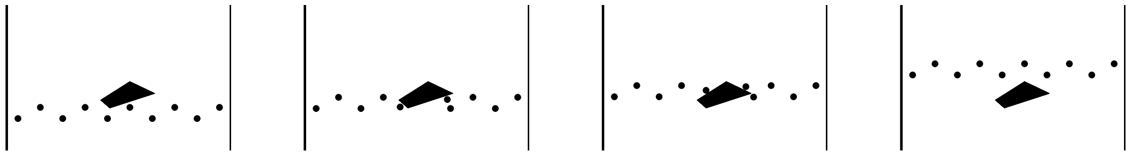


Figure 4: A group of 10 robots passing an obstacle. Only robots 5, 6 and 7 apply wall-following around the obstacle during the maneuver. This is shown in the second and third picture. The first picture shows the formation at the moment of encountering the obstacle. The fourth picture presents the formation after having completely past the obstacle.

dex *i* tries to reach the following coordinates:

$$\begin{cases} (x_{i-1} + d\sin\frac{\pi}{4}, y_{i-1} + (-1)^i d\cos\frac{\pi}{4}), & \text{if } i \in S_1, \\ (x_{i+1} - d\sin\frac{\pi}{4}d, y_{i+1} + (-1)^i d\cos\frac{\pi}{4}), & \text{if } i \in S_2. \end{cases} \tag{3}$$

These coordinates are considered with respect to a right-handed (x,y)-frame with the y-axis parallel to the strip boundary, and directed into the driving direction of the leader robots. Each robot still tries to stay in the preferred formation, but in order to do so only takes information of one neighbor into account. Moreover, the condition on the relative position between the neighboring robots $N/2$ and $N/2+1$ is suspended, which will lead to the splitting of the robot group. Notice that indifferent of the robot that observes the obstacle first, the group will split between robots $N/2$ and $N/2+1$. This choice is motivated in Section 2.4.

Consider the situation for robot *i* where one of the following occurs:

- The desired position (3) cannot be reached,
- The obstacle is blocking the straight path between the present position of robot *i* and its desired position,
- Robot *i* does not detect its neighbor necessary to determine its preferred position.

If this situation occurs, the robot receives the order to follow the edge of the obstacle, keeping the obstacle on its right if $i \in S_1$, or its left if $i \in S_2$. This behavior is called wall-following. The robot continues to wall-follow around the obstacle until none of the above conditions is satisfied. After that, it assumes its desired position again. If all robots have past the obstacle, each robot is again able to reach its desired position in the preferred formation. In particular, robots $N/2$ and $N/2+1$ will meet again in their desired relative position. When this happens a signal is sent to all robots with the message that the group has past the obstacle.

A simulation of the above described algorithm is presented in Figure 4 with $N = 10$ and $m = 6$.

Remark. It may occur that a robot cannot reach its desired position because it is located too far away from its present position. Then the robot simply rides towards the desired position at the maximum allowed velocity, trying to catch up.

Remark. If the number of robots N^* in the (sub)group is not even, then the indices of the robots where the robot group splits are $\lfloor N^*/2 \rfloor$ and $\lceil N^*/2 \rceil$, where $\lfloor . \rfloor$ is the function giving the largest integer less than or equal to its argument, and similarly, $\lceil . \rceil$ gives the smallest integer greater than or equal to its argument.

2.3 Multiple Obstacles

Suppose the robot group is already split into two subgroups and a robot in one of the subgroups encounters a second obstacle. The above obstacle avoidance algorithm can be made modular in order to solve this problem. A group can only split if both robots at the extremities of the group are leader robots, similar to the initial configuration. Assume group S_1 encounters a second obstacle. Robot $N/2$ is then turned into a leader robot. Instead of following a strip boundary it is ordered to follow the edge of the first obstacle, until it meets its neighbor $N/2+1$ or until group S_1 has past the second obstacle. In the latter case, robot $N/2$ takes on its role as a follower of robot $N/2-1$ again, in the former case it turns into a follower of $N/2+1$. The group S_1 is split into the middle and the algorithm described in the previous section is performed with leader robots 1 and $N/2$. In order for each robot to know which role to assume, it keeps track of how many times its subgroup is split.

Clearly, the number of times this splitting can be repeated is limited. We require a subgroup to consist of at least 3 robots: two leader robots on each side of the group, plus at least one robot in the middle attempting to follow both leaders while maintaining the formation structure. The middle robot ensures that the discs of sensed area of the separate robots overlap for all time instants.

2.4 Adaptation of the Basic Algorithm

Consider a worst case scenario as sketched in Figure 5. The robot formation splits into two subgroups, and the group on the left hand side moves through the gap between the obstacle and the left boundary of the scanning strip. Once past the gap the robots in this subgroup have to spread out, since the distance between the obstacle and the left boundary increases and we want to sense all of free space between the boundary and the obstacle. The obstacle has such a shape that the robots have to spread out across almost the entire width of the scanning strip before meeting a robot member of the right subgroup. The basic algorithm is modified as follows. When robot $N/2$ (resp. $N/2+1$) encounters the obstacle, it is now programmed to follow the obstacle's edge *until it meets its neighbor* $N/2+1$ *(resp.* $N/2$*)*. Additionally, it ensures that its neighbor $N/2-1$ (resp. $N/2+1$) stays in its detection range by sending a signal to the other robots of its subgroup to increase the angle $\pi/4$ of (3). This changes the desired position of each robot in the subgroup resulting in a stretching out of the group, as far as necessary. The above modified algorithm justifies our choice of initial formation and width of the scanning strip. If we had naively chosen a value $(N-1)d$ as the width of a scanning strip, the initial preferred robot formation would be able to span this entire distance, namely by forming a line with the angles defined in Section 2.1 equal to $\alpha_i = -\beta_i = \pi/2$. However, one subgroup, consisting of only half of the number of robots, would not be able to span this distance, resulting in either an error message from the algorithm or in unscanned areas, if a situation described in Figure 5 was encountered.

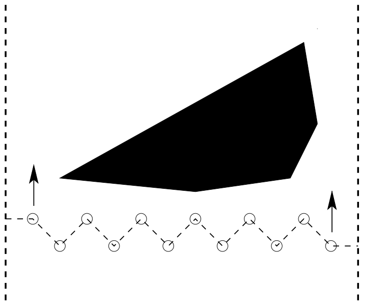


Figure 5: A depiction of the worst case scenario in the algorithm.

Closely related to this observation is the choice to split the robot group precisely in the middle. Since the sensor range of each robot is limited and the robots operate in an unknown environment, the shape of each obstacle is unknown. To guarantee that the area around the obstacle is fully covered by the sensors, we have to supply a sufficient number of robots to both sides of the obstacle. For instance, when the shape of the obstacle in Figure 5 is known a priori, one can decide to send more than half of the robots to the left of the obstacle. Consider the case where the obstacle is reflected with respect to the vertical axis. In this case sending less than half of the robots to the right would lead to uncovered areas or an error message in the algorithm. With limited sensor information it is not possible to discriminate between the situation of Figure 5 and its reflected version. This leads us to always split the group into two equal parts.

An alternative solution for this problem could be a more intelligent algorithm where individual robots transmit more sensor data to the others in order to find possible bounds on the size of the encountered obstacle. If the obstacle is small, a better way to split is right at the place where the obstacle is encountered. The robots do not have to deviate much from the preferred formation, which decreases the probability on error messages from deforming the robot configuration. This idea may be incorporated in future work.

2.5 Obstacles Located on the Boundary between Two Strips

Throughout the paper the obstacles are assumed to have a convex shape, in order to avoid robot groups getting stuck in a dead end. However, there is one case of dead ends we cannot avoid by the above assumption. A dead end can occur when an obstacle is located on the boundary between two strips, as presented on the left hand side of Figure 6. Since the robots have limited sensor information, they cannot conclude a priori whether an encountered obstacle stretches out into a neighboring strip or not. We are forced to let the algorithm run until a dead end is observed.

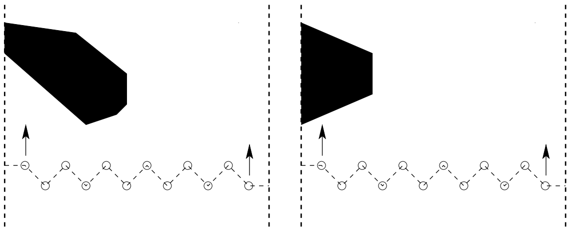


Figure 6: Two situations where an obstacle is located on the boundary between strips. On the left hand side a dead end situation arises; on the right hand side one of the leader robots guides the group around the obstacle.

Before tackling the dead end problem, let us treat the case presented on the right hand side of Figure 6, which does not lead to a dead end situation. Consider an (x,y)-frame with the y-axis parallel to the strip boundary, and directed into the driving direction of the leader robots. When the leader robot encounters the obstacle, the algorithm assigns to this leader a wall-following procedure around the obstacle. The leader keeps the obstacle either on its right or left (depending on its position in the robot formation) while moving into the interior of the strip away from the strip boundary. As can be concluded from the picture, the y-coordinate of the leader increases while moving around the obstacle. We wish to keep the velocity component of the leader robot parallel to the strip boundary equal to v. Since the robot deviates from its straight path parallel to the strip boundary, this implies it has to speed up. When the leader reaches the strip boundary again, it switches back to the original task of moving parallel to the boundary.

Now consider the left hand side of Figure 6. A dead end is detected by the algorithm when two conditions are satisfied:

- one of the leader robots cannot move into the desired direction parallel to the strip boundary, because an obstacle is blocking the way.
- when the leader robot starts wall-following the obstacle as described above, the value of its y-coordinate decreases.

As soon as a dead end is observed by the leader robot, it changes its behavior and stops its wall following algorithm. Instead, it projects its corresponding strip boundary $(N/2-1)d/8$ units outwards and resumes the original scanning algorithm with respect to the new boundary. If the extra width turns out to be insufficient to guide the robot subgroup around the obstacle outside of the original scanning strip, the boundary is projected a second (third,...) time. This way the subgroup which was stuck in the dead end is guided around the obstacle. When both subgroups reestablish contact, the leader robot returns to the original strip boundary. This behavior is faster and easier to implement than a turning-back scenario, where the subgroup of robots which meets a dead end retraces it steps to go around the obstacle inside the original scanning strip.

Remark. The above situation with a solid wall as strip boundary, forcing a turning-back maneuver, is precluded.

2.6 The Transition from One Strip to the Next

When the robot group reaches the end of a scanning strip, it needs to be transported to the next strip. This is done in a few easy steps. Consider the situation of Figure 7. First the right leader changes its behavior into that of a robot in the interior of the formation, i.e. it tries to attain the desired formation. The left leader moves $(N/2-1)d$ units to the left perpendicular to the strip boundary. The rightmost robot resumes its leader role and all robots reverse their forward direction with respect to the desired direction in the previous strip. Naturally, every time the end of a strip is reached, the roles of left and right leader alternate, so that the robot group does not get trapped into a loop consisting of two strips.

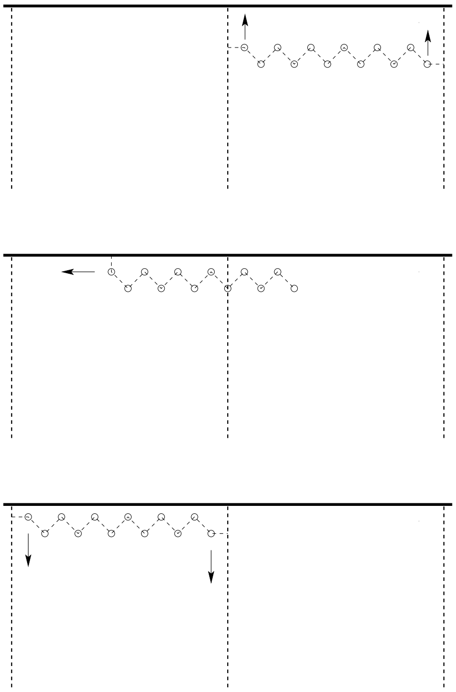


Figure 7: The robot group moves from the end of a scanning strip to the start of the next strip.

3 COMPARISON WITH COMPLETE COVERAGE ALGORITHMS

The algorithm presented in this paper is inspired by the complete coverage algorithm presented in (Rekleitis et al., 2004), which is depicted in Figure 8. The authors of (Rekleitis et al., 2004) propose the following robot configuration: 2 leader robots, following the strip boundaries, and a number of interior robots, traveling back and forth between the strip boundaries physically covering all the free space. Contrary to the algorithm proposed in the present paper, the leader robots maintain line-of-sight contact between each other. When an obstacle appears between the two leaders the line-of-sight contact is lost and the obstacle is detected. An appropriate control action is then taken by splitting the platoon and the algorithm is repeated on both sides of the obstacle. The splitting procedure includes the creation of two extra leader robots, as shown in Figure 8. Remark that the leaders are allowed to move ahead of the rest of the robot group and hence group coherence is not maintained or desired, contrary to our approach.

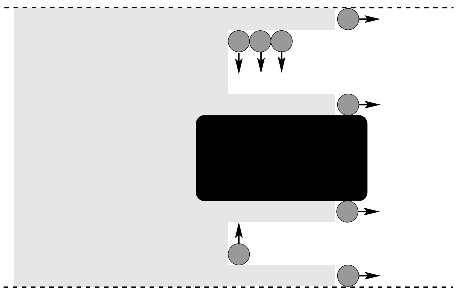


Figure 8: A depiction of the complete coverage algorithm by Rekleitis et al.

In the remainder of this section we will compare speed of performance of the present algorithm with the algorithm of (Rekleitis et al., 2004). In order to do so, realistic distance values are considered. Chemical vapor sensors detecting mines have a range $s_t = 1.70$ m. Obstacles and other robots can be detected by laser based sensors with a range of $s_r = 3.3$ m such that (1) is satisfied. Assume the robots themselves possess a diameter of 0.3 m and set the fixed interdistance d between neighboring robots in the preferred formation equal to s_r. With N the number of robots in the group, this yields a strip width of $1.65(N-2)$ m.

When no obstacles are encountered, the robots are allowed to move at a preset maximum velocity $v_{\max}$. In the algorithm of the present paper $v_{\max}$ is directed parallel to the strip boundary, whereas the interior robots in (Rekleitis et al., 2004) travel back and forth inside the strip at $v_{\max}$. It can be proven that for the latter case with the values given above the speed of progress parallel to the strip boundary is $v_{\max}/6$.

In the presence of obstacles a comparison is more difficult. First consider the complete coverage algorithm (Rekleitis et al., 2004). As can be concluded from Figure 8, in the presence of an obstacle the robots will advance faster parallel to the strip boundary, since the space occupied by the obstacle does not have to be covered. The robot group will proceed fastest when the shape of the obstacle is such that there is no space left for the robots to travel back and forth between obstacle and strip boundary. Hence, depending on size and shape of the obstacle the robots advance with a speed between $v_{\max}/6$ and $v_{\max}$. Now, consider the algorithm of the present paper. Some interior robots perform wall-following around the obstacles. This implies their path is longer than the path of the leader robots. If the leader robots keep moving at the maximum allowed velocity, those interior robots will never again be able to reach their desired position inside the formation after the obstacle is past. Hence, when an obstacle is encountered the leaders have to take on a velocity v_0 which is smaller than $v_{\max}$. This velocity v_0 is determined as follows. The middle robots $N/2$ and $N/2+1$ transmit their positions via the other robots to their respective leader robots. The leaders adjust their velocity v_0 such that the difference between their y-coordinate and the y-coordinate of the corresponding robot $N/2$ or $N/2+1$ stays at all time within a prespecified bound. The middle robots only slow down the group significantly during the first and last stage of their obstacle following, i.e. when moving away from or towards the strip boundary without significantly advancing parallel to it. As soon as there is enough free space ahead of the middle robots, the subgroup is again allowed to move parallel to the strip boundary with a speed close to $v_{\max}$.

From the above observations the following is concluded. The robot group in the present algorithm slows down to pass an obstacle, but for most of the time the speed will be close to $v_{\max}$. The robot group of the complete coverage algorithm speeds up when passing an obstacle, but for most obstacles the algorithm still requires a robot group moving back and forth between the obstacle and the strip boundary. This implies that the increased speed will on average be closer to $v_{\max}/6$ than to $v_{\max}$. Hence, in generic cases, the present algorithm performs faster than the complete coverage strategy even in the presence of obstacles.

4 CONCLUSIONS

The present paper described a novel strategy for multi-robot exploration of an unknown environment with guarantee of total sensor coverage. The algorithm we proposed is inspired by methods used for complete coverage as described in (Rekleitis et al., 2004). We took advantage of the fact that only complete *sensor* coverage is required. We let the robots form a spring-like formation which scans the area in strips. In the presence of obstacles the formation is deformed and split in two in order to circumvent the obstacles and to adapt to the varying width of the free space.

ACKNOWLEDGEMENTS

This paper presents research results of the Belgian Programme on Interuniversity Attraction Poles, initiated by the Belgian Federal Science Policy Office. The scientific responsibility rests with its authors.

REFERENCES

Burgard, W., Moors, M., Stachniss, C., and Schneider, F. (2005). Coordinated multi-robot exploration. *IEEE Transactions on Robotics*, 21(3):376–386.

Choset, H. (2001). Coverage for robotics – a survey of recent results. *Annals of Mathematics and Artificial Intelligence*, 31:113–126.

Cortés, J., Martínez, S., Karatas, T., and Bullo, F. (2004). Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, 20(2):243–255.

Gage, D. (1995). Many-robots mcm search systems. In *Proceedings of Autonomous Vehicles in Mine Countermeasures Symposium*.

Kerr, W., Spears, D., Spears, W., and Thayer, D. (2004). Two formal gas models for multi-agent sweeping and obstacle avoidance. In *Formal Approaches to Agent-Based Systems, Third International Workshop*, pages 111–130.

Keymeulen, D. and Decuyper, J. (1994). The fluid dynamics applied to mobile robot motion: the stream field method. In *Proceedings of 1994 IEEE International Conference on Robotics and Automation*, pages 378–385, Piscataway, NJ, USA.

Kong, C. S., Peng, N. A., and Rekleitis, I. (2006). Distributed coverage with multi-robot system. In *Proceedings of 2006 IEEE International Conference on Robotics and Automation*, pages 2423–2429, Orlando, Florida, USA.

Kurabayashi, D., Ota, J., Arai, T., and Yosada, E. (1996). Cooperative sweeping by multiple robots. In *Proc. 1996 IEEE International Conference on Robotics and Automation*.

Ota, J. (2006). Multi-agent robot systems as distributed autonomous systems. *Advanced engineering informatics*, 20:59 – 70.

Rekleitis, I., Lee-Shue, V., New, A. P., and Choset, H. (2004). Limited communication, multi-robot team based coverage. In *Proc. 2004 IEEE International Conference on Robotics and Automation*.

Wong, S. and MacDonald, B. (2004). Complete coverage by mobile robots using slice decomposition based on natural landmarks. In *Proc. Eighth Pacific Rim International conference on Artificial Intelligence. Lecture Notes in Artificial Intelligence.*, volume 3157, pages 683–692.

Zheng, X., Jain, S., Koenig, S., and Kempe, D. (2005). Multi-robot forest coverage. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*.

CALIBRATION OF QUASI-ISOTROPIC PARALLEL KINEMATIC MACHINES: ORTHOGLIDE

Anatoly Pashkevich, Roman Gomolitsky
Robotic Laboratory, Department of Control Systems
Belarusian State University of Informatics and Radioelectronics
6 P.Brovka St., Minsk 220027, Belarus
pap@bsuir.unibel.by

Philippe Wenger, Damien Chablat
Institut de Recherche en Communications et Cybernétique de Nantes
1, rue de la Noe B.P. 6597, 44321 Nantes Cedex 3, France
{ Philippe.Wenger, Damien.Chablat}@irccyn.ec-nantes.fr

Keywords: Parallel robots, calibration, parameter identification, error compensation.

Abstract: The paper proposes a novel approach for the geometrical model calibration of quasi-isotropic parallel kinematic mechanisms of the Orthoglide family. It is based on the observations of the manipulator leg parallelism during motions between the specific test postures and employs a low-cost measuring system composed of standard comparator indicators attached to the universal magnetic stands. They are sequentially used for measuring the deviation of the relevant leg location while the manipulator moves the TCP along the Cartesian axes. Using the measured differences, the developed algorithm estimates the joint offsets and the leg lengths that are treated as the most essential parameters. Validity of the proposed calibration technique is confirmed by the experimental results.

1 INTRODUCTION

Parallel kinematic machines (PKM) are commonly claimed as appealing solutions in many industrial applications due to their inherent structural rigidity, good payload-to-weight ratio, high dynamic capacities and high accuracy (Tlusty et al., 1999; Merlet, 2000; Wenger et al., 2001). However, while the PKM usually exhibit a much better repeatability compared to serial mechanisms, they may not necessarily posses a better accuracy, which is limited by manufacturing/assembling errors in numerous links and passive joints (Wang and Masory, 1993). Thus, the PKM accuracy highly relies on the accurate kinematic model, which must be carefully tuned for each manipulator separately.

Similar to the serial manipulators, the PKM calibration techniques are based on the minimization of a parameter-dependent error function, which incorporates residuals of the kinematic equations. For the parallel manipulators, the inverse kinematic equations are considered computationally more efficient (contrary to the direct kinematics, which is usually analytically unsolvable for the PKM) (Innocenti, 1995; Iurascu & Park, 2003; Jeong et al., 2004, Huang et al., 2005). But the main difficulty with this technique is the full-pose measurement requirement, which is very hard to implement accurately. Hence, a number of studies have been directed at using the subset of the pose measurement data, which however creates another problem, the identifiability of the model parameters.

Popular approaches in the parallel robot calibration deal with one-dimensional pose errors using a double-ball-bar system or other measuring devices, as well as imposing mechanical constraints on some elements of the manipulator (Daney, 2003). However, in spite of hypothetical simplicity, it is hard to implement in practice since an accurate extra mechanism is required to impose these constraints. Additionally, such methods reduce the workspace size and consequently the identification efficiency.

Another category of the methods, the self- or autonomous calibration, is implemented by minimizing the residuals between the computed and measured values of the active and/or redundant joint sensors. Adding extra sensors at the usually unmeasured joints is very attractive from computational point of view, since it allows getting the data in the whole workspace and potentially reduces impact of the measurement noise. However, only a partial set of the parameters may be identified in this way, since the internal sensing is unable to provide sufficient information for the robot end-effector absolute location.

More recently, several hybrid calibration methods were proposed that utilize intrinsic properties of a particular parallel machine allowing extracting the full set of the model parameters (or the most essential of them) from a minimum set of measurements. It worth mentioning an innovative approach developed by Renaud et al. (2004, 2005) who applied the vision-based measurement system for the PKM calibration from the leg observations. In this technique, the source data are extracted from the leg images, without any strict assumptions on the end-effector poses. The only assumption is related to the manipulator architecture (the mechanism is actuated by linear drives located on the base). However, current accuracy of the camera-based measurements is not high enough yet to apply this method in industrial environment.

This paper extends our previous research (Pashkevich et al., 2006) and focuses on the calibration of the Orthoglide-type mechanisms, which is also actuated by linear drives located on the manipulator base and admits technique of Renaud et al. (2004, 2005). But, in contrast to the known works, our approach assumes that the leg location is observed for specific manipulator postures, when the tool-center-point moves along the Cartesian axes. For these postures and for the nominal Orthoglide geometry, the legs are strictly parallel to the corresponding Cartesian planes. So, the deviation of the manipulator parameters influences on the leg parallelism that gives the source data for the parameter identification. The main advantage of this approach is the simplicity and low cost of the measuring system that can avoid using computer vision and is composed of standard comparator indicators attached to the universal magnetic stands.

The remainder of the paper is organized as follows. Section 2 describes the manipulator geometry, its inverse and direct kinematics, and also contains the sensitivity analysis of the leg parallelism at the examined postures with respect to the geometrical parameters. Section 3 focuses on the parameter identification, with particular emphasis on the calibration accuracy under the measurement noise. Section 4 contains experimental results that validate the proposed technique, while Section 5 summarizes the main contributions.

2 ORTHOGLIDE MECHANISM

2.1 Manipulator Architecture

The Orthoglide is a three degrees-of-freedom parallel manipulator actuated by linear drives with mutually orthogonal axes. Its kinematic architecture is presented in Figure 1 and includes three identical parallel chains, which will be further referred as “legs”. Each manipulator leg is formally described as PRP_aR - chain, where P, R and P_a denote the prismatic, revolute, and parallelogram joints respectively (Figure 2). The output machinery (with a tool mounting flange) is connected to the legs in such a manner that the tool moves in the Cartesian space with fixed orientation (translational motions).

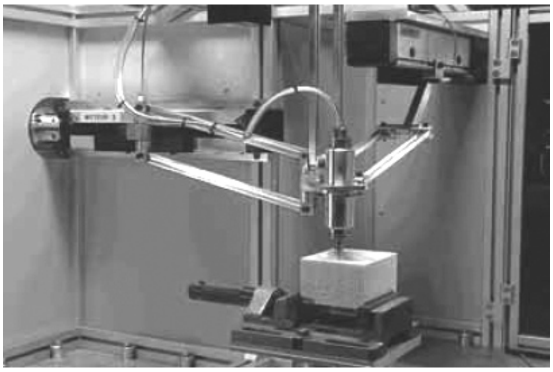


Figure 1: The Orthoglide kinematic architecture. (© CNRS Photothèque / CARLSON Leif)

The Orthoglide workspace has a regular, quasi-cubic shape. The input/output equations are simple and the velocity transmission factors are equal to one along the x, y and z direction at the isotropic configuration, like in a conventional serial PPP machine (Wenger et al., 2000; Chablat and Wenger, 2003). The latter is an essential advantage of the Orthoglide architecture, which also allows referring it as the “quasi-isotropic” kinematic machine.

Another specific feature of the Orthoglide mechanism, which will be further used for the calibration, is displayed during the end-effector motions along the Cartesian axes. For example, for the x-axis motion in the Cartesian space, the sides of the x-leg parallelogram must also retain strictly parallel to the x-axis. Hence, the observed deviation of the mentioned parallelism may be used as the data source for the calibration algorithms.

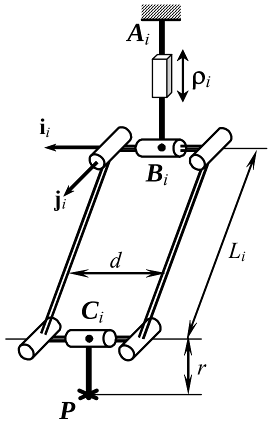


Figure 2: Kinematics of the Orthoglide leg.

For a small-scale Orthoglide prototype used for the calibration experiments, the workspace size is approximately equal to 200×200×200 mm^3 with the velocity transmission factors bounded between 1/2 and 2 (Chablat & Wenger, 2003). The legs nominal geometry is defined by the following parameters: L_i = 310.25 mm, d = 80 mm, r = 31 mm where L_i, d are the parallelogram length and width, and r is the distance between the points Ci and the tool centre point P (see Figure 2).

2.2 Modelling Assumptions

Following previous studies on the PKM accuracy (Wang & Massory, 1993; Renaud et al., 2004), the influence of the joint defects is assumed negligible compared to the encoder offsets and the link length deviations. This validates the following modelling assumptions:

(i) the manipulator parts are supposed to be rigid bodies connected by perfect joints;

(ii) the manipulator legs (composed of a prismatic joint, a parallelogram, and two revolute joints) generate a four degrees-of-freedom motions;

(iii) the articulated parallelograms are assumed to be perfect but non-identical;

(iv) the linear actuator axes are mutually orthogonal and intersected in a single point to insure a translational movement of the end-effector;

(v) the actuator encoders are perfect but located with some errors (offsets).

Using these assumptions, there will be derived new calibration equations based on the observation of the parallel motions of the manipulator legs.

2.3 Basic Equations

Since the kinematic parallelograms are admitted to be non-identical, the kinematic model developed in in our previous papers (Pashkevich et al., 2005, 2006) should be extended to describe the manipulator with different length of the legs.

Under the adopted assumptions, similar to the equal-leg case, the articulated parallelograms may be replaced by the kinematically equivalent bar links. Besides, a simple transformation of the Cartesian coordinates (shift by the vector $(r, r, r)^T$, see Figure 2) allows to eliminate the tool offset. Hence, the Orthoglide geometry can be described by a simplified model, which consists of three rigid links connected by spherical joints to the tool centre point (TCP) at one side and to the allied prismatic joints at another side. Corresponding formal definition of each leg can be presented as PSS, where P and S denote the actuated prismatic joint and the passive spherical joint respectively.

Thus, if the origin of a reference frame is located at the intersection of the prismatic joint axes and the x, y, z-axes are directed along them, the manipulator kinematics may be described by the following equations

$$\mathbf{p} = \begin{bmatrix} (\rho_x + \Delta\rho_x) + \cos\theta_x \cos\beta_x L_x + e \\ \sin\theta_x \cos\beta_x L_x \\ -\sin\beta_x L_x \end{bmatrix}; \qquad (1a)$$

$$\mathbf{p} = \begin{bmatrix} -\sin\beta_y L_y \\ (\rho_y + \Delta\rho_y) + \cos\theta_y \cos\beta_y L_y + e \\ \sin\theta_y \cos\beta_y L_y \end{bmatrix}; \qquad (1b)$$

$$\mathbf{p} = \begin{bmatrix} \sin\theta_z \cos\beta_z L_z \\ -\sin\beta_z L_z \\ (\rho_z + \Delta\rho_z) + \cos\theta_z \cos\beta_z L_z + e \end{bmatrix}, \qquad (1c)$$

where $\mathbf{p} = (p_x, p_y, p_z)^T$ is the output vector of the TCP position, $\boldsymbol{\rho} = (\rho_x, \rho_y, \rho_z)^T$ is the input vector of the prismatic joints variables, $\Delta\boldsymbol{\rho} = (\Delta\rho_x, \Delta\rho_y, \Delta\rho_z)^T$ is the encoder offset vector, θ_i, β_i, $i \in \{x, y, z\}$ are the parallelogram orientation angles (internal variables), and L_i are the length of the corresponding leg.

After elimination of the internal variables θ_i , β_i , the kinematic model (1) can be reduced to three equations

$$\left(p_i - (\rho_i + \Delta\rho_i)\right)^2 + p_j^2 + p_k^2 = L_i^2 , \qquad (2)$$

which includes components of the input and output vectors $\boldsymbol{p}$ and $\boldsymbol{\rho}$ only. Here, the subscripts

$i, j, k \in \{x, y, z\}$, $i \neq j \neq k$ are used in all combinations, and the joint variables ρ_i are obeyed the prescribed limits $\rho_{\min} < \rho_i < \rho_{\max}$ defined in the control software (for the Orthoglide prototype, $\rho_{\min}$ = -100 mm and $\rho_{\max}$ = +60 mm).

It should be noted that, for the case $\Delta\rho_x = \Delta\rho_y = \Delta\rho_z = 0$ and $L_x = L_y = L_z = L$, the nominal ''mechanical-zero'' posture of the manipulator corresponds to the Cartesian coordinates $\mathbf{p}_0 = (0, 0, 0)^T$ and to the joints variables $\rho_0 = (L, L, L)$. Moreover, in such posture, the *x*-, y- and *z*-legs are oriented strictly parallel to the corresponding Cartesian axes. But the joint offsets and the leg length differences cause the deviation of the "zero" TCP location and corresponding deviation of the leg parallelism, which may be measured and used for the calibration.

Hence, six parameters ($\Delta\rho_x$, $\Delta\rho_y$, $\Delta\rho_z$, L_x, L_y, L_z) define the manipulator geometry and are in the focus of the proposed calibration technique.

2.4 Inverse and Direct Kinematics

The *inverse kinematic* relations are derived from the equations (2) in a straightforward way and only slightly differ from the "nominal" case:

$$\rho_i = p_i + s_i \sqrt{L_i^2 - p_j^2 - p_k^2} - \Delta\rho_i\,, \qquad (3)$$

where $s_x, s_y, s_z \in \{\pm 1\}$ are the configuration indices defined for the "nominal" geometry as the signs of $\rho_x - p_x$, $\rho_y - p_y$, $\rho_z - p_z$, respectively. It is obvious that expressions (3) give eight different solutions, however the Orthoglide prototype assembling mode and the joint limits reduce this set to a single case corresponding to the $s_x = s_y = s_z = 1$.

For the direct kinematics, equations (2) can be subtracted pair-to-pair that gives linear relations between the unknowns p_x, p_y, p_z, which may be expressed in the parametric form as

$$p_i = \frac{\rho_i + \Delta\rho_i}{2} + \frac{t}{\rho_i + \Delta\rho_i} - \frac{L_i^2}{2(\rho_i + \Delta\rho_i)}, \qquad (4)$$

where *t* is an auxiliary scalar variable. This reduces the direct kinematics to the solution of a quadratic equation $At^2 + Bt + C = 0$ with the coefficients

$$A = \sum_{i \neq j} (\rho_i + \Delta\rho_i)^2 (\rho_j + \Delta\rho_j)^2;$$

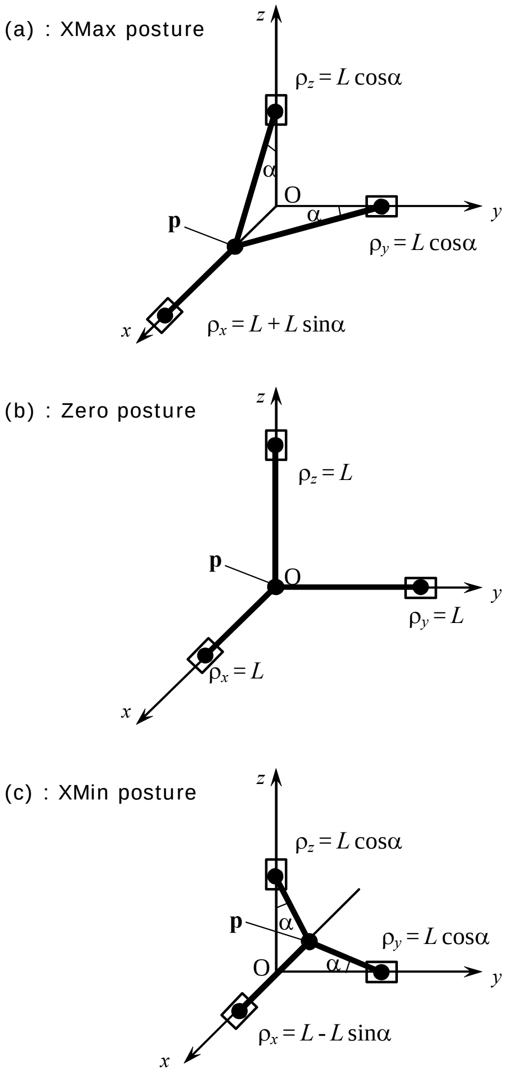


Figure 3: Specific postures of the Orthoglide (for the x-leg motion along the Cartesian axis X).

$$B = \prod_i (\rho_i + \Delta\rho_i)^2 - \sum_{i \neq j \neq k} L_i^2 (\rho_j + \Delta\rho_j)^2 (\rho_k + \Delta\rho_k)^2;$$

$$C = \prod_i (\rho_i + \Delta\rho_i)^2 \cdot \left(\sum_i (\rho_i + \Delta\rho_i)^2 / 4 - \sum_i L_i^2 / 2 \right) + \\ + \sum_{i \neq j \neq k} L_i^4 (\rho_j + \Delta\rho_j)^2 (\rho_k + \Delta\rho_k)^2 / 4$$

where $i, j, k \in \{x, y, z\}$. From two possible solutions that gives the quadratic formula, the Orthoglide prototype (see Figure 1) admit a single one $t = (-B + \sqrt{B^2 - 4AC}) / 2A$ corresponding to the manipulator assembling mode.

2.5 Differential Relations

To obtain the calibration equations, first let us derive the differential relations for the TCP deviation for three types of the Orthoglide postures:

(i) "*maximum displacement*" postures for the directions x, y, z (Figure 3a);

(ii) "*mechanical zero*" or the isotropic posture (Figure 3b);

(iii) "*minimum displacement*" postures for the directions x, y, z (Figure 3c);

These postures are of particular interest for the calibration since, in the "nominal" case, a corresponding leg is parallel to the relevant pair of the Cartesian planes.

The manipulator Jacobian with respect to the parameters $\Delta\boldsymbol{\rho} = (\Delta\rho_x, \Delta\rho_y, \Delta\rho_z)$ and $\boldsymbol{L} = (L_x, L_y, L_z)$ can be derived by straightforward differentiating of the kinematic equations (2), which yields

$$\begin{bmatrix} p_x-\rho_x & p_y & p_z \\ p_x & p_y-\rho_y & p_z \\ p_x & p_y & p_z-\rho_z \end{bmatrix} \cdot \frac{\partial \mathbf{p}}{\partial \boldsymbol{\rho}} = \begin{bmatrix} p_x-\rho_x & 0 & 0 \\ 0 & p_y-\rho_y & 0 \\ 0 & 0 & p_z-\rho_z \end{bmatrix}$$

$$\begin{bmatrix} p_x-\rho_x & p_y & p_z \\ p_x & p_y-\rho_y & p_z \\ p_x & p_y & p_z-\rho_z \end{bmatrix} \cdot \frac{\partial \mathbf{p}}{\partial \mathbf{L}} = \begin{bmatrix} L_x & 0 & 0 \\ 0 & L_y & 0 \\ 0 & 0 & L_z \end{bmatrix}.$$

Thus, after the matrix inversions and multiplications, the desired Jacobian can be written as

$$\mathbf{J}(\mathbf{p}, \boldsymbol{\rho}) = \left[\mathbf{J}_\rho(\mathbf{p}, \boldsymbol{\rho});\ \mathbf{J}_L(\mathbf{p}, \boldsymbol{\rho}) \right], \tag{5}$$

where

$$\mathbf{J}_\rho(.) = \begin{bmatrix} 1 & \dfrac{p_y}{p_x-\rho_x} & \dfrac{p_z}{p_x-\rho_x} \\ \dfrac{p_x}{p_y-\rho_y} & 1 & \dfrac{p_z}{p_y-\rho_y} \\ \dfrac{p_x}{p_z-\rho_z} & \dfrac{p_y}{p_z-\rho_z} & 1 \end{bmatrix}^{-1}$$

$$\mathbf{J}_L(.) = \begin{bmatrix} \dfrac{p_x-\rho_x}{L_x} & \dfrac{p_y}{L_x} & \dfrac{p_z}{L_x} \\ \dfrac{p_x}{L_y} & \dfrac{p_y-\rho_y}{L_y} & \dfrac{p_z}{L_y} \\ \dfrac{p_x}{L_z} & \dfrac{p_y}{L_z} & \dfrac{p_z-\rho_z}{L_z} \end{bmatrix}^{-1}$$

It should be noted that, for the computing convenience, the above expression includes both the Cartesian coordinates p_x, p_y, p_z and the joint coordinates ρ_x, ρ_y, ρ_z, but only one of these sets may be treated as an independent taking into account the inverse/direct kinematic relations.

For the "*Zero*" posture, the differential relations are derived in the neighbourhood of the point $\{\boldsymbol{p}_0 = (0, 0, 0)\,;\ \ \boldsymbol{\rho}_0 = (L, L, L)\}$, which after substitution to (5) gives the Jacobian matrix

$$\mathbf{J}_0 = \left[\begin{array}{ccc:ccc} 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \end{array}\right]. \tag{6}$$

Hence, in this case, the TCP displacement is related to the joint offsets and the leg legs variations ΔL_i by trivial equations

$$\Delta p_i = \Delta\rho_i - \Delta L_i;\ \ i \in \{x, y, z\}. \tag{7}$$

For the "*XMax*" posture, the Jacobian is computed in the neighbourhood of the point $\{\ \mathbf{p} = (LS_\alpha, 0, 0)\,;\ \boldsymbol{\rho} = (L{+}LS_\alpha, LC_\alpha, LC_\alpha)\ \}$, where α is the angle between the *y*-, *z*-legs and the X-axes: $\alpha = a\sin(\rho_{\max}/L)$; $S_\alpha = \sin(\alpha)$, $C_\alpha = \cos(\alpha)$. This gives the Jacobian

$$\mathbf{J}_x^+ = \left[\begin{array}{ccc:ccc} 1 & 0 & 0 & -1 & 0 & 0 \\ T_\alpha & 1 & 0 & -T_\alpha & -C_\alpha^{-1} & 0 \\ T_\alpha & 0 & 1 & -T_\alpha & 0 & -C_\alpha^{-1} \end{array}\right], \tag{8}$$

where $T_\alpha = \tan(\alpha)$. Hence, the desired equations for the TCP displacement may be written as

$$\begin{aligned} \Delta p_x &= \Delta\rho_x - \Delta L_x \\ \Delta p_y &= T_\alpha \Delta\rho_x + \Delta\rho_y - T_\alpha \Delta L_x - C_\alpha^{-1} \Delta L_y \\ \Delta p_z &= T_\alpha \Delta\rho_x + \Delta\rho_z - T_\alpha \Delta L_x - C_\alpha^{-1} \Delta L_z \end{aligned} \tag{9}$$

It can be proved that similar results are valid for the "*YMax*" and "*ZMax*" postures (differing by the indices only), and also for the "*XMin*", "*YMin*", "*ZMin*" postures. In the latter case, the angle α should be computed as $\alpha = a\sin(\rho_{\min}/L)$.

3 CALIBRATION METHOD

3.1 Measurement Technique

To evaluate the leg/surface parallelism, we propose a single-sensor measurement technique. It is based on the fixed location of the measuring device for two distinct leg postures corresponding to the minimum/maximum values of the joint coordinates (Figure 4). Relevant calibration experiment consists of the following steps:

Step 1. Move the manipulator to the "*Zero*" posture; locate two gauges in the middle of the X-leg (parallel to the axes Y and Z); get their readings.

Step 2. Move the manipulator to the "*XMax*" and "*XMin*" postures, get the gauge readings, and compute differences.

Step 3+. Repeat steps 1, 2 for the Y- and Z-legs and compute corresponding differences.

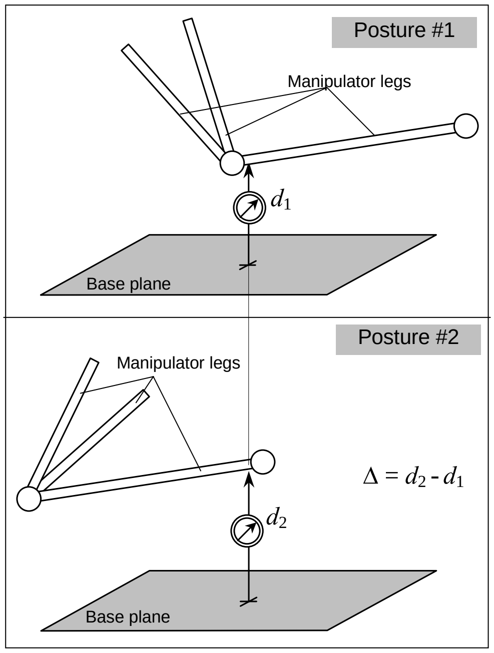


Figure 4: Measuring the leg/surface parallelism.

3.2 Calibration Equations

The system of calibration equations can be derived in two steps. First, it is required to define the gauge initial locations that are assumed to be positioned at the leg middle at the "Zero" posture, i.e. at the points $(\mathbf{p}+\mathbf{r}_i)/2$, $i \in \{x, y, z\}$ where the vectors $\boldsymbol{r}_i$ define the prismatic joints centres: $\mathbf{r}_x = (L+\Delta\rho_x; 0; 0)$; $\mathbf{r}_y = (0; L+\Delta\rho_y; 0)$; $\mathbf{r}_z = (0;\ 0;\ L+\Delta\rho_z)$.

Hence, using the equation (7), the gauge initial locations can be expressed as

$$\mathbf{g}_x^0 = \left(\frac{L-\Delta L_x}{2} + \Delta\rho_x;\quad \frac{\Delta\rho_y - \Delta L_y}{2};\quad \frac{\Delta\rho_z - \Delta L_z}{2}\right)$$

$$\mathbf{g}_y^0 = \left(\frac{\Delta\rho_x - \Delta L_x}{2};\quad \frac{L-\Delta L_y}{2} + \Delta\rho_y;\quad \frac{\Delta\rho_z - \Delta L_z}{2}\right)$$

$$\mathbf{g}_z^0 = \left(\frac{\Delta\rho_x - \Delta L_x}{2};\quad \frac{\Delta\rho_y - \Delta L_y}{2};\quad \frac{L-\Delta L_z}{2} + \Delta\rho_z\right)$$

Afterwards, in the "*XMax*", "*YMax*", "*ZMax*" postures, the leg location is also defined by two points, namely, (i) the TCP, and (ii) the centre of the prismatic joint $\boldsymbol{r}_i$. For example, for the "*XMax*" posture, the TCP position is

$$\mathbf{p}_x^{\max} = (LS_\alpha + \Delta\rho_x - \Delta L_x;\ *;\ *),$$

and the joint position is

$$\mathbf{r}_x^{\max} = (L + LS_\alpha + \Delta\rho_x;\ 0;\ 0).$$

So, the leg is located along the line

$$\mathbf{s}_x(\mu) = \mu \cdot \mathbf{p}_x^{\max} + (1-\mu)\cdot \mathbf{r}_x^{\max},$$

where μ is a scalar parameter, μ∈[0, 1]. Since the x-coordinate of the gauge is independent of the posture, the parameter μ may be obtained from the equation $[\mathbf{s}_x(\mu)]_x = [\mathbf{g}_x^0]_x$, which solution yields:

$$\mu = 0.5 + S_\alpha - S_\alpha \cdot \Delta L_x / L,$$

Hence, after some transformations, the deviations of the X-leg measurements (between the "*XMax*" and "*Zero*" postures) may be expressed as

$$\Delta y_x^+ = (0.5+S_\alpha)T_\alpha\ \Delta\rho_x + S_\alpha \Delta\rho_y - \\ -(0.5+S_\alpha)T_\alpha \Delta L_x - ((0.5+S_\alpha)C_\alpha^{-1} - 0.5)\Delta L_y$$

$$\Delta z_x^+ = (0.5+S_\alpha)T_\alpha\ \Delta\rho_x + S_\alpha \Delta\rho_z - \\ -(0.5+S_\alpha)T_\alpha \Delta L_x - ((0.5+S_\alpha)C_\alpha^{-1} - 0.5)\Delta L_z$$

Similar approach may be applied to the "*XMin*" posture, as well as to the corresponding postures for the Y- and Z-legs. This gives the system of twelve linear equations in six unknowns:

$$\begin{bmatrix} a_1 & b_1 & 0 & -c_1 & -b_1 & 0 \\ b_1 & a_1 & 0 & -b_1 & -c_1 & 0 \\ a_2 & b_2 & 0 & -c_2 & -b_2 & 0 \\ b_2 & a_2 & 0 & -b_2 & -c_2 & 0 \\ \hline 0 & a_1 & b_1 & 0 & -c_1 & -b_1 \\ 0 & b_1 & a_1 & 0 & -b_1 & -c_1 \\ 0 & a_2 & b_2 & 0 & -c_2 & -b_2 \\ 0 & b_2 & a_2 & 0 & -b_2 & -c_2 \\ \hline a_1 & 0 & b_1 & -c_1 & 0 & -b_1 \\ b_1 & 0 & a_1 & -b_1 & 0 & -c_1 \\ a_2 & 0 & b_2 & -c_2 & 0 & -b_2 \\ b_2 & 0 & a_2 & -b_2 & 0 & -c_2 \end{bmatrix} \begin{bmatrix} \Delta\rho_x \\ \Delta\rho_y \\ \Delta\rho_z \\ \Delta L_x \\ \Delta L_y \\ \Delta L_z \end{bmatrix} = \begin{bmatrix} \Delta x_y^+ \\ \Delta y_x^+ \\ \Delta x_y^- \\ \Delta y_x^- \\ \hline \Delta y_z^+ \\ \Delta z_y^+ \\ \Delta y_z^- \\ \Delta z_y^- \\ \hline \Delta x_z^+ \\ \Delta z_x^+ \\ \Delta x_z^- \\ \Delta z_x^- \end{bmatrix} \qquad (10)$$

where
$a_i = S_{\alpha_i}$, $b_i = (0.5 + S_{\alpha_i})T_{\alpha_i}$, $c_i = (0.5 + S_{\alpha_i})C_{\alpha_i}^{-1} - 0.5$
and $\alpha_1 = \operatorname{asin}(\rho_{\max}/L) > 0$, $\alpha_2 = \operatorname{asin}(\rho_{\min}/L) < 0$.

This system can be solved using the pseudoinverse of Moore-Penrose, which ensures the minimum of the residual square sum.

4 EXPERIMENTAL RESULTS

The measuring system is composed of standard comparator indicators attached to the universal magnetic stands allowing fixing them on the manipulator bases. The indicators have resolution of 10 μm and are sequentially used for measuring the X-, Y-, and Z-leg parallelism while the manipulator moves between the *Max*, *Min* and *Zero* postures. For each measurement, the indicators are located on the mechanism base in such manner that a corresponding leg is admissible for the gauge contact for all intermediate postures (Figure 5).

For each leg, the measurements were repeated three times for the following sequence of motions: Zero → Max → Min → Zero→ …. Then, the results were averaged and used for the parameter identification (the repeatability of the measurements is about 0.02 mm).

To validate the developed calibration technique and the adopted modelling assumptions, there were carried out three experiments targeted to the following objectives: (#1) validation of modelling assumptions; (#2) collecting the experimental data for the parameter identification; and (#3) verification of the calibration results.

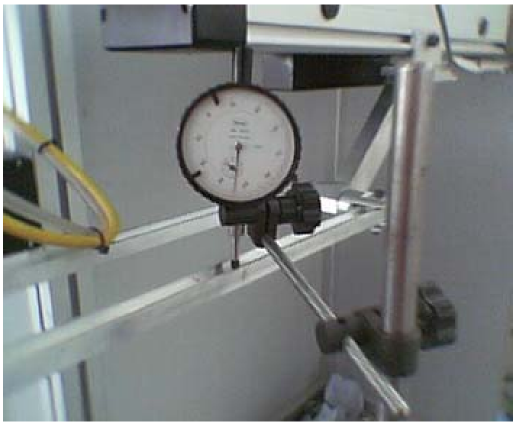


Figure 5: Experimental Setup.

Table 1: Calibration results.

Parameters (mm)						R.m.s. (mm)
$\Delta\rho_x$	$\Delta\rho_y$	$\Delta\rho_z$	ΔL_x	ΔL_y	ΔL_z	
4.66	-5.36	1.46	5.20	-5.96	3.16	0.12
-0.48	0.49	-1.67	–	–	–	0.14
–	–	–	0.50	-0.52	1.69	0.14

The *first experiment* produced rather high parallelism deviation, which impels to conclude that the mechanism mechanics requires more careful tuning. Consequently, the location of the joint axes was adjusted mechanically to ensure the leg parallelism for the Zero posture.

The *second experiment* (after mechanical tuning) yielded lower deviations, twice better than for the first experiment. For these data, the developed calibration algorithm was applied for three sets of the model parameters: for the full set {Δρ, Δ***L***} and for the reduced sets {Δρ} and {Δ***L***}. As follows from the identification results (Table 1), the algorithms is able to identify simultaneously both the joint offsets and Δρ and the link lengths Δ***L***. However, both Δρ and Δ***L*** (separately) demonstrate roughly the same influence on the residual reduction, from 0.32 mm to 0.14 mm, while the full set {Δρ, Δ***L***} gives further residual reduction to the 0.12 mm only. This motivates considering Δρ as the most essential parameters to be calibrated. Accordingly, the identified vales of the joint offsets were input into the control software.

The *third experiment* demonstrated good agreement with the expected results. In particular, the average deviation reduced down to 0.15 mm, which corresponds to the measurement accuracy. On the other hand, further adjusting of the model to the new experimental data does not give the residual reduction.

Hence, the calibration results confirm validity of the proposed identification technique and its ability to tune the joint offsets and link lengths from observations of the leg parallelism. Other conclusion is related to the modelling assumption: for further accuracy improvement it is prudent to generalize the manipulator model by including parameters describing the orientation of the prismatic joint axes, i.e. relaxing assumption (iv) (see sub-section 2.2).

5 CONCLUSIONS

This paper proposes further developments for a novel calibration approach for parallel manipulators, which is based on observations of manipulator leg parallelism with respect to some predefined planes. This technique employs a simple and low-cost measuring system composed of standard comparator indicators, which are sequentially used for measuring the deviation of the relevant leg location while the manipulator moves the TCP along the Cartesian axes. From the measured differences, the calibration algorithm estimates the joint offsets and the link lengths that are treated as the most essential parameters to be tuned. The validity of the proposed approach and efficiency of the developed numerical algorithm were confirmed by the calibration experiments with the Orthoglide prototype, which allowed essential reduction of the residuals and corresponding improvement of the accuracy.

Future work will focus on the expanding the set of the identified model parameters, their identifiably analysis, and compensation of the non-geometric errors.

REFERENCES

Chablat, D., Wenger, Ph., 2003. Architecture Optimization of a 3-DOF Parallel Mechanism for Machining Applications, the Orthoglide. *IEEE Transactions On Robotics and Automation*, Vol. 19 (3), pp. 403-410.

Daney, D., 2003. Kinematic Calibration of the Gough platform. *Robotica*, 21(6), pp. 677-690.

Huang, T., Chetwynd, D.G. Whitehouse, D.J., Wang, J., 2005. A general and novel approach for parameter identification of 6-dof parallel kinematic machines. *Mechanism and Machine Theory*, Vol. 40 (2), pp. 219-239.

Innocenti, C., 1995. Algorithms for kinematic calibration of fully-parallel manipulators. In: *Computational Kinematics*, Kluwer Academic Publishers, pp. 241-250.

Iurascu, C.C. Park, F.C., 2003. Geometric algorithm for kinematic calibration of robots containing closed loops. *ASME Journal of Mechanical Design*, Vol. 125(1), pp. 23-32.

Jeong J., Kang, D., Cho, Y.M., Kim, J., 2004. Kinematic calibration of redundantly actuated parallel mechanisms. *ASME Journal of Mechanical Design*, Vol. 126 (2), pp. 307-318.

Merlet, J.-P., 2000. *Parallel Robots*, Kluwer Academic Publishers, Dordrecht, 2000.

Pashkevich, A., Wenger, P., Chablat, D., 2005. Design Strategies for the Geometric Synthesis of Orthoglide-type Mechanisms. *Mechanism and Machine Theory*, Vol. 40 (8), pp. 907-930.

Pashkevich A., Chablat D., Wenger P., 2006. Kinematic Calibration of Orthoglide-Type Mechanisms. *Proceedings of IFAC Symposium on Information Control Problems in Manufacturing* (INCOM'2006), Saint Etienne, France, 17-19 May, 2006, p. 151 - 156

Renaud, P., Andreff, N., Pierrot, F., Martinet, P., 2004. Combining end-effector and legs observation for kinematic calibration of parallel mechanisms. *IEEE International Conference on Robotics and Automation* (ICRA'2004), New-Orleans, USA, pp. 4116-4121.

Renaud, P., Andreff, N., Martinet, P., Gogu, G., 2005. Kinematic calibration of parallel mechanisms: a novel approach using legs observation. *IEEE Transactions on Robotics* and Automation, 21 (4), pp. 529-538.

Tlusty, J., Ziegert, J.C., Ridgeway, S., 1999. Fundamental Comparison of the Use of Serial and Parallel Kinematics for Machine Tools. *CIRP Annals*, Vol. 48 (1), pp. 351-356.

Wang, J. Masory, O. 1993. On the accuracy of a Stewart platform - Part I: The effect of manufacturing tolerances. *IEEE International Conference on Robotics and Automation* (ICRA'93), Atlanta, Georgia, pp. 114–120.

Wenger, P., Gosselin, C., Chablat, D., 2001. Comparative study of parallel kinematic architectures for machining applications. In: Workshop on Computational Kinematics, Seoul, Korea, pp. 249-258.

Wenger, P., Gosselin, C. Maille, B., 1999. A comparative study of serial and parallel mechanism topologies for machine tools. In: *Proceedings of PKM'99*, Milan, Italy, pp. 23–32.

MAKING SENSOR NETWORKS INTELLIGENT

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

Masanori Sugisaka
Department of Electrical and Electronic Engineering, Oita University
700 Oaza Dannoharu 870-1192 Japan
Tel: 097-554-7831, Fax: 097-554-7841
msugi@cc.oita-u.ac.jp

Joaquim Filipe
Departamento Sistemas e Informática, Escola Superior de Tecnologia de Setúbal
Setúbal 2910-761, Portugal
j.filipe@est.ips.pt

Keywords: Sensor networks, intelligent management, distributed scenario language, distributed interpreter, tracking objects, hierarchical data fusion.

Abstract: A universal solution for management of dynamic sensor networks will be presented, covering both networking and application layers. A network of intelligent modules, overlaying the sensor network, collectively interprets mission scenarios in a special high-level language that can start from any nodes and cover the network at runtime. The spreading scenarios are extremely compact, which may be useful for energy saving communications. The code will be exhibited for distributed collection and fusion of sensor data, also for tracking mobile targets by scattered and communicating sensors.

1 INTRODUCTION

Sensor networks are a sensing, computing and communication infrastructure that allows us to instrument, observe, and respond to phenomena in the natural environment, and in our physical and cyber infrastructure (Culler at al., 2004; Chong, Kumar, 2003). The sensors themselves can range from small passive microsensors to larger scale, controllable platforms. Their computation and communication infrastructure will be radically different from that found in today's Internet-based systems, reflecting the device- and application-driven nature of these systems.

Of particular interest are wireless sensor networks, WSN (Wireless; Zhao, Guibas, 2004) consisting of spatially distributed autonomous devices using sensors to cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants, at different locations. WSN, however, have many additional problems in comparison to the wired ones. The individual devices in WSN are inherently resource constrained--they have limited processing speed, storage capacity, and communication bandwidth. These devices have substantial processing capability in the aggregate, but not individually, so we must combine their many vantage points on the physical phenomena within the network itself. In addition to one or more sensors, each node in a sensor network is typically equipped with a radio transceiver or other wireless communications device, a small microcontroller, and an energy source, usually a battery. The size of a single sensor node can vary from shoebox-sized nodes down to devices the size of grain of dust.

Typical applications of WSNs include monitoring, tracking, and controlling. Some of the specific applications are habitat monitoring, object tracking, nuclear reactor controlling, fire detection, traffic monitoring, etc. In a typical application, a WSN is scattered in a region where it is meant to collect data through its sensor nodes. They could be deployed in wilderness areas, where they would remain for many years (monitoring some environmental variable) without the need to recharge/replace their power supplies. They could form a perimeter about a property and monitor the progression of intruders (passing information from one node to the next). At present, there are many uses for WSNs throughout the world.

In a wired network like the Internet, each router connects to a specific set of other routers, forming a routing graph. In WSNs, each node has a radio that provides a set of communication links to nearby nodes. By exchanging information, nodes can discover their neighbors and perform a distributed algorithm to determine how to route data according to the application's needs. Although physical placement primarily determines connectivity, variables such as obstructions, interference, environmental factors, antenna orientation, and mobility make determining connectivity a priori difficult. Instead, the network discovers and adapts to whatever connectivity is present.

Fig. 1 shows what we will mean as a sensor network for the rest of this paper.

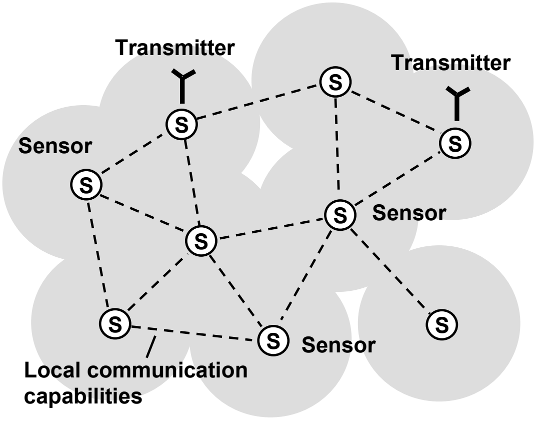


Figure 1: Distributed sensors and their emergent network.

It will hypothetically consist of (a great number of) usual sensors with local communication capabilities, and (a limited number of) those that can additionally transmit collected information outside the area (say, via satellite channels). Individual sensors can be on a move, some may be destroyed while others added at runtime (say, dropped from the air) to join the existing ones in solving cooperatively distributed problems.

The aim of this paper is to show how any imaginable (or even so far unimaginable) distributed problems can be solved by dynamic self-organized sensor networks if to increase their intelligence as a whole, with a novel distributed processing and control ideology and technology effectively operating in computer networks.

2 THE DISTRIBUTED MANAGEMENT MODEL

The distributed information technology we are using here is based on a special Distributed Scenario Language (DSL) describing parallel solutions in computer networks as a seamless spatial process rather than the traditional collection and interaction of parts (agents). Parallel scenarios in DSL can start from any interpreter of the language, spreading and covering the distributed space at runtime, as in Fig. 2.

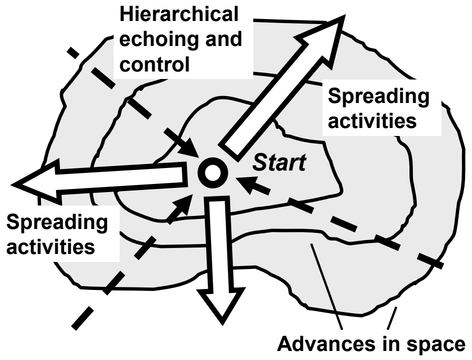


Figure 2: Runtime coverage of space by parallel scenarios.

The overall management of the evolving scenarios is accomplished via the distributed track system providing hierarchical command and control for scenario execution, with a variety of special echo messages. We will mention here only key features of DSL, as the current language details can be found elsewhere (Sapaty et al., 2007), also its basics from the previous versions (Sapaty, 1999, 2005; Sapaty et al., 2006).

A DSL program, or *wave*, is represented as one or more constructs called *moves* (separated by a comma) embraced by a *rule*, as follows:

wave → *rule* ({ *move* , })

Rules may serve as various supervisory, regulatory, coordinating, integrating, navigating, and data processing functions, operations or constraints over moves.

A move can be a *constant* or *variable*, or recursively an arbitrary wave itself:

move → *constant* | *variable* | *wave*

Variables classify as *nodal*, associated with space positions and shared by different waves, *frontal*, moving in space with program control, and *environmental*, accessing the environment navigated. Constants may reflect both information and physical matter.

Wave, being applied in a certain position of the distributed world, can perform proper actions in a distributed space, terminating in the same or in other positions. It provides final result that unites local results in the positions (nodes) reached, and also produces resultant control state. The (distributed) result and the state can be subsequently used for further data processing and decision making on higher program levels. Parallel waves can start from different nodes in parallel, possibly intersecting in the common distributed space when evolving in it independently.

If moves are ordered to advance in space one after the other (which is defined by a proper rule), each new move is applied in parallel in all the nodes reached by the previous move. Different moves (by other rules) can also apply independently from the same node, reaching new nodes in parallel. The functional style syntax shown above can express any program in DSL, but if useful, other notations can be used, like the infix one. For example, an advancement in space can use period as operator (separator) between successive steps, whereas parallel actions starting from same node can be separated by a semicolon. For improving readability, spaces can be inserted in any places of the programs--they will be automatically removed before execution (except when embraced by quotes).

The interpreter may have its own physical body (say, in the form of mobile or humanoid robot), or can be mounted on humans (like in mobile phones). A network of the interpreters can be mobile and open, changing its volume and structure, as robots or humans can move at runtime. We will be assuming for the rest of this paper that every sensor has the DSL interpreter installed, which may have a software implementation or can be a special hardware chip.

In the following sections we will show and explain the DSL code for a number of important problems to be solved by advanced sensor networks, which confirms an efficiency of the proposed distributed computational and control model.

3 ELEMENTARY EXAMPLE

3.1 The Task

An elementary task to be programmed in DSL may look like follows. *Let it needs to go to the physical locations of a disaster zone with coordinates (using x-y pair here for simplicity) x25_y12, x40_y36, and x55_y21, measure temperature there, and transmit its value, together with exact coordinates of the locations reached, to a collection center.* The corresponding program in DSL will be as follows:

```
Hop(x25_y12, x40_y36, x55_y21).
Transmit(Temperature & Location)
```

The program moves independently to the three locations given, and in every destination reached measures temperature using special environmental variable `Temperature`. Using another environmental variable `Location`, it attaches to the previous value exact coordinates of the current physical position (which, by using GPS, may differ from the initially intended, rough coordinates). The two-value results are then independently transmitted from the three locations to a collection center.

This program reflects semantics of the task to be performed in a distributed space, regardless of possible equipment that can be used for this. The latter may, for example, be a set of sensors scattered in advance throughout the disaster zone, where hopping by coordinates may result in a wireless access of the sensors already present there--not necessarily moving into these points physically. As another solution, this program may task mobile robots (single or a group) to move into these locations in person and perform the needed measurement and transmission upon reaching the destinations.

3.2 Single-Robot Solution

Let us consider how the previous program will be executed with only a single robot available (let it be Robot 1, assuming other robots not reachable). After its injection into the robot's interpreter (see Fig. 3), the first, broadcasting statement:

```
Hop(x25_y12, x40_y36, x55_y21)
```

will be analyzed first. It naturally splits into three independent hops, but only one can be performed at the start by a single robot.

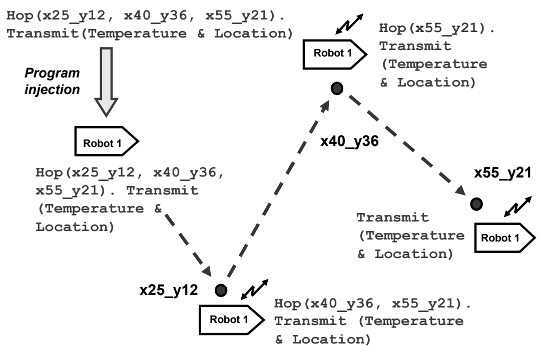

Hop(x25_y12, x40_y36, x55_y21).
Transmit(Temperature & Location)
Program injection
Robot 1
Hop(x25_y12, x40_y36, x55_y21). Transmit (Temperature & Location)
x25_y12
Robot 1
Hop(x40_y36, x55_y21). Transmit (Temperature & Location)
x40_y36
Robot 1
Hop(x55_y21). Transmit (Temperature & Location)
x55_y21
Transmit (Temperature & Location)
Robot 1

Figure 3: Single-robot solution.

The interpreter hops virtually into the point `x25_y12` ordering robot to move into the corresponding physical location. Upon arrival, the second statement:

```
Transmit(Temperature & Location)
```

will be executed by measuring temperature, attaching coordinates, and transmitting the result via channels available. The rest of the program, represented as:

```
Hop(x40_y36, x55_y21).
Transmit(Temperature & Location)
```

will be analyzed again, with hop to `x40_y36` extracted, robot moved into the second location, and measurement result transmitted as before. The program's remainder, now as:

```
Hop(x55_y21).
Transmit(Temperature & Location)
```

will cause movement and operation in the third location `x55_y21`. The program terminates after shrinking to:

```
Transmit(Temperature & Location)
```

All these steps are depicted detail in Fig. 3.

3.3 Multiple-Robot Solution

Let us consider now the case where other robots can be requested and accessed from the robot into which we injected our program (let it be Robot 1 again), see Fig. 4. After analyzing the first statement, splitting it into individual hops and attaching to each of them the replicated rest of the program (i.e. second statement) the interpreter in Robot 1 will produce the following three independent programs:

```
Hop(x25_y12).
Transmit(Temperature & Location)

Hop(x40_y36).
Transmit(Temperature & Location)

Hop(x55_y21).
Transmit(Temperature & Location)
```

Leaving one of them (say, the first) for execution in itself, Robot 1 requesting other available robots (let them be Robot 2 and Robot 3) by a wireless channel sends electronically the obtained other two programs to them. After this, all three programs will be operating in the three mentioned robots independently and in parallel.

Each robot first executes the hop statement, moving into the location given by physical coordinates, and upon reaching the destination, executes the second statement measuring temperature and detecting exact coordinates, subsequently transmitting these to the collection center, exactly as for the previous case with a single robot. All this is depicted in detail in Fig. 4.

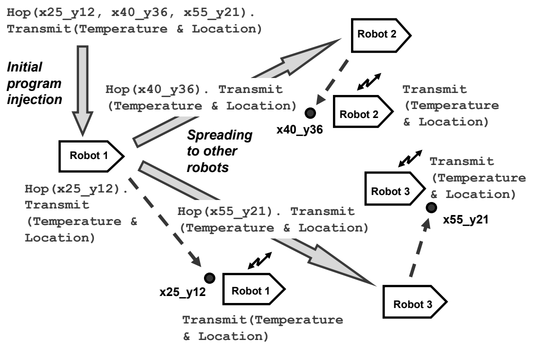

Hop(x25_y12, x40_y36, x55_y21).
Transmit(Temperature & Location)
Initial program injection
Robot 1
Hop(x40_y36). Transmit (Temperature & Location)
Spreading to other robots
Robot 2
x40_y36
Robot 2
Transmit (Temperature & Location)
Hop(x25_y12). Transmit (Temperature & Location)
Hop(x55_y21). Transmit (Temperature & Location)
Robot 3
Transmit (Temperature & Location)
x55_y21
x25_y12
Robot 1
Transmit(Temperature & Location)
Robot 3

Figure 4: Multiple-robot solution.

The shown above were only elementary examples of DSL code and rules of its execution. In other, more complex cases DSL allows us to dynamically form networked knowledge arbitrarily distributed between dynamic resources, also providing hierarchical control of distributed processes. DSL can provide description of system missions on a semantic level, with telling what, where, and why to do rather than how and who should do this, effectively abstracting from the resources (computer networks, robots, humans) that can implement them, which can be emergent and not known in advance. These programs can be executed by any number of technical or biological processors, where organization of resources, their communication and synchronization are delegated to efficient automatic interpretation, with strict and simple implementation rules.

Application programs in DSL are often hundreds of times shorter than in conventional languages (like C or Java). The DSL programs are interpreted rather than compiled, adjusting to the execution environment at runtime, which is especially important when system components may be changed or damaged during the execution. The approach is based on the new, universal, formal system within which any distributed and parallel algorithms can be expressed and implemented.

4 COLLECTING EVENTS THROUGHOUT THE REGION

Starting from all transmitter nodes, the following program regularly (with interval of 20 sec.) covers stepwise, via local communications between sensors, the whole sensor network with a spanning forest, lifting information about observable events in each node reached. Through this forest, by the internal interpretation infrastructure, the lifted data in nodes is moved and fused upwards the spanning trees, with final results collected in transmitter nodes and sent in parallel outside the system using rule `Transmit` (See Fig. 5).

```
Hop(all transmitters).
Loop(
 Sleep(20).
 IDENTITY = TIME.
 Transmit(
  Fuse(
   Repeat(free(observe(events));
    Hop(directly reachable,
        first come)))))
```

Globally looping in each transmitter node (rule `loop`), the program repeatedly navigates (rule `repeat`) the sensor set (possibly, in competition with navigation started from other transmitters), activating local space observation facilities in parallel with the further navigation.

The resultant forest-like coverage is guaranteed by allowing sensor nodes to be visited only once, on the first arrival in them. The hierarchical fusion rule `fuse`, collecting the scattered results, also removes record duplicates, as the same event can be detected by different sensors, leaving only most credible in the final result. To distinguish each new global navigation process from the previous one, it always spreads with a new identity for which, for example, current system time may be used (using environmental variables `IDENTITY` and `TIME` of the language).

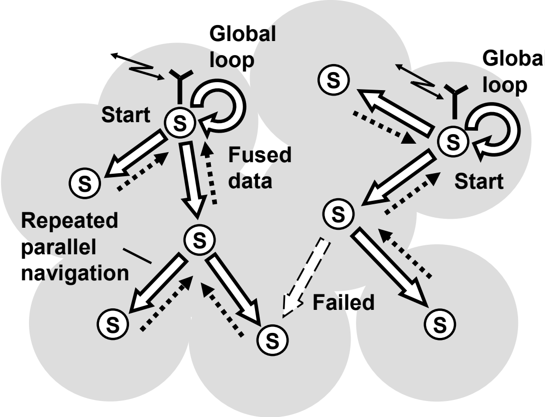


Figure 5: Parallel navigation and data collection.

5 CREATING HIERARCHICAL INFRASTRUCTURES

In the previous program, we created the whole spanning forest for each global data collection loop, which may be costly. To optimize this process, we may first create a persistent forest infrastructure, remembering which nodes were linked to which, and then use it for a frequent regular collection and fusion of the scattered data. As the sensor neighborhood network may change over time, we can make this persistent infrastructure changeable too, updating it with some time interval (much larger, however, than the data collection one), after removing the previous infrastructure version. This can be done by the following program that regularly creates top-down oriented links named `infra` starting from the transmitter nodes (as shown in Fig. 6).

```
Hop(all transmitters).
Loop(
 Sleep(120).
 IDENTITY = TIME.
 Repeat(
  Hop(directly reachable,
    first come).
  Remove links(all).
  Stay(create link(-infra, BACK)))
```

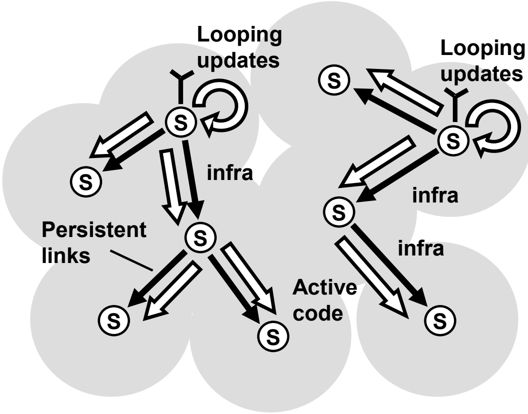


Figure 6: Runtime creation of hierarchical infrastructure.

This infrastructure creation program provides competitive asynchronous spatial processes, so each time even if the sensors do not change their positions, the resultant infrastructure may differ, as in Fig. 7.

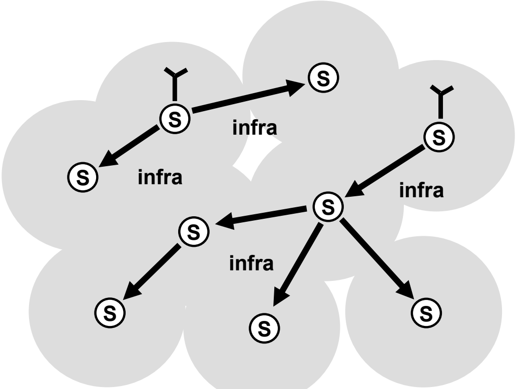


Figure 7: Another possible infrastructure.

Having created a persistent infrastructure, we can use it frequently by the event collection program, which can be simplified now as follows:

```
Hop(all transmitters).
Loop(
 Sleep(20).
 Transmit(
  Fuse(
   Repeat(
    Free(observe(events));
    Hop(+infra)))))
```

The global infrastructure creation program (looping rarely) and the event collection and fusion one (looping frequently) can operate simultaneously, with the first one guiding the latter on the data collection routes, which may change over time.

6 ROUTING LOCAL EVENTS TO TRANSMITTERS

We have considered above the collection of distributed events in the top-down and bottom-up mode, always with the initiative stemming from root nodes of the hierarchy--the latter serving as a parallel and distributed tree-structured computer. In this section, we will show quite an opposite, fully distributed solution, where each sensor node, being an initiator itself, is regularly observing the vicinity for the case an event of interest might occur.

Having discovered the event of interest, each node independently from others launches a spatial cyclic self-routing process via the infrastructure links built before, which eventually reaches the transmitter node, bringing with it the event information, as shown in Fig. 8. The data brought to the transmitters should be fused with the data already existing there.

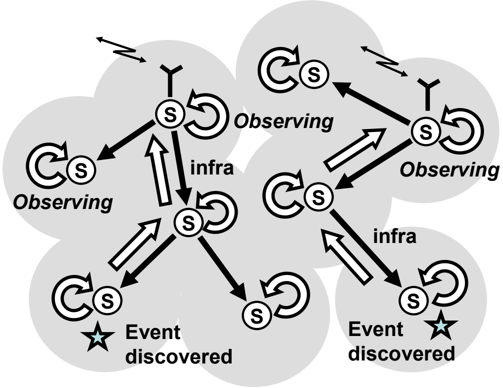


Figure 8: Routing scattered events to transmitters.

The corresponding program will be as follows.

```
Hop(all nodes).
Frontal(Transfer).
Nodal(Result).
Loop(
 Sleep(5).
 Nonempty(
  Transfer = observe(events)).
 Repeat(hop(-infra)).
 Result = Result & Transfer)
```

The transmitter nodes accumulating and fusing local events, arriving from sensor nodes independently, can subsequently send them outside the system. Different strategies can be used here. For example, one could be waiting until there are enough event records collected in the transmitter before sending them, and the other one waiting for a threshold time and only then sending what was accumulated (if any at all). The following program combines these two cases within one solution, where arriving data from sensors is accumulated in nodal variable `Result`.

```
Hop(all transmitters).
Loop(
 Or(
  Quantity(Result) >= 100,
  (Sleep(60). Result != nil)).
 Transmit(Result))
```

This program in every transmitter can work in parallel with the previous program collecting events and looping in every sensor (in transmitters as well, assumed to be sensors too), and also with the earlier program, starting in transmitters, for the regular infrastructure updates.

7 TRACKING MOBILE OBJECTS

Let us consider some basics of using DSL for tracking mobile (say ground or aerial) objects moving through a region controlled by communicating sensors, as shown in Fig. 9. Each sensor can handle only a limited part of space, so to keep the whole observation continuous the object seen should be handed over between the neighboring sensors during its movement, along with the data accumulated during its tracking and analysis.

The space-navigating power of the model discussed can catch each object and accompany it individually, moving between the interpreters in different sensors, thus following the movement in physical space via the virtual space (Sapaty, 1999). This allows us to have an extremely compact and integral solution unattainable by other approaches based on communicating agents. The following program, starting in all sensors, catches the object it sees and follows it wherever it goes, if not seen from the current point any more (more correctly: if its visibility becomes lower than a given threshold).

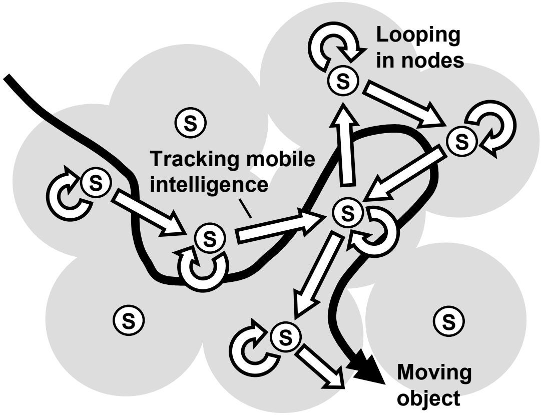


Figure 9: Active tracking of a mobile object.

```
Hop(all nodes).
Frontal(Threshold) = 0.1.
Frontal(Object) = search(aerial).
Visibility(Object) > Threshold.
Repeat(
 Loop(
  Visibility(Object) > Threshold).
 Maximum destination(
  Hop(directly reachable).
  Visibility(Object) > Threshold))
```

The program investigates the object's visibility in all neighboring sensors in parallel and moves control along with program code and accumulated data to the neighboring sensor seeing it best (if visibility there exceeds the threshold given).

This was only a skeleton program in DSL, showing the space tracing techniques for controlling single physical objects. It can be extended to follow collectively behaving groups of physical objects (say, flocks of animals, mobile robots, or troops). The spreading individual intelligences can cooperate in the distributed sensor space, self-optimizing jointly for the pursuit of global goals.

8 AVERAGING PARAMETERS FROM A REGION

Let us consider how it could be possible to asses the generalized situation in a distributed region given, say, by a set of its border coordinates, in a fully distributed way where sensors located in the region communicate with direct neighbors only. Assume, for example, that the data of interest is maximum pollution level throughout the whole region (it may also be temperature, pressure, radiation level, etc.) together with coordinates of the location showing this maximum.

The following program, starting in all sensors located in the region, regularly measures the pollution level in its vicinity, updates local maximum and, by communication with direct neighbors, attempts to increase the recorded maximum there too. Eventually, after some time of this purely local communication activity all sensors will have the same maximum value registered in them and corresponding to the maximum on the whole region (see the overall organization in Fig. 10).

```
Nodal(Level, Max, Region).
Frontal(Transfer).
Region = region definition.
Hop(all nodes, Region).
Loop(
 Or parallel(
  Loop(
   Sleep(5).
   Level = measure(pollution).
   Stay(Level > Max. Max=Level).
   Transfer = Max.
   Hop(directly reachable,Region).
   Transfer > Max. Max=Transfer),
  sleep(120)).
 Level == Max.
 Transfer = Max & WHERE.
 Repeat(hop(-infra)).
 Transmit(Transfer))
```

As there may be many sensors located in the region of interest, we will need forwarding only a single copy of this resultant maximum value to a transmitter for an output. This can be achieved by delegating this task only to the sensor whose measured local value is equal to the accumulated maximum in it, which corresponds to the overall region's maximum.

Having discovered that it is the leader (after a certain time delay), such a sensor organizes repeated movement to the nearest transmitter via the earlier created virtual infrastructure, carrying the resultant maximum value in frontal variable `Transfer`, and sending it outside the system in the transmitter reached, as shown in Fig. 10.

Similar, fully distributed, organization may be introduced for finding averaged values, or even for assembling the global picture of the whole region with any details collected by individual sensors (the latter may be costly, however, with a more practical solution skeleton shown in the next section).

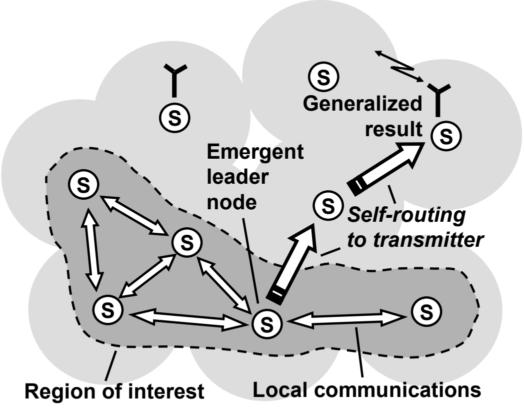


Figure 10: Distributed averaging with active routing.

9 ASSEMBLING FULL PICTURE OF THE REGION

To collect details from some region via local sensors and merge them into the whole picture could, in principle, be possible via local single-level exchanges only, as in the previous section, but the amount of communications and data transfer as well as time needed may be unacceptably high. We were finding only a single value (maximum) via frequent internode communications, with minimal code length.

But for obtaining the detailed global picture of the region or of some distributed phenomenon, we may have to gradually paint (grow) this picture in every sensor node simultaneously, with high communication intensity between the nodes. Also, there may be difficult to determine the completeness of this picture staying in local sensors only. A higher integrity and hierarchical process structuring may be needed to see a distributed picture via the dispersed sensors with limited visual capabilities and casual communications.

Different higher-level approaches can be proposed and expressed in DSL for this. We will show only a possible skeleton with spanning tree coverage of the distributed phenomenon and hierarchical collection, merging, and fusing partial results into the global picture. The latter will be forwarded to the nearest transmitter via the previously created infrastructure (using links `infra`), as in Fig. 11.

```
Hop(random, all nodes,
    detected(phenomenon)).
Loop(
 Frontal(Full) = fuse(
   Repeat(
     Free(collect(phenomenon));
     Hop(directly reachable,
         first come,
         detected(phenomenon)))).
 Repeat(hop(-infra)).
 Transmit(Full))
```

In the more complex situations, which can be effectively programmed in DSL too, we may have a number of simultaneously existing phenomena, which can intersect in a distributed space. We may also face combined phenomena integrating features of different ones. The phenomena (like flocks of birds, manned or unmanned groups or armies, spreading fire or flooding) covering certain regions may change in size and shape, they may also move as a whole preserving internal organization, etc.

In the previous language versions (Sapaty, 1999, 2005; Sapaty et al., 2006), a variety of complex topological problems in computer networks were investigated and successfully programmed in a fully distributed and parallel manner, which included connectivity, graph patterns matching, weak and strong components like articulation points and cliques, also diameter and radius, optimum routing tables, etc., as well as topological self-recovery after indiscriminate damages (Sapaty, 1999).

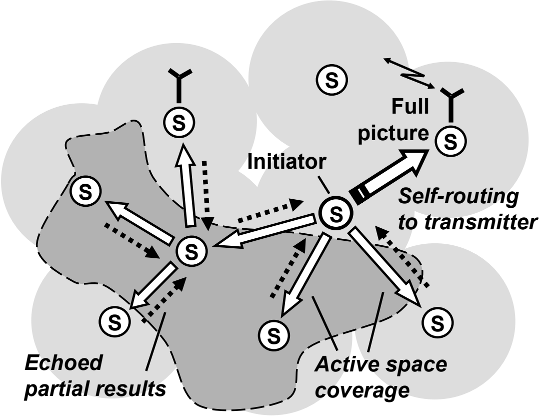


Figure 11: Space coverage with global picture fusion.

10 CONCLUSIONS

We have presented a universal and flexible approach of how to convert distributed sensor networks with limited resources in nodes and casual communications into a *universal spatial machine* capable of not only collecting and forwarding data but also solving complex computational and logical problems as well as making autonomous decisions in distributed environments.

The approach is based on quite a different type of high-level language allowing us to represent system solutions in the form of *integral seamless spatial processes* navigating and covering distributed worlds at runtime. This makes parallel and distributed application programs extremely short, which may be especially useful for the energy saving communications between sensors.

The code compactness and simplicity are achieved because most of traditional synchronization and data or agent exchanges (which are also on a high level, with minimum code sent) are shifted to efficient automatic implementation, allowing us concentrate on *global strategies and global solutions* instead.

REFERENCES

Chong, C.-Y., Kumar, S. P., 2003. Sensor networks: Evolution, opportunities, and challenges. *Proc. of the IEEE*, Vol. 91, No. 8, August, pp.1247-1256.

Culler, D., Estrin, D., Srivastava, M., 2004. Overview of sensor networks, *Computer,* August, pp.41-49, publ. by the IEEE Computer Society.

Sapaty, P. S., 1999. *Mobile Processing in Distributed and Open Environments*, John Wiley & Sons, ISBN: 0471195723, New York, February, 436p. (www.amazon.com).

Sapaty, P. S., 2005. *Ruling Distributed Dynamic Worlds*, John Wiley & Sons, New York, May, 256p, ISBN 0-471-65575-9 (www.amazon.com).

Sapaty, P., Sugisaka, M., Finkelstein, R., Delgado-Frias, J., Mirenkov, N., 2006. Advanced IT support of crisis relief missions. *Journal of Emergency Management*, Vol.4, No.4, ISSN 1543-5865, July/August, pp.29-36 (www.emergencyjournal.com).

Sapaty, P., Morozov, A., Finkelstein, R., Sugisaka, M., Lambert, D., 2007. A new concept of flexible organization for distributed robotized systems. *Proc. Twelfth International Symposium on Artificial Life and Robotics (AROB 12th'07)*, Beppu, Japan, Jan 25-27, 8p.

Wireless sensor network. *Wikipedia, the free encyclopedia*, www.wikipedia.org.

Zhao, F., Guibas, L., 2004. *Wireless Sensor Networks: An Information Processing Approach (The Morgan Kaufmann Series in Networking)*. Morgan Kaufmann, 376p.

BAYES-BASED OBJECT TRACKING BOOSTED BY PARTICLE SWARM OPTIMIZATION

Yuhua Zheng and Yan Meng
Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ 07030, USA
yzheng1@stevense.edu, yan.meng@stevens.edu

Keywords: Vision, object detection, tracking, particle swarm optimization, Bayes law.

Abstract: This paper presents a novel Bayes-based object tracking framework boosted by a particle swarm optimization (PSO) algorithm, which is a population based searching algorithm. Basically two searching steps are conducted in this method. First, the object model is projected into a high-dimensional feature space, and a PSO algorithm is applied to search over this high-dimensional space and converge to some global optima, which are well-matched candidates in terms of object features. Second, a Bayes-based filter is used to identify the one with the highest possibility among these candidates under the constraint of object motion estimation. The proposed algorithm considers not only the object features but also the object motion estimation to speed up the searching procedure. Experimental results demonstrate that the proposed method is efficient and robust in object tracking.

1 INTRODUCTION

Object detection and tracking in images is an active research area which has attracted extensive attentions from multi-disciplinary fields, and it has wide applications in many fields like service robots, surveillance systems, public security systems, and virtual reality interfaces. Detection and tracking of moving object like car and walking people are more concerned, especially flexible and robust tracking algorithms under dynamic environments, where lightening condition may change and occlusions may happen.

Up to now, the underlying mathematical models of most tracking methods are Bayes' law estimation and Hidden Markov Model (HMM). The most popular approaches to predict discrete probability distribution are Kalman filter (G. Welch and G. Bishop, 2001), condensation (M. Isard, 1998), particle filter (S. Maskell and N. Gordon, 2001) and mean shift (D. Comaniciu, and P. Meer, 2002). Kalman filter has the same idea with HMM, while Kalman filter deals with discrete variables. Some researchers proposed different control and noise models into the recursion function for image processing, however those assumptions are dependent on varied applications and need to be tuned carefully. Condensation methods mainly focus on how to sample probabilities and likelihoods. When these methods are applied to multiple objects, a dominant peak is established if an object has large likelihood values more frequently, which may depress and lose other objects. The performance of particle filter based methods is limited by dimensionality of state space, which may be feasible in the cases with fewer targets, but may be intractable with a large amount of targets. Generally speaking, the mean-shift algorithm is efficient for object tracking. However the searching window may drift away from the object under dynamic conditions. For example, if the kernel is lost from the tracked target in one frame under some emergent situations, such as illumination condition change, it would be difficult for the tracker to recover itself from this unpredicted event.

Usually for object tracking, an analysis window based on the expectation of objects features is built and scan over the image to find out areas of interest (AOI). However, most conventional analysis-window based trackers are influenced by the shape and size of the window, which may vary from one frame to another. It is difficult to find the appropriate window for each frame, especially under dynamic environments where the content of the images may be dramatically changed.

There are various features can be used for object detection and tracking, such as color, shape, texture, gesture, contour, and motion. Some successful methods take advantage of knowledge of objects, such as shape or structures. However, the shape-based methods cannot handle the cases with occlusions efficiently. Appearance histogram is applied as tracking cue in this paper due to its independency with objects' shape and structure.

A Bayes-based object tracking approach using a particle swarm optimization (PSO) algorithm is employed to search for an optimal window in a super feature space based on appearance histogram instead of image plane directly. The PSO algorithm (J. Kennedy, R. C. Eberhart, and Y. Shi, 2001) was inspired by the social behavior of a flock of birds. In PSO algorithm, birds in a flock are symbolically represented as particles. These particles can be considered as simple agents "flying" through a problem space. A particle's location in the multi-dimensional problem space represents one solution for the problem. When a particle moves to a new location, a different solution is generated. This solution is evaluated by a fitness function that provides a quantitative value of the solution's utility.

The PSO algorithm is effective for optimization of a wide range of searching problems. In this problem, particles fly around the feature space, trying to find the best-fit tracking window parameters based on the fitness function of object features using appearance histogram. When some particles successfully detect the objects, they will share that information with their neighbors, and their neighbors may follow the directions to reach objects very quickly. Each particle makes its own decision not only based on its neighbors, but also on its own cognition, which provides the flexibility and ability of exploring new areas. This decision-making procedure can efficiently prevent the local optimum, which may cause the searching window drift.

By using PSO, the problem of identifying tracking window is translated from one-to-one estimation into one-to-many searching, which brings more flexibility. Since this searching procedure is conducted only in the object feature space, to improve searching results, a Bayes law filter is constructed based on the motion constraints of tracked objects to identify the most possible solution. Generally it is reasonable to assume that objects move consecutively in successive frames. The Bayes law filter tries to keep inertia of the object motion. Compared with conventional window-tracking algorithms, the proposed method can be executed automatically, and moving objects can be detected and tracked in a more flexible and robust way.

This paper is organized as follows. Section 2 simply reviews some related work in object detection and tracking. Section 3 introduces the PSO algorithm. The Bayes-based adaptive-window approach boosted by the PSO algorithm is described in Section 4. Experimental results are discussed and analyzed in Section 5. Conclusion and further work are given in section 6.

2 RELATED WORKS

There are many systems proposed in the past few decades for object detection and tracking. Zhang et al. (Zhang et al., 2006) proposed a robust method to detect moving objects at distance using a mobile camera. Through the utilization of the focus of expansion (FOE) and its associated residual map, the proposed method is able to detect and separate independently moving objects (IMOs) from the "moving" background caused by the camera motion. Leykin and Hammoud (Leykin and Hammoud, 2006) used a combined input from RGB and thermal cameras to build background model and tracker for pedestrians. This method showed robustness for outdoor environments. Olson and Brill (T. Olson and F. Brill, 1997) built a general purpose system for moving object detection and event recognition, where objects were detected and tracked by both first-order prediction and nearest neighbor matching.

The work which is most related to our method is (Yuri Owechko, Swarup Medasani, and Narayan Srinivasa, 2004), where the authors treated every particle as a classifier with different parameters. Those classifiers swarm in the solution space to converge to the optimal analysis window. However this is a simple application of PSO for people detection only. Reza Akbari etc. (Reza Akbari, Mohammad Davarpanah Jazi, and Maziar Palhang, 2006) employed both PSO algorithm and Kalman filter in a hybrid framework of region and object tracking, where vehicles were tracked in a cluttered background. A PSO algorithm was proposed in (Luis Anton-Canalis, Mario Hernandez-Tejera, and Elena Sanchez-Nielsen etc., 2006) to drive particles flying over image pixels directly, where object tracking emerged from interaction between particles and their environment.

3 PARTICLE SWARM OPTIMIZATION

PSO algorithm is an efficient optimization method proposed by Kennedy and Eberhart in 1995 (R. Eberhart and J. Kennedy, 1995) (J. Kennedy and R.C. Eberhart,1995) from the simulation of a simplified social model, which obviously has its root in artificial life in general, and in bird flocking, fish schooling and swarming theory in particular. On the other hand, it is also a method of evolutionary computation, related with both genetic algorithm and evolutionary programming.

The PSO algorithm is population-based: a set of potential solutions evolves to approach a convenient solution for a problem. Being an optimization method, the aim is to find the global optimum of a real-valued fitness function defined in a given search space. Rather than just being a social simulation, PSO can be treated as a powerful new search algorithm, capable of optimizing a wide range of N-dimensional problems.

The social metaphor that leads to this algorithm can be summarized as follows: the individuals that are part of a society hold an opinion that is part of a "belief space" (the search space) shared by neighboring individuals. Individuals may modify this "opinion state" based on three factors:

- The knowledge of the environment (inertia part)
- The individual's previous history of states (individual part)
- The previous history of states of the individual's neighborhood (social part)

An individual's neighborhood may be defined in several ways, configuring somehow the "social network" of the individuals. Following certain rules of interaction, the individuals in the population adapt their scheme of belief to the ones that are more successful among their social network. Over the time, a culture arises, in which the individuals hold opinions that are closely related.

In the PSO algorithm each individual is called a "particle", and is subject to a movement in a multidimensional space that represents the belief space. Particles have memory, thus retaining part of their previous states. There is no restriction for particles to share the same point in belief space, but in any case their individuality is preserved. Each particle's movement is the composition of an initial random velocity and two randomly weighted influences: individuality, the tendency to return to the particle's best previous position, and sociality, the tendency to move towards the neighborhood's best previous position.

The velocity and position of the particle at any iteration is updated based on the following equations:

$$v_{id}^{t+1} = w \cdot v_{id}^{t} + c_1 \cdot \varphi_1 \cdot (p_{id}^{t} - x_{id}^{t}) + c_2 \cdot \varphi_2 \cdot (p_{gd}^{t} - x_{id}^{t}) \quad (1)$$

$$x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1} \quad (2)$$

where v_{id}^{t} is the component in dimension *d* of the *i*th particle velocity in iteration *t*, x_{id}^{t} is the component in dimension d of the *i*th particle position in iteration *t*, c_1, c_2 are constant weight factors, p_{id}^{t} is the best position achieved by particle *i*, p_{gd}^{t} is the best position found by the neighbors of particle *i*, φ_1, φ_2 are random factors in the (0,1) interval, and w is the inertia weight. The PSO requires tuning of some parameters: the individual and sociality weights c_1, c_2 , and the inertia factor w .

The mechanism of PSO implicitly assumes that in most real world situations, the optima have better residence around them. Experimentally during the search, regions with high fitness values attract more particles and make particles concentrated after a few iterations. So this type of search is faster and more effective than traditional scanning and gradient methods. On the other hand, PSO is simpler than genetic algorithm since all particles employ the same mechanism during evolutions. Although basic PSO is designed for only single optimum, there are many works have been done to process more complex issues (Kennedy, J. &R.Eberhart, 1997).

4 THE APPROACH

4.1 General Idea

Basically, object tracking can be considered as a probability-based classification and estimation, which searches for the best match of the target model. Usually searching algorithms rely on two factors: searching space and searching window. In terms of the searching space, the more features the object has, the larger the searching space will be. To expedite the search, we can either bound the searching space with some constraints, or develop an efficient searching algorithm. Considering the searching window, adaptive windows have been extensively utilized due to its robustness.

In this paper, we propose a framework which combines a PSO-based searching algorithm and a Bayes-based probability algorithm to achieve the efficiency and robustness of the tracking systems. Basically, a PSO-based searching algorithm identifies the changes in the scene, and the probability-based algorithm estimates the best candidate of the object with the highest possibility. More specifically, the PSO algorithm takes fast scouting in a high-dimensional feature space and finds out some object candidates. Then Bayes law filter decides which one is the best match.

4.2 Object Detection

Usually object detection and recognition depend on the features of the object, such as color, texture, and shape. As indicated in (J. R. Jain and A. K. Jain, 1981), most changes in video content are typically due to the motion of objects in the depicted scene relative to the imaging plane, and a small amount of motion can result in large differences in the values of the samples in a picture, especially near the edges of objects. Often, predicting an area of the current picture from a region of the previous picture that is displayed by a few samples in spatial location can significantly reduce the need for a refining difference approximation. We call this special displacement motion vectors.

Since only the moving objects are considered to be tracked in this paper, the object detection turns into motion detection where a simple background subtraction method is applied. When the detection starts, the first several frames are looked as the background. In the following frames, the moving targets can be easily detected by a motion detection algorithm using background subtraction. During this procedure, the histogram model of background is built and updated by averaging every coming frame to achieve higher robustness. The motion vector $V_i, i = 1,2,...,N$ can be obtained, where V_i represents motion vectors of particle *i*, and *N* represents the total number of particles. Once a valid object is identified, the tracking algorithm kicks in.

4.3 PSO-based Searching Algorithm

From the view of searching, the PSO algorithm is a distributed convergence method. The key is to take advantage of sharing information between the particles as well as their own past experiences to accelerate the convergence. The PSO algorithm would provide an optimal or near-optimal solution using appropriate fitness functions without the complete knowledge of the searching space.

To identify an object in an image, usually rectangle windows are utilized in most cases. Four parameters will be identified to describe the rectangle windows, including 2D location of the central point, width and height of the rectangle, as shown in Figure 1. These parameters can build up a four-dimensional search space.

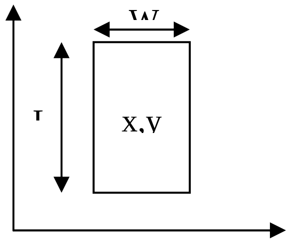


Figure 1: The four parameters associated with a particle window.

So in such a space, each particle presents a search window with specific values of parameters, which can be defined as:

$$P = \{ p_i \mid p_i(x_i, y_i, l_i, w_i), i = 1,2,..., N\} \qquad (3)$$

Where x_i and y_i represent the central point of the rectangle related to particle *i*; l_i and w_i represents the length and width related to particle *i*; and *N* is the population of swarm particles. Each individual particle has different values of these parameters. In other words, they are distributed in a four-dimensional search space.

Generally a four-dimensional feature space is very large, which makes search algorithms to be computation extensive. Some motion-based constraints can be applied to limit the search area to a smaller region where particles are initialized and move around. A straightforward constraint is using the continuity of movement since it is reasonable to assume that motion is continuous under most tracking situations. In other words, the tracking window of a new frame should be adjacent to its previous one. In this way, the initialization of particles could be accelerated. Suppose $p_{b'}(x_{b'}, y_{b'}, l_{b'}, w_{b'}, \theta_{b'})$ is the best particle (i.e., tracking window) in last frame, the initialized particles $p_i(x_i, y_i, l_i, w_i, \theta_i)$, where *i* = 1,2,…,*N*, in the new frame should be around $p_{b'}$ with some offsets in each dimension. In our experiments, locations are shifted up to 15 pixels, and sizes are shrunk and extended up to 20 percent. Therefore, by dispersing particles in a relatively smaller region instead of the whole space, searching procedure can be definitely accelerated.

Then particles move around, communicate and share information among the society, follow the better directions of their neighbors, and converge to the optima. This process is automatic and independent on knowledge of image contents. After a number of iterations, particles cluster around one or several optimal points in the space, which correspond to some regions with varied locations and sizes. These regions are candidates for the Bayes filter, which will be discussed in later section.

4.4 Fitness Function

The behaviors of particles are guided by the associated fitness function, which defines the search criteria underlying the PSO searching algorithm. In terms of object tracking, fitness function can be defined as a function of features of the tracked object. Lots of features are used for objects detection and tracking, including color, texture, shape and motion, which can be employed independently or several features can be combined together. In this paper, the appearance histogram is applied to construct the fitness function.

First, images are transformed from RGB format into HSV format, and the later one is more natural for people's eyes. Then, the values of hue are abstracted to build the histogram. Such histogram refers to the gradation of color within the visible spectrum. When a PSO-based searching algorithm is applied, each particle at every moment is associated with a histogram. The best matched one can be obtained by comparing these histograms with the target histogram. Therefore, a special criterion is required to measure the similarity between the searched window and the target window, which means a method to measure the distance between two histograms is required.

In statistics (T. Kailath, 1967), the Bhattacharyya Coefficient measures the similarity of two discrete probability distributions. It is a divergence-type measure that can be seen as the scalar product of the two vectors having components as the square root of the probability of the points $x \in X$. It thereby lends itself to a geometric interpretation: the Bhattacharyya Coefficient is the cosine of the angle enclosed between these two vectors. Therefore, the Bhattacharyya Coefficient is used to measure the similarity between these two histograms, which is defined as:

$$BC(H_i, H_g) = \sum_{x \in X} \sqrt{H_i(x) H_g(x)} \qquad (4)$$

Where H_i represents the histogram of particle i, H_g represents the histogram of the target, and X denotes the distribution domain, which is the range of hue values from 0 to 255. $H_i(x)$ and $H_g(x)$ are pixel numbers with a specific hue value x for the particle and target, respectively.

By using (4), the distance between two histograms can be defined as (D. Comaniciu, V. Ramesh, and P. Meer, 2004):

$$D(H_i, H_g) = \sqrt{1 - BC(H_i, H_g)}\ . \qquad (5)$$

This distance is invariant to the scale of the target, while the popular used histogram intersection is scale variant (M.J. Swain, D.H. Ballard, 1991). The smaller this distance is, the better the particle is matched with the target object. Thus given the target histogram, the fitness function for particle i is inversely proportional to the distance between H_i and H_g :

$$F(p_i, g) = 1 / D(H_i, H_g) \qquad (6)$$

The higher the fitness value, the more similar the corresponding area is with the target.

4.5 Bayes-Based Filter

For each frame a motion vector V can be calculated according to a motion trajectory of the tracking window. The motion vector is zero in the first frame. And for others, it is the shift from the previous position to the current one.

Given the previous tracking window associated with the target histogram and the motion vector $\{H_g, V_g\}$, where V_g represents the motion vector of target. The PSO-based searching algorithm returns a set of candidate windows, which can be represented by $\{H_i, V_i \mid i = 1,2,...,m\}$, where H_i represents histograms of particle i, V_i represents motion vectors of particle i, and m is the number of the selected candidate windows. All of these candidate windows are good enough in terms of appearance features and their fitness values are higher than a preset threshold.

According to Bayes law, the problem can be described as:

$$p(H_i, V_i \mid H_g, V_g) = \frac{p(H_g, V_g \mid H_i, V_i)\, p(H_i, V_i)}{p(H_g, V_g)} \qquad (7)$$

$p(H_i, V_i \mid H_g, V_g)$ represents the condition probability of a particle with $\{H_i, V_i\}$ given $\{H_g, V_g\}$. $p(H_g, V_g)$ represents the probability of the target window, which is same for all particles. $p(H_g, V_g \mid H_i, V_i)$ represents the back projection from candidates to the previous tracking window. Since all of particles can point back to the target window in different ways, it is hard to tell which particle is the most possible one without any predefined knowledge of the image environment. In this paper, we simply assume that all $p(H_g, V_g \mid H_i, V_i)$, i = 1,2,…,m, are equal. However this assumption may not hold in some practical applications, for instance a mobile vision system where the previous motion trajectory of the mobile platform would provide more information for the back projection, which will be investigated in our future work.

Considering that the PSO-based searching algorithm returns all of candidates which are good enough in appearance histogram, it is reasonable to ignore the histogram here and simplify (7) as:

$$p(V_i \mid V_g) = cp(V_i)\ , \qquad (8)$$

where c is a positive constant factor, and $p(V_i)$ represents the probability of a particle on the motion trajectory. According to the inertia of motion, $p(V_i)$ depends on the distance between V_i and V_g . The closer two vectors are, the higher the possibility of the corresponding particle, which makes (8) as the following equation:

$$p(V_i \mid V_g) = cp(V_i) = \frac{k}{D(V_i, V_g)} \qquad (9)$$

where k is a positive factor. If two vectors are shifted to the same original point, the distance between two vectors turns into the distance between two points, where Euclidean distance can be calculated.

5 EXPERIMENTAL RESULTS

To evaluate the proposed algorithm, some video clips from PETS database are applied in this paper. The program is written in C++ using OPENCV library, running on a Pentium4 desktop. Most data come from a surveillance system with a stationary camera.

Figure 2 shows the process of identifying moving objects by motion detection, where pictures from left to right are true data, foregrounds, and backgrounds, respectively. If there is no moving object, as shown in Figure 2(a), the background is the same with true image and the foreground is empty since no object is detected. With some general preprocessing, the noise can be depressed and the model of background can be enhanced. When a car drives in, it is detected and recognized as an object. As shown in Figure 2(b), a car shape appears in the foreground while the background keeps the same with the true image. For most testing data with static background, motion detection can detect moving objects quickly. For those testing data under dynamic environment, some pre-knowledge of objects, such as moving behaviors, would help to improve the detection performance.

Figure 3 shows the procedure of the proposed PSO algorithm searching for candidate windows. A number of particles are distributed around the target according to the tracking window of previous frame in Figure 3(a). Due to the uncertainty of the object movement, initially, these windows are set up as different sizes and locations near the detected object using motion detection. Then particles start to move around and eventually converge to some optimal points under PSO rules. Figure 3(b) shows these optimal points, which are good candidates of tracking windows. As shown in Figure 3(b), it is obviously that these candidate windows are much closer to the car compared with those initial windows in Figure 3(a), which demonstrates the efficiency of the PSO-based searching algorithm. Then Bayers filter is applied to select the best match from those good candidates, as shown in Figure 3(c). Usually, the PSO-based searching algorithm converges quickly. In our experiments, initially 20 windows are generated, then after 10 to 15 time steps, those windows cluster to the object.

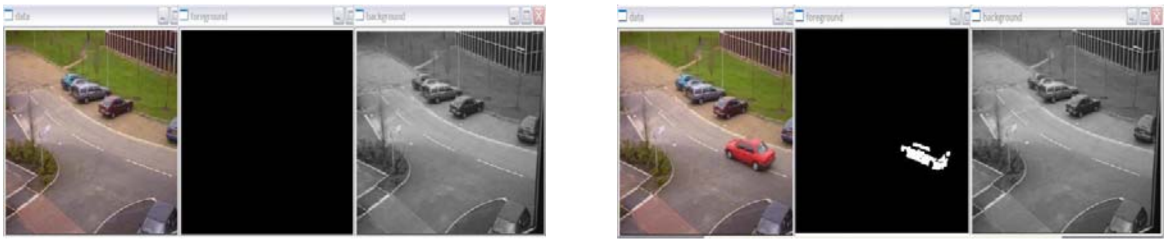
To evaluate the robustness of the proposed tracking method under occlusion, another experiment is carried out as shown in Figure 4. First, a white car drives in and is detected as the target by a blue rectangle window as shown in Figure 4(a). Then, the white car traverses the scene and is occluded by a block of texts in the image, as shown in Figure 4(b) and (c). During the occlusion, the tracking window changes with scenes, but still tracks the car. As shown in Figure 4(b), when the car starts moving into the block, the tracking has almost the same size with the one in Figure 4(a). Under the influence of the block, the tracking window shifts a little and shrinks. But the object is still locked. When the car moves away as shown in Figure 4(d), the window becomes smaller until disappeared. It can be seen that the tracker can still lock the object under occlusion.

The above experiments demonstrate the proposed algorithm is efficient and robust. However under some complex situations, such as dynamic background, more robust motion detection is required. For some noisy videos, the tracking window may be lost due to frame skips. A recovery algorithm may need to increase the system reliability.

6 CONCLUSION

In this paper, a robust adaptive-window based tracking algorithm is proposed to automatically detect and track moving objects. First, a motion detection algorithm is applied to detect the moving object. Then a PSO-based searching algorithm comes to search for good candidates of adaptive tracking windows with parameters on the new fame. Last, Bayes-based filter is employed to identify the best-matched tracking window under the motion constraints of the tracked object. The experimental results demonstrate that the proposed algorithm is robust and efficient in some popular used video data

There are still several issues remained and need to be improved and extended in the future work. The first one is to investigate new object detection approaches under dynamic environment where the background of the image and illumination conditions may be dramatically changed and the motion detection and histogram-based method applied in this paper will not be reliable any more. The second one is to concrete the Bayes filter using some predefined knowledge of the tracked targets.



(a) (b)

Figure 2: Motion detection to recognize objects, (a) to (b) from left to right.

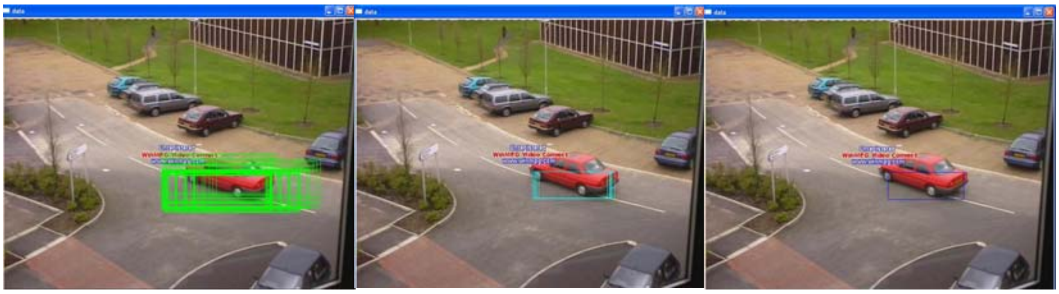
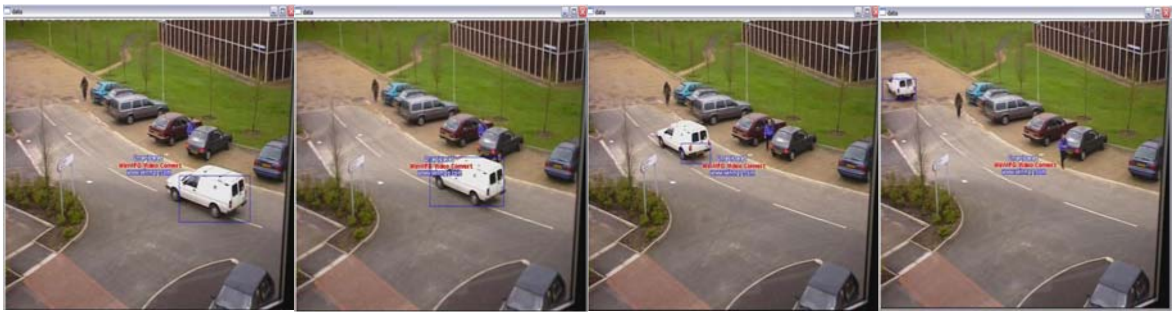


Figure 3: Tracking procedure using PSO-based searching.



(a) (b) (c) (d)

Figure 4: Tracking under occlusion.

REFERENCES

G. Welch and G. Bishop, "An Introduction to Kalman Filter", *SIGGRAPH 2001*, Course 8, 2001.

M. Isard, "Visual Motion Analysis by Probabilistic Propagation of Conditional Density", *D.Phil. Thesis*, Oxford University, 1998.

S. Maskell and N. Gordon, "A Tutorial on Particle Filters for On-line Nonlinear/Non-Gaussian Bayes'Law Tracking", 2001.

D. Comaniciu, and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Trans. Pattern Analysis Machine Intel.*, Vol. 24, No. 5, pp. 603-619, 2002.

J. Kennedy, R. C. Eberhart, and Y. Shi, *Swarm intelligence*, San Francisco, Morgan Kaufmann Publishers, 2001.

T. Olson and F. Brill. "Moving Object Detection and Event Recognition Algorithms for Smart Cameras*", in Proc. DARPA Image Understanding Workshop,* pp. 159-175, 1997.

Yuri Owechko, Swarup Medasani, and Narayan Srinivasa, "Classifier Swarms for Human Detection in Infrared Imagery," in *2004 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04)*, Vol. 8, pp. 121, 2004.

Reza Akbari, Mohammad Davarpanah Jazi, and Maziar Palhang "A Hybrid Method for Robust Multiple Objects Tracking in Cluttered Background", Vol. 1, 24-28, *ICTTA '06*.

Luis Anton-Canalis, Mario Hernandez-Tejera, and Elena Sanchez-Nielsen etc. "Particle Swarms as Video Sequence Inhabitants For Object Tracking in Computer Vision", Vol. 2, pp. 604-609, *ISDA '06*.

R. Eberhart and J. Kennedy. "A New Optimizer Using Particles Swarm Theory", in *Proc. Sixth International Symposium on Micro Machine and Human Science*, Nagoya, Japan., IEEE Service Center, Piscataway, NJ, 39-43, 1995.

J. Kennedy and R.C. Eberhart, "Particle Swarm Optimization", in *Proceedings of IEEE International Conference on Neural Networks*, 1942-1948, 1995.

Kennedy, J. and R. Eberhart, "A discrete binary version of the particle swarm algorithm," in *Proceedings of the Conference on Systems, Man and Cybernetics*, pp. 4104-4109, Piscataway, New Jersey, IEEE Press, 1997.

J. R. Jain and A. K. Jain, "Displacement measurement and its application in interframe image coding", *IEEE Trans. Commun*. Vol. COM-29, no.12, pp.1799-1808, Dec. 1981.

T. Kailath, "The Divergence and Bhattacharyya Dis-tance Measures in Signal Selection," *IEEE Trans. Com-mun. Tech*., COM-15:52-60, 1967.

D. Comaniciu, V. Ramesh, and P. Meer, "Real-Time Tracking of Non-Rigid Objects using Mean Shift", *CVPR*, Vol. 2, pp. 142-149, 2000.

M. J. Swain, D.H. Ballard, "Color Indexing," *Intern. J.Comp. Vis.*, 7(1):11-32, 1991.

A. Leykin and R. I. Hammoud, "Robust Multi-Pedestrian Tracking in Thermal-Visible Surveillance Videos", *IEEE Computer Vision and Pattern Recognition Workshop, OTCBVS'06*, New York,USA, 17-22 June 2006 pp:136 – 141

Y. Zhang, S. J. Kiselewich, W. A. Bauson, and R. Hammoud, "Robust Moving Object Detection at Distance in the Visible Spectrum and Beyond Using a Moving Camera", *IEEE Computer Vision and Pattern Recognition Workshop (CVPRW), on Object Tracking and Classification Beyond the Visible Spectrum*, New York, USA, 17-22 June 2006, pp: 1331-137

SMARTMOBILE – AN ENVIRONMENT FOR GUARANTEED MULTIBODY MODELING AND SIMULATION

Ekaterina Auer and Wolfram Luther
IIIS, University of Duisburg-Essen, Lotharstr. 63, Duisburg, Germany
{auer,luther}@inf.uni-due.de

Keywords: Validated method, interval, Taylor model, initial value problem, guaranteed multibody modeling and simulation.

Abstract: Multibody modeling and simulation is important in many areas of our life from a computer game to space exploration. To automatize the process for industry and research, a lot of tools were developed, among which the program MOBILE plays a considerable role. However, such tools cannot guarantee the correctness of results, for example, due to possible errors in the underlying finite precision arithmetic. To avoid such errors and simultaneously prove the correctness of results, a number of so called validated methods were developed, which include interval, affine and Taylor form based arithmetics. In this paper, we present the recently developed multibody modeling and simulation tool SMARTMOBILE based on MOBILE, which is able to guarantee the correctness of results. The use of validated methods there allows us additionally to take into account the uncertainty in measurements and study its influence on simulation. We demonstrate the main concepts and usage with the help of several mechanical systems, for which kinematical or dynamic behavior is simulated in a validated way.

1 INTRODUCTION

Modeling and simulation of kinematics and dynamics of mechanical systems is employed in many branches of modern industry and applied science. This fact contributed to the appearance of various tools for automatic generation and simulation of models of multibody systems, for example, MOBILE (Kecskeméthy, 1993). Such tools produce a model (mostly a system of differential or algebraic equations or both) from a formalized description of the goal mechanical system. The system is then solved using a corresponding numerical algorithm. However, the usual implementations are based on finite precision arithmetic, which might lead to unexpected errors due to round off and similar effects. For example, unreliable numerics might ruin an election (German Green Party Convention in 2002) or cost people lives (Patriot Missile failure during the Golf War), cf. (Huckle, 2005).

Aside from finite precision errors, possible measurement uncertainties in model parameters and errors induced by model idealization encourage the employment of a technique called interval arithmetic and its extensions in multibody modeling and simulation tools. Essential ideas of interval arithmetic were developed simultaneously and independently by several people whereas the most influential theory was formulated by R. E. Moore (Moore, 1966). Instead of providing a point on the real number axis as an (inexact) answer, intervals supply the lower and upper bounds that are guaranteed to contain the true result. These two numbers can be chosen so as to be exactly representable in a given finite precision arithmetic, which cannot be always ensured in the usual finite precision case. The ability to provide a guaranteed result supplied a name for such techniques – ”validated arithmetics”. Their major drawback is that the output might be too uncertain (e.g. $[-\infty;+\infty]$) to provide a meaningful answer. Usually, this is an indication that the problem might be ill conditioned or inappropriately formulated, and so the finite precision result wrong.

To minimize the possible influence of overestimation on the interval result, this technique was extended with the help of such notions as affine (de Figueiredo and Stolfi, 2004) or Taylor forms/models (Neumaier, 2002). Besides, strategies and algorithms much less vulnerable to overestimation were developed. They include rearranging expression evaluation, coordinate transformations, or zonotopes (Lohner, 2001).

The second focus in this paper is MOBILE, an object oriented C++ environment for modeling and simulation of kinematics and dynamics of mechanical systems based on the multibody modeling method. Its central concept is a transmission element which maps motion and force between system states. For example, an elementary joint modeling revolute and prismatic joints is such a transmission element. Mechanical systems are considered to be concatenations of these entities. In this way, serial chains, tree type or closed loop systems can be modeled. With the help of the global kinematics, the transmission function of the complete system chain can be obtained from transmission functions of its parts. The inverse kinematics and the kinetostatic method (Kecskeméthy and Hiller, 1994) help to build dynamic equations of motion, which are solved with common initial value problem (IVP) solvers. MOBILE belongs to the numerical type of modeling software, that is, it does not produce a symbolic description of the resulting model. Only the values of output parameters for the user-defined values of input parameters and the source code of the program itself are available.

SMARTMOBILE (Simulation and Modeling of dynAmics in MOBILE: Reliable and Template based) enhances the usual, floating point based MOBILE with validated arithmetics and IVP solvers (Auer et al., 2006a). In this way, it can model and perform validated simulation of the behavior of various classes of mechanical systems including non-autonomous and closed-loop ones as well as provide more realistic models by taking into account the uncertainty in parameters.

In this paper, we give an overview of the structure and the abilities of SMARTMOBILE. First, the main validated techniques and software are referenced briefly in Section 2. In Section 3, the main features of MOBILE are described in short to provide a better understanding of the underlying structure of SMARTMOBILE. In Section 4, we describe the implementation of SMARTMOBILE in some detail and validate kinematical and dynamic behavior of several example systems with the help of this environment. We summarize the paper in Section 5. On the whole, we give an overview of the potential of validated methods in mechanical modeling, and, in particular, the potential of SMARTMOBILE.

2 VALIDATED METHODS AND SOFTWARE

To guarantee the correctness of MOBILE results, it is necessary to enhance this tool with validated concepts. Fortunately, we do not need to implement these concepts from scratch. In the last decades, various libraries were implemented that supported different aspects of validated calculus. In the first Subsection, we name several of these tools. After that, we give a very brief overview of interval arithmetic and an algorithm for solving IVPs (initial value problems) to provide a reference about the difference of validated methods to the usual floating point ones.

2.1 Validating Multibody Tools of the Numerical Type

To validate the results of multibody modeling and simulation software of the numerical type, the following components are necessary. First, the means are required to work with arithmetic operations and standard functions such as sine or cosine in a guaranteed way. Here, the basic principles of interval calculus and its extensions are used. Interval arithmetic is implemented in such libraries as PROFIL/BIAS (Knüppel, 1994), FILIB++ (Lerch et al., 2001), C-XSC (Klatte et al., 1993). LIBAFFA (de Figueiredo and Stolfi, 2004) is a library for affine arithmetic, whereas COSY (Berz and Makino, 2002) implements Taylor models.

Second, validated algorithms for solving systems of algebraic, differential or algebraic-differential equations are necessary. C-XSC TOOLBOX (Hammer et al., 1995) offers a general means of solving different classes of systems as well as an implementation in C-XSC. For IVP solving in interval arithmetic, there exist such packages as AWA (Lohner, 1988), VNODE (Nedialkov, 2002), and recently developed VALENCIA-IVP (Auer et al., 2006a). In the framework of Taylor models, the solver COSY VI (Berz and Makino, 1998) was developed.

Finally, almost all of the above mentioned solvers need means of computing (high order) derivatives automatically. Some of them, for example, COSY VI, use the facilities provided by the basis arithmetic in COSY. Interval implementations do not possess this facility in general; external tools are necessary in this case. The symbolic form of the mathematical model, which is necessary to be able to obtain derivatives automatically, is not available in case of software of the numerical type. However, a method called algorithmic differentiation (Griewank, 2000) offers a possibility to obtain the derivatives using the code of the program itself.

There are two main techniques to implement algorithmic differentiation of a piece of program code: overloading and code transformation. In the first case, a new data type is developed that is capable of computing the derivative along with the function value. This new data type is used instead of the simple one in the code piece. The drawback of this method is the lack of automatic optimization during the derivative computation. FADBAD++ (Stauning and Bendtsen, 2005) is a generic library implementing this approach for arbitrary user-defined basic data types. The technique of code transformation presupposes the development of a compiler that takes the original code fragment and the set of differentiation rules as its input and produces a program delivering derivatives as its output. This approach might be difficult to implement for large pieces of code which are self-contained programs themselves. However, derivatives can be evaluated more efficiently with this technique. An implementation is offered in the library ADOL-C (Griewank et al., 1996).

This list of tools is not supposed to be complete. All of the above mentioned packages are implemented (or have versions) in C++, an important criterium from our point of view since MOBILE is also implemented in this language.

2.2 Theory Overview

In this Subsection, we consider the basic principles of validated computations using the example of interval arithmetic. First, elementary operations in this arithmetic are described. Then a basic interval algorithm for solving IVPs is outlined to give an impression of the difference to floating point analogues. In particular, the latter passage makes clear why automatic differentiation is unavoidable while simulating dynamics of mechanical systems, that is, solving systems of differential equations.

An interval $[\underline{x};\overline{x}]$, where $\underline{x}$ is the lower, $\overline{x}$ the upper bound, is defined as $[\underline{x};\overline{x}] = \{x \in \mathbb{R} : \underline{x} \leq x \leq \overline{x}\}$. For any operation $\circ = \{+,-,\cdot,/\}$ and intervals $[\underline{x};\overline{x}]$, $[\underline{y};\overline{y}]$, the corresponding interval operation can be defined as $[\underline{x};\overline{x}] \circ [\underline{y};\overline{y}] =$

$$[min(\underline{x}\circ\underline{y},\underline{x}\circ\overline{y},\overline{x}\circ\underline{y},\overline{x}\circ\overline{y}); max(\underline{x}\circ\underline{y},\underline{x}\circ\overline{y},\overline{x}\circ\underline{y},\overline{x}\circ\overline{y})] \ .$$

Note that the result of an interval operation is also an interval. Every possible combination of $x \circ y$, where $x \in [\underline{x};\overline{x}]$ and $y \in [\underline{y};\overline{y}]$, lies inside this interval. (For division, it is assumed that $0 \notin [\underline{y};\overline{y}]$.)

To be able to work with this definition on a computer using a finite precision arithmetic, a concept of a machine interval is necessary. The machine interval has machine numbers as the lower and upper bounds. To obtain the corresponding machine interval for the real interval $[\underline{x};\overline{x}]$, the lower bound is rounded down to the largest machine number equal or less than $\underline{x}$, and the upper bound is rounded up to the smallest machine number equal or greater than $\overline{x}$.

Consider an algorithm for solving the IVP

$$\dot{x}(t) \ = \ f(x(t)), \quad x(t_0) \ \in \ [x_0], \qquad (1)$$

where $t \in [t_0, t_n] \subset \mathbb{R}$ for some $t_n > t_0$, $f \in C^{p-1}(\mathcal{D})$ for some $p > 1$, $\mathcal{D} \subseteq \mathbb{R}^m$ is open, $f : \mathcal{D} \mapsto \mathbb{R}^m$, and $[x_0] \subset \mathcal{D}$. The problem is discretized on a grid $t_0 < t_1 < \cdots < t_n$ with $h_{k-1} = t_k - t_{k-1}$. Denote the solution with the initial condition $x(t_{k-1}) = x_{k-1}$ by $x(t; t_{k-1}, x_{k-1})$ and the set of solutions $\{x(t; t_{k-1}, x_{k-1}) \mid x_{k-1} \in [x_{k-1}]\}$ by $x(t; t_{k-1}, [x_{k-1}])$. The goal is to find interval vectors $[x_k]$ for which the relation $x(t_k; t_0, [x_0]) \subseteq [x_k]$, $k = 1, \ldots, n$ holds.

The (simplified) kth time step of the algorithm consists of two stages (Nedialkov, 1999) :

1. Proof of existence and uniqueness. Compute a step size h_{k-1} and an a priori enclosure $[\tilde{x}_{k-1}]$ of the solution such that

(i) $x(t; t_{k-1}, x_{k-1})$ is guaranteed to exist for all $t \in [t_{k-1}; t_k]$ and all $x_{k-1} \in [x_{k-1}]$,

(ii) the set of solutions $x(t; t_{k-1}, [x_{k-1}])$ is a subset of $[\tilde{x}_{k-1}]$ for all $t \in [t_{k-1}; t_k]$.

Here, Banach's fixed-point theorem is applied to the Picard iteration.

2. Computation of the solution. Compute a tight enclosure $[x_k] \subseteq [\tilde{x}_{k-1}]$ of the solution of the IVP such that $x(t_k; t_0, [x_0]) \subseteq [x_k]$. The prevailing algorithm is as follows.

2.1. Choose a one-step method

$x(t; t_k, x_k) = x(t; t_{k-1}, x_{k-1}) + h_{k-1}\varphi(x(t; t_{k-1}, x_{k-1})) + z_k$,

where $\varphi(\cdot)$ is an appropriate method function, and z_k is the local error which takes into account discretization effects. The usual choice for $\varphi(\cdot)$ is a Taylor series expansion.

2.2. Find an enclosure for the local error z_k. For the Taylor series expansion of order $p-1$, this enclosure is obtained as $[z_k] = h_{k-1}^{p} f^{[p]}([\tilde{x}_{k-1}])$, where $f^{[p]}([\tilde{x}_{k-1}])$ is an enclosure of the pth Taylor coefficient of the solution over the state enclosure $[\tilde{x}_{k-1}]$ determined by the Picard iteration in Stage One.

2.3. Compute a tight enclosure of the solution. If mean-value evaluation for computing the enclosures of the ranges of $f^{[i]}([x_k])$, $i = 1, ..., p-1$, instead of the direct evaluation of $f^{[i]}([x_k])$ is used, tighter enclosures can be obtained.

Note that Taylor coefficients and their Jacobians (used in the mean-value evaluation) are necessary to be able to use this algorithm.

3 MOBILE

A transmission element, the basis of MOBILE, maps motion and loads between state objects (coordinate frames or variables) according to

$$\begin{array}{rclrcl} q' &=& \phi(q), & \dot{q}' &=& \mathbf{J}_\phi\,\dot{q}, \\ \ddot{q}' &=& \mathbf{J}_\phi\,\ddot{q} + \dot{\mathbf{J}}_\phi\,\dot{q}, & Q &=& \mathbf{J}_\phi^{\mathrm{T}}\,Q' \ . \end{array} \tag{2}$$

$\mathbf{J}_\phi$ is the Jacobian of ϕ, the mapping for the motion transmission. Other characteristics are vectors of dimension depending on the degrees of freedom of a mechanical system. Here, q and q' are the generalized positions, $\dot{q}$ and $\dot{q}'$ the velocities, $\ddot{q}$ and $\ddot{q}'$ the accelerations, as well as Q and Q' the forces of the transmission element in the original and final states, respectively. The transmission of force is assumed to be ideal. That is, power is neither generated nor consumed.

Models in MOBILE are concatenations of transmission elements. The overall mapping of this concatenation from the original state into the final one is obtained by the composition of the corresponding mappings of the intermediate states. Concatenated elements are considered as a single transmission element. This helps to solve the task of the global kinematics: to obtain the positions, the orientations, the velocities, and the accelerations of all bodies of a mechanical system from the given q, $\dot{q}$, and $\ddot{q}$.

All transmission elements are derived from the abstract class `MoMap`, which supplies their main functionality including the methods `doMotion()` and `doForce()` for transmission of motion and force. For example, elementary joints are modeled by the `MoMap`-derived class `MoElementaryJoint`. Besides, there exist elements for modeling mass properties and applied forces. The corresponding representations of the mapping (2) for these elements are described in (Kecskeméthy, 1993).

Transmission elements are assembled to chains implemented by the class `MoMapChain`. The methods `doMotion()` and `doForce()` can be used for a chain representing the system to determine the corresponding composite transmission function.

To model dynamics of a mechanical system, the equations of motion have to be built and solved. Their minimal form is given by

$$M(q;t)\,\ddot{q} + b(q,\dot{q};t) = Q(q,\dot{q};t)\ , \tag{3}$$

where $M(q;t)$ is the generalized mass matrix, $b(q,\dot{q};t)$ the vector of generalized Coriolis and centrifugal forces, and $Q(q,\dot{q};t)$ the vector of applied forces. The class `MoEqmBuilder` is responsible for generation of such equations, that is, computation of M, b, and Q for each given q, $\dot{q}$, and t.

After the introduction of a state vector $x = \left[q^T, \dot{q}^T\right]^T$, the state-space form of the state equations is obtained as

$$\dot{x} = \left[\begin{array}{c} \dot{q} \\ \ddot{q} \end{array}\right] = \left[\begin{array}{c} \dot{q} \\ -M^{-1}\widehat{Q} \end{array}\right]\ , \tag{4}$$

where $\widehat{Q}(q,\dot{q};t) = b(q,\dot{q};t) - Q(q,\dot{q};t)$. This is the responsibility of the class `TMoMechanicalSystem`.

Finally, an IVP corresponding to (4) is solved by an appropriate integrator algorithm, for example, Runge–Kutta's using the class `MoRungeKuttaIntegrator` derived from the basic class `MoIntegrator`.

MOBILE models and simulates mechanical systems directly as executable programs. This allows the user to embed the resulting modules in existing libraries easily. Besides, the core of MOBILE is extendable owing to its open system design.

As already mentioned, MOBILE belongs to the numerical type of the modeling software, by which we mean that it does not produce the symbolic description of the resulting mathematical model, as opposed to the symbolical type. In the latter case, the process of validation of the model is basically reduced to the application of the validated methods to the obtained system of equations. In the former case, it is necessary to integrate verified techniques into the core of the software itself, the task which cannot always be solved since many modeling tools are not open source.

4 SMARTMOBILE

In this Section, we first describe the main features of the recently developed multibody modeling and simulation tool SMARTMOBILE which produces guaranteed results in the constraints of the given model. There, the modeling itself can be enhanced by taking into account the uncertainty in parameters, which might result, for example, from measurements. After that, we demonstrate the possibilities SMARTMOBILE offers by simulating kinematical and dynamic behavior of two example systems in a validated way.

4.1 Main Features

The focus of SMARTMOBILE is to model and simulate dynamics of mechanical systems in a guaranteed way. The concept behind MOBILE, however, presupposes that kinematics is also modeled (almost as a by-product) and so it is easy to simulate it afterwards. That is why SMARTMOBILE is one of the rare validated tools that possess both functionalities.

To simulate dynamics, it is necessary to solve an IVP for the differential(-algebraic) equations of motion of the system model in the state space form. As already mentioned, validated IVP solvers need derivatives of the right side of these equations. They can be obtained using algorithmic differentiation, the method that is practicable but might consume a lot of CPU time in case of such a large program as MOBILE. An alternative is to make use of the system's mechanics for this purpose. This option is not provided by MOBILE developers yet and seems to be rather difficult to algorithmize for (arbitrary) higher orders of derivatives. That is why it was decided to employ the first possibility in SmartMOBILE.

To obtain the derivatives, SmartMOBILE uses the overloading technique. In accordance with Subsection 2.1, all relevant occurrences of `MoReal` (an alias of `double` in MOBILE) have to be replaced with an appropriate new data type. Almost each validated solver needs a different basic validated data type. Therefore, the strategy in SmartMOBILE is to use pairs type/solver. To provide interval validation of dynamics with the help of VNODE-based solver `TMoAWAIntegrator`, the basic data type `TMoInterval` including data types necessary for algorithmic differentiation should be used. The data type `TMoFInterval` enables the use of `TMoValenciaIntegrator`, an adjustment of the basic version of ValEncIA-IVP. The newly developed `TMoRiotIntegrator` is based on the IVP solver from the library RiOT, an independent C++ version of COSY and COSY VI, and requires the class `TMoTaylorModel`, a SmartMOBILE-compatible wrapper of the library's own data type `TaylorModel`. Analogously, to be able to use an adjustment of COSY VI, the wrapper `RDAInterval` is necessary. Modification of the latter solver for SmartMOBILE is currently work in progress.

In general, kinematics can be simulated with the help of all of the above mentioned basic data types. However, other basic data types might become necessary for more specific tasks such as finding of equilibrium states of a system since they require specific solvers. SmartMOBILE provides an option of modeling equilibrium states in a validated way with the help of the interval-based data type `MoFInterval` and the class `MoIGradientStaticEquilibriumFinder`, a version of the zero-finding algorithm from the C-XSC Toolbox.

The availability of several basic data types in SmartMOBILE points out its second feature: the general data type independency through its template structure. That is, `MoReal` is actually replaced with a placeholder and not with a concrete data type. For example, the transmission element `MoRigidLink` from MOBILE is replaced with its template equivalent `TMoRigidLink`, the content of the placeholder for which (e.g. `TMoInterval` or `MoReal`, cf. Figure 1) can be defined at the final stage of the system assembly. This allows us to use a suitable pair consisting of the data type and solver depending on the application at hand. If only a reference about the form of the solution is necessary, `MoReal` itself and a common numerical solver (e.g. Runge-Kutta's) can be used. If a relatively fast validation of dynamics without much uncertainty in parameters is of interest, `TMoInterval` and `TMoAWAIntegrator` might be the choice. For validation of highly nonlinear systems with a considerable uncertainty, the slower combination of `TMoTaylorModel` and `TMoRiOTIntegrator` can be used.

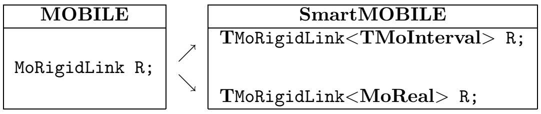


Figure 1: Template usage.

A MOBILE user can easily switch to SmartMOBILE because the executable programs for the models in both environments are similar. In the validated environment, the template syntax should be used. The names of transmission elements are the same aside from the preceding letter `T`. The methods of the classes have the same names, too. Only the solvers are, of course, different, although they follow the same naming conventions.

4.2 Examples

First, we give an example of the guaranteed simulation of kinematics of a five arm manipulator, the system defined in detail in (Traczinski, 2006). The modeling of the system itself can be enhanced in SmartMOBILE by using so called sloppy joints (Traczinski, 2006) instead of usual revolute ones. In the transmission element responsible for the modeling of the

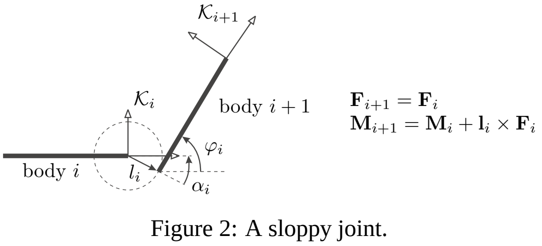

$\mathcal{K}_{i+1}$
$\mathcal{K}_i$
body $i+1$
φ_i
body i
l_i
α_i
$\mathbf{F}_{i+1} = \mathbf{F}_i$
$\mathbf{M}_{i+1} = \mathbf{M}_i + \mathbf{l}_i \times \mathbf{F}_i$

Figure 2: A sloppy joint.

Table 1: Widths of the position enclosures.

	x	y	CPU (s)
`TMoInterval`	1.047	1.041	0.02
`TMoTaylorModel`	0.163	0.290	0.14

sloppy joint, it is no longer assumed that the joint connects the rotational axes of two bodies exactly concentrically. Instead, the relative distance between the axes is supposed to be within a specific (small) range. Two additional parameters are necessary to describe the sloppy joint (cf. Figure 2): radius $l_i \in [0; l_{max}]$ and the the relative orientation angle $\alpha_i \in [0; 2\pi)$ (the parameter φ_i that describes the relative orientation between two connected bodies is the same both for the sloppy and the usual joint).

The considered system consists of five arms of the lengths $l_0 = 6.5$m, $l_1 = l_2 = l_3 = 4$m, and $l_4 = 3.5$m, each with the uncertainty of ±1%. The arms are connected with five sloppy joints, for which $l_{max} = 2$mm and initial angle constellation is $\varphi_0 = 60^\circ$, $\varphi_1 = \varphi_2 = \varphi_3 = -20^\circ$, and $\varphi_4 = -30^\circ$. Each of these angles has an uncertainty of $\pm 0.1^\circ$.

In Figure 3, left, the (abridged) SMARTMOBILE model of the manipulator is shown (the system geometry described above is omitted). First, all the necessary coordinate frames are defined with the help of the array `K`. Analogously, the required rigid links `L` and sloppy joints `R`, with which arms and their connections are modeled, are declared. They are defined later inside the `for` loop. The array `l` characterizes the lengths of the rigid links, and `phi` is used to define the angles of sloppy joints. All elements are assembled into one system using the element `manipulator`. By calling the method `doMotion()` for this element, we can obtain the position of the tip of the manipulator, which equals the rotational matrix `R` multiplied by the translational vector `r`, both stored in the last coordinate frame `K[10]`.

We simulate kinematics with the help of intervals and Taylor models. That is, the placeholder `type` is either `TMoInterval` or `TMoTaylorModel`. Both position enclosures are shown in Figure 3, right. Note that enclosures obtained with intervals are wider than whose obtained with Taylor models (cf. also Table 1, where the widths of the corresponding intervals are shown). Taylor models are bounded by intervals to provide a comparison. Numbers are rounded up to the third digit after the decimal point. CPU times are measured on a Pentium 4, 3.0 GHz PC under CYGWIN.

The statistic analysis of the same system with the help of the Monte-Carlo method (Metropolis and Ulam, 1949) carried out in (Hörsken, 2003) shows that the results, especially those for Taylor models, are acceptable. Although the set of all possible positions obtained statistically with the help of 50,000 simulations (solid black area in Figure 3) is not as large as even the rectangle obtained with `TMoTaylorModel`, there might exist parameter constellations which lead to the results from this rectangle. Besides, statistical simulations require a lot of CPU time, which is not the case with SMARTMOBILE. Additionally, the results are proven to be correct there through the use of validated methods.

Table 2: Performance of `TMoAWAIntegrator`, `TMoRiOTIntegrator`, and `TMoValenciaIntegrator`. for the double pendulum over the time interval $[0; 0.4]$.

Strategy	`AWA`	`RiOT`	`Valencia`
Break-down	0.424	0.820	0.504
CPU time	1248	9312	294

The next example is the double pendulum with an uncertain initial angle of the first joint from (Auer et al., 2006a). The lengths of both massless arms of the pendulum are equal to 1m and the two point masses amount to 1kg each with the gravitational constant $g = 9.81\frac{\text{m}}{\text{s}^2}$ The initial values for angles (specified in rad) and angular velocities (in $\frac{\text{rad}}{\text{s}}$) are given as

$$\begin{bmatrix} \frac{3\pi}{4} & -\frac{11\pi}{20} & 0.43 & 0.67 \end{bmatrix}^T ,$$

where the initial angle of the first joint has an uncertainty of ±1% of its nominal value.

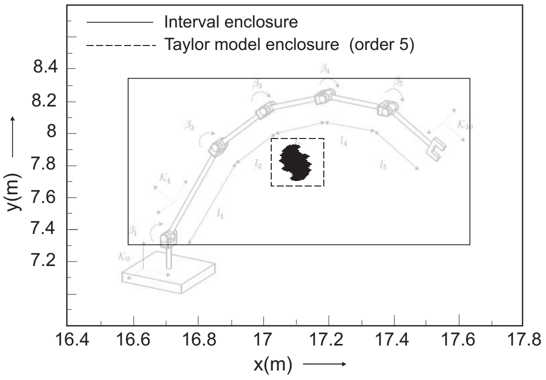
The interval enclosures of the two angles β_1 and β_2 of the double pendulum are shown for identical time intervals in Figure 4. Besides, Table 2 summarizes the results. The line ”Break-down” contains the time in seconds after which the corresponding method no longer works. That is, the correctness of results cannot be guaranteed after that point. This happens here due to both the chaotic character of the considered system and the resulting overestimation. The last line indicates the CPU time (in seconds) which the solvers take to obtain the solution over the integration interval $[0; 0.4]$. Note that the CPU times are provided only as a rough reference since the solvers can be further optimized in this respect.

The use of `TMoValenciaIntegrator` improves both the tightness of the resulting enclosures and the CPU time in comparison to `TMoAWAIntegrator` for this example. Although `TMoRiOTIntegrator` breaks down much later than the both former solvers, it needs a lot of CPU time.

The double pendulum is a simple example of the opportunities that SMARTMOBILE offers for dy-

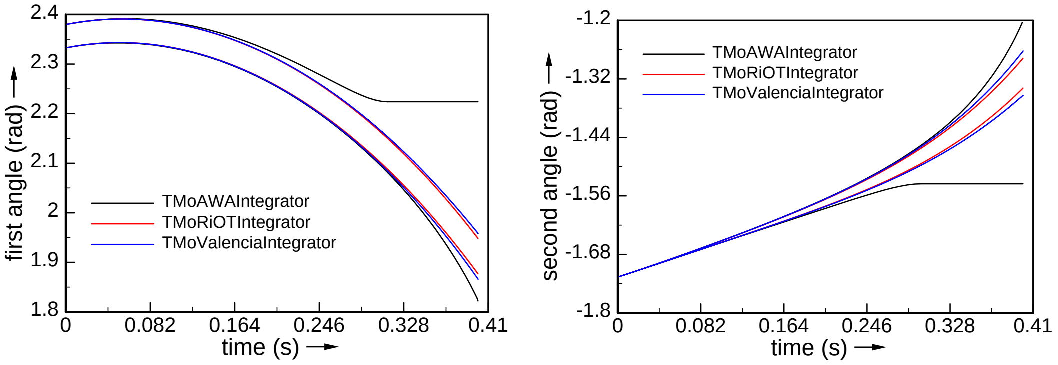
```
TMoFrame<type>      *K=new TMoFrame<type>[2*5+1];
TMoAngularVariable<type> *phi=new
                          TMoAngularVariable<type>[5];
TMoVector<type>        *l=new TMoVector<type>[5];
TMoSlacknessJoint<type> *R=new
                           TMoSlacknessJoint<type>[5];
TMoRigidLink<type> *L=new TMoRigidLink<type>[5];
TMoMapChain<type>                    manipulator;
double l_max;
for(int i=0;i<5;i++){
    TMoSlacknessJoint<type> joint(K[2*i],K[2*i+1],
                                  phi[i],zAxis,l_max);
    TMoRigidLink<type> link(K[2*i],K[2*i+1],l[i]);
    R[i]=joint; L[i]=link;
    Manipulator<<R[i]<<L[i];
}
Manipulator.doMotion();
cout<<"Position="<<K[10].R*K[10].r;
```

(a) SMARTMOBILE model.



(b) Enclosures of the position of the tip.

Figure 3: Kinematics of the five arm manipulator with uncertain parameters.



(a) Enclosure of the first joint angle.

(b) Enclosure of the second joint angle.

Figure 4: Interval enclosures for the first and second state variable of the double pendulum.

namic simulation. More close-to-life examples are treated in (Auer et al., 2006b), (Auer, 2006), (Auer et al., 2004).

At last, consider the previous example once again but without the uncertainty in β_1. To find equilibrium states of this system, we use the basic data type MoFInterval instead of TMoInterval and apply MoIGradientStaticEquilibriumFinder to the element manipulator instead of using TMoMechanicalSystem and an integrator. All four possible equilibria (stable and unstable) are found by the validated solver MoIGradientStaticEquilibriumFinder in the starting interval $[-3.15;1]$ for all coordinates (shown rounded up to the third digit after the decimal point):

```
1: ([-3.142;-3.142];[-3.142;-3.142])
2: ([-3.142;-3.142];[-0.000; 0.000])
3: ([-0.000; 0.000];[-3.142;-3.142])
4: ([-0.000; 0.000];[-0.000; 0.000])
```

Since we do not have any uncertainties in the model, the intervals obtained are very close to point intervals, that is, $\underline{\beta}_i \approx \overline{\beta}_i$. The difference is noticeable only after the 12-th digit after the decimal point. However, if the same problem is modeled using the non-verified model in MOBILE, only one (unstable) equilibrium state $[\beta_1, \beta_2] =$[3.142;-3.142] is obtained (using the identical initial guess).

5 CONCLUSIONS

In this paper, we presented a recently developed tool SMARTMOBILE for guaranteed modeling and simulation of kinematics and dynamic of mechanical systems. With its help, the behavior of different classes of systems including non-autonomous and closed-loop ones can be obtained with the guarantee of correctness, the option which is not given in tools based on floating point arithmetics. Besides, the uncertainty in parameters can be taken into account in a natural way. Moreover, SMARTMOBILE is flexible and allows the user to choose the kind of underlying arithmetics according to the task at hand. The functionality of the tool was demonstrated using three examples.

The main directions of the future development will include enhancement of validated options for modeling and simulation of closed-loop systems in SMARTMOBILE as well as integration of further verified solvers into its core.

REFERENCES

Auer, E. (2006). Interval Modeling of Dynamics for Multibody Systems. In *Journal of Computational and Applied Mathematics*. Elsevier. Online.

Auer, E., Kecskeméthy, A., Tändl, M., and Traczinski, H. (2004). Interval Algorithms in Modeling of Multibody Systems. In Alt, R., Frommer, A., Kearfott, R., and Luther, W., editors, *LNCS 2991: Numerical Software with Result Verification*, pages 132 – 159. Springer, Berlin Heidelberg New York.

Auer, E., Rauh, A., Hofer, E. P., and Luther, W. (2006a). Validated Modeling of Mechanical Systems with SMARTMOBILE: Improvement of Performance by VALENCIA-IVP. In *Proc. of Dagstuhl Seminar 06021: Reliable Implementation of Real Number Algorithms: Theory and Practice*, Lecture Notes in Computer Science. To appear.

Auer, E., Tändl, M., Strobach, D., and Kecskeméthy, A. (2006b). Toward validating a simplified muscle activation model in SMARTMOBILE. In *Proceedings of SCAN 2006*. submitted.

Berz, M. and Makino, K. (1998). Verified integration of ODEs and flows using differential algebraic methods on high-order Taylor models. *Reliable Computing*, 4:361–369.

Berz, M. and Makino, K. (2002). COSY INFINITY Version 8.1. User's guide and reference manual. Technical Report MSU HEP 20704, Michigan State University.

de Figueiredo, L. H. and Stolfi, J. (2004). Affine arithmetic: concepts and applications. *Numerical Algorithms*, 37:147158.

Griewank, A. (2000). *Evaluating derivatives: principles and techniques of algorithmic differentiation*. SIAM.

Griewank, A., Juedes, D., and Utke, J. (1996). ADOL–C, a package for the automatic differentiation of algorithms written in C/C++. *ACM Trans. Math. Software*, 22(2):131–167.

Hammer, R., Hocks, M., Kulisch, U., and Ratz, D. (1995). *C++ toolbox for verified computing I - Basic Numerical Problems*. Springer-Verlag, Heidelberg and New York.

Hörsken, C. (2003). *Methoden zur rechnergestützten Toleranzanalyse in Computer Aided Design und Mehrkörpersystemen*. PhD thesis, Univerisity of Duisburg-Essen.

Huckle, T. (2005). Collection of software bugs. `www5.in.tum.de/~huckle/bugse.html`.

Kecskeméthy, A. (1993). *Objektorientierte Modellierung der Dynamik von Mehrkörpersystemen mit Hilfe von Übertragungselementen*. PhD thesis, Gerhard Mercator Universität Duisburg.

Kecskeméthy, A. and Hiller, M. (1994). An object-oriented approach for an effective formulation of multibody dynamics. *CMAME*, 115:287–314.

Klatte, R., Kulisch, U., Wiethoff, A., Lawo, C., and Rauch, M. (1993). *C–XSC: A C++ Class Library for Extended Scientific Computing*. Springer-Verlag, Berlin Heidelberg.

Knüppel, O. (1994). PROFIL/BIAS — a fast interval library. *Computing*, 53:277–287.

Lerch, M., Tischler, G., Wolff von Gudenberg, J., Hofschuster, W., and Krämer, W. (2001). The Interval Library `filib++ 2.0` : Design, Features and Sample Programs. Technical Report 2001/4, Wissenschaftliches Rechnen / Softwaretechnologie, Bergische Universität GH Wuppertal.

Lohner, R. (1988). *Einschließung der Lösung gewönlicher Anfangs- und Randwertaufgaben und Anwendungen*. PhD thesis, Universität Karlsruhe.

Lohner, R. (2001). On the ubiquity of the wrapping effect in the computation of the error bounds. In Kulisch, U., Lohner, R., and Facius, A., editors, *Perspectives on Enclosure Methods*, pages 201–217. Springer Wien New York.

Metropolis, N. and Ulam, S. (1949). The monte carlo method. *Journal of the American Statistic Association*, 44:335–341.

Moore, R. E. (1966). *Interval Analysis*. Prentice-Hall, New York.

Nedialkov, N. S. (1999). *Computing rigorous bounds on the solution of an initial value problem for an ordinary differential equation*. PhD thesis, University of Toronto.

Nedialkov, N. S. (2002). *The design and implementation of an object-oriented validated ODE solver*. Kluwer Academic Publishers.

Neumaier, A. (2002). Taylor forms — use and limits. *Reliable Computing*, 9:43–79.

Stauning, O. and Bendtsen, C. (2005). Fadbad++. Web page `http://www2.imm.dtu.dk/~km/FADBAD/`.

Traczinski, H. (2006). *Integration von Algorithmen und Datentypen zur validierten Mehrkrpersimulation in MOBILE*. PhD thesis, Univerisity of Duisburg-Essen.

GRASP CONFIGURATION MATCHING
Using Visual and Tactile Sensor Information

Madjid Boudaba
Design Center
TES Electonic Solution GmbH, Zettachring 8, 70567 Stuttgart, Germany
madjid.boudaba@tesbv.com

Alicia Casals
GRINS: Research Group on Intelligent Robots and Systems
Technical University of Catalonia, Pau Gargallo 5, 08028 Barcelona, Spain
alicia.casals@upc.edu

Keywords: Visual image, Tactile image, Grasp planning, Block matching algorithm.

Abstract: Finding the global shape of a grasped object directly from touch is time consuming and not highly reliable. This paper describes the relationship between visual features and grasp planning, and correlates visual and tactile information for a better description of the object's shape and grasping points determination. The grasping process proposed is experimented with a three fingered robotic hand.

1 INTRODUCTION

Grasping has been an active area of robotics research in the last decades. Although a great number of sensory systems have been used to monitor and to control grasping, their usefulness is often limited by the ability of handling all aspects of detection/recognition, guidance, alignment and grasping. To place our approach in perspective, we review existing methods for sensor based grasp planning. The existing literature can be broadly classified into two categories; vision based and tactile based. For both categories, the extracted image features are of concern, they can range from geometric primitives such as edges, lines, vertices, and circles to optical flow estimates. The first category uses image features to estimate the robot's motion with respect to the object pose (Maekawa et al., 1995), (Smith and Papanikolopoulos, 1996), (Allen et al., 1999). Once the robot hands is already aligned with the object, then, it only needs to know where the fingers are placed on the object. The second category of sensor uses image features to estimate the touch sensing area in contact with the object (Berger and Khosla, 1991), (Chen et al., 1995), (Lee and Nicholls, 1999). A practical drawback is that the grasp execution is hardly reactive to sensing errors such as finger positioning errors. A vision sensor, meanwhile, is unable to handle occlusions. Since an object is grasped according to its CAD model (Kragic et al., 2001), an image also contains redundant information that could become a source of errors and inefficient in the processing.

This paper is an extension of our previous work (Boudaba et al., 2005) and (Boudaba and Casals, 2006) on grasp planning using visual features. In this work, we demonstrate its utility in the context of grasp (or fingers) positioning. Consider the problem of selecting and executing a grasp. In most tasks, one can expect various uncertainties. Grasping an object implies building a relationship between the robot hand and the object model. The latter is often unavailable or poorly known. Thus, selecting a grasp position from such model can be unprecise or unpracticable in real time applications. In our approach, we avoid using any object model and instead we work directly from edge features. In order to avoid fingers positioning errors, a sizable image blocks are defined that represent the features of grasping contact points. This not only avoids detection/localization errors but also saves computation effort that could affect the reliability of the system. Our features matching based approach can play the critical role of forcing the fingers to move to the desired positions before the task of grasping is executed. To achieve a high level of matching efficiency, the visual image is first divided into squared blocks of pixels. Then for each one of these blocks the algorithm tries to find its correspondence in the target block that is the closest to it according to a predetermined criterion. Finally, we reduce or eliminate redundant information contained in the image by transforming the result of the matching algorithm to the frequency domain. Then a compression scheme is proposed to the image coding.

The proposed work is divided into two phases:

1. **Grasp planning phase:** For each two-dimensional view of an object in a visual frame features of its contour are calculated. These features are then used as input data, both for the grasp planning and features matching phases. The grasping positions are generated in the planning task, so a relationship between visual features and grasp planning is proposed. Then a set of geometrical functions is analysed to find a feasible solution for grasping. The result of grasp planning is a database containing a set of valid grasps, the most favorable as well as those rejected.
2. **Sensor features matching phase:** Unlike vision which provides global features of the object, tactile sensor provides local features when the fingertip is in touch with the object. In order to identify and locate the features that best fit the two domains (vision and touch) of features, a contour splitting process divides the object's contour into blocks, so that different matching techniques can be applied. For the purpose of features matching, extracting edge features are of concern using the basic results from different approaches. The matching is conducted in two-dimensional space. Each edge in the block is treated as features.

2 GRASP BACKGROUND

Geometric formulation and grasp feasibility are reviewed and discussed based on (Hirai, 2002). Given a grasp which is characterized by a set of contact points and the associated contact models, the problem is determining whether the grasp has a force-closure. For finger contact, a commonly used model is point contact with friction (PCWF). In this model, fingers can exert any force pointing into the friction cone at the edge of contacts (We use edge contact instead of point contact, which can be described as a linear combination of two vectors, see Figure 1(b)). To fully analyze grasp feasibility, we need to examine the full space of forces acting on the object. Forming the convex hull of this space is difficult due to the nonlinear friction cone constraints imposed by the contact models. In this section, we only focus in precision grasps, where only the fingertips are in contact with the object. After discussing the friction cone modeling, a formalizme is used to analyze force closure grasps using the theory of polyhedral convex cones.

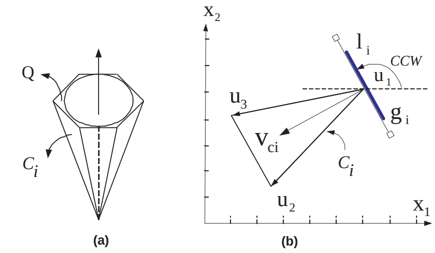


Figure 1: Friction cone modelleing.

2.1 Friction Cone Modelling

For the analysis of the contact forces in planar grasps, we simplify the problem by linearizing the friction cone by a polyhedral convex cone. In the plane, a cone has the appareance shown in Figure 1(b). This means that we can reduce the number of cone sides, *m=6* to one face.

Let's denote by P, the convex polytopes of a face cone, and $\{u_1, u_2, u_3\}$ its three representative vertices. We can define such polytopes by

$$P = \left\{ x = \sum_{i=1}^{3} \delta_i u_i : 0 \le \delta_i \le 1, \sum_{i=1}^{3} \delta_i = 1 \right\} \qquad (1)$$

2.2 Grasp Space Evaluation

The full space of a grasp is evaluated by analysing its convex hull. For a set of friction cone intersections, the full space can be defined by

$$C_1^k = C(P_1) \cap C(P_2) \cap C(P_k) \qquad (2)$$

where k is the number of grasping contacts. Note that the result of C_1^k is a set of friction cone intersections and produces either an empty set or a bounded convex polytope. Therefore, the solution of (2) can be expressed in terms of its extreme vertices

$$\Omega_1^{v_p}(U) = \left\{ \sum_{i=1}^{v_p} \alpha_i u_{ci}, \quad \sum_{i=1}^{v_p} \alpha_i = 1, \quad \alpha_i \ge 0 \right\} \quad (3)$$

where v_p is the total number of extreme vertices.

Figure 2 illustrates an example of feasible solution of $\Omega_1^{v_p}(U)$ and its grasp space represented by its extreme vertices $P = \{v_1, v_2, \ldots, v_5\}$.

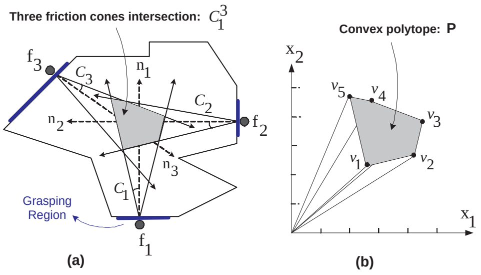


Figure 2: Feasible solution of a three-fingered grasp.

3 FEATURES-BASED GRASPING

In robotic grasping tasks, when several input sensors are available simultaneously, it is generally necessary to precisely analyze all the data along the entire grasping task. We can acquire the precise position and orientation of the target object and robot hand motion from a vision system and can acquire force, torque, and touch information from tactile sensing. The object being extracted from a video sequence requires encoding its contour individually in a layered manner and provide at the receiver's side an enhanced accessibility of visual information. In the same way, for the object being extracted from a tactile sensor, the tactile layer processes and provides the tactile information at its receiver's side. Obviously, the accuracy of this data is of significant importance for the eventual matching algorithm.

Figure 3 illustrates a layered composition of a tactile and a vision sensor. Given as input two consecutive images data S (tactile) and V (visual), the success (or simply the completion time) of the task depends on the level of processing efficiency.

3.1 Visual Features Extraction

Due to their semantically rich nature, contours are one of the most commonly used shape descriptors, and various methods for representing the contours of 2D objects have been proposed in the literature (Costa and Cesar, 2001). Extracting meaningful features from digital curves, finding lines or segments in an image is highly significant in grasping applications. Most of the available methods are variations of the dominant point detection algorithms (M. Marji, 2003). The advantage of using dominant points is that both, high data compression and feature extraction can be achieved. Other works prefer the method of polygonal approximation using linking and merging algorithms (Rosin, 1997) and curvature scale space (CSS).

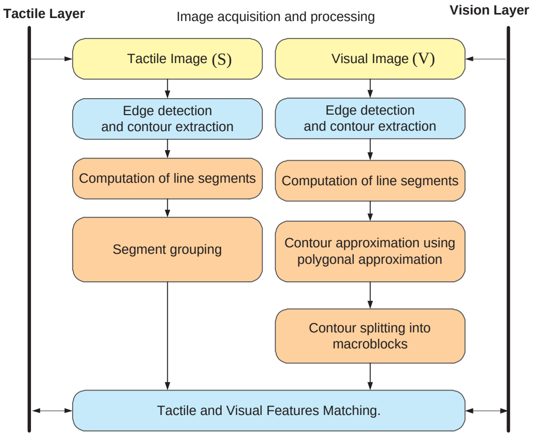

Tactile Layer
Image acquisition and processing
Vision Layer
Tactile Image (S)
Visual Image (V)
Edge detection and contour extraction
Edge detection and contour extraction
Computation of line segments
Computation of line segments
Segment grouping
Contour approximation using polygonal approximation
Contour splitting into macroblocks
Tactile and Visual Features Matching.

Figure 3: Tactile and visual features data processing.

We denote by V a function regrouping parameters of visual features defined by

$$V = \{vlist, slist, llist, com\} \tag{4}$$

where *vlist* and *slist* are the lists of consecutive contour's vertices and segments, respectively. *llist* is a list containing the parameters of segments, calculated with respect to the object's center of mass, *com*. The resulting parameters of V fully describe the two-dimensional location of features with respect to the image plane. The visual features obtained can be used as input data for both, grasp planning and features matching for grasp control.

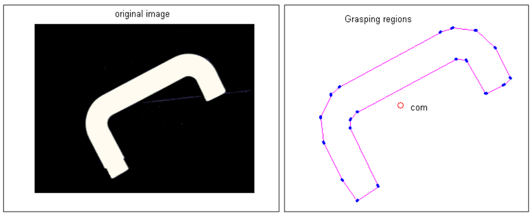


Figure 4: Visual features extraction.

3.2 Tactile Features Extraction

Unlike vision which provides global features of the object, tactile sensor provides local features when the fingertip is in touch with the object. To simplify the problem, tactile features are treated as visual features using the basic results from different approaches (Lee and Nicholls, 1999). For the purpose of sensor features matching, extracting edge features are of interest. Figure 5 illustrates three examples of tactile sensor in touch with the object. From left to right side, the sensitive area is shown with hashed region and located at upper corner side, bottom side and covering the entire object area, respectively. The tactile sensor device consists of a printed circuit board with a tactile array of electrodes (called tactels) on its surface and circuitry to deliver the tactile data of local features extracted from the sensing elements, as well as circuity to monitor and adjust the power supply voltage to the sensor. The raw data from the sensing elements is processed and produces as output a vector S containing the parameters that define tactile features.

$$S = \{elist, slist, plist\} \tag{5}$$

where *elist* and *slist* are list of consecutive contour edges and segments, respectively. *plist* is a list containing the parameters tied to segments, such as location and orientation in the tactile sensor plane.

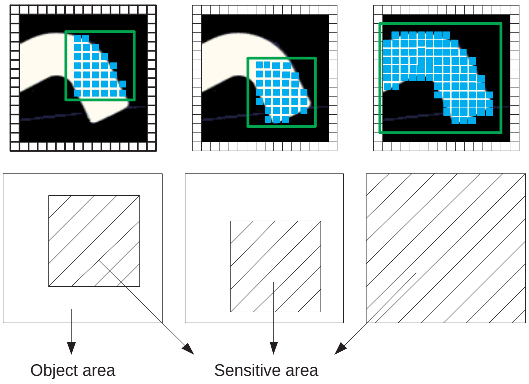


Figure 5: Sensitive area of the tactile sensor. This information can be determined by examination of the frequency domain, and is shown in Figure 10.

4 GRASP PLANNING

Grasp planning can be seen as constructing procedures for placing point contacts on the surface of a given object to achieve force-closure grasps. Taking as input the set of visual features extracted from the contour of the object, the output is a set of valid grasps. The relationship between visual features and grasp planning is given in the next section.

4.1 Grasp Point Generation

Generating a number of valid grasps from a list of candidates and classifying the best among them is quite time consuming. Thus a preprocessing (or prefiltering) is necessary before the grasping points generation takes place. A fingertip is estimated to be as a sphere with radius f_r (see Figure. 2(b)), a grasping region must be large enough for placing contact points on it. Hence, a prefiltering is applied to the list, *slist* defined in (4) to discard those segments with length less than the diameter of the sphere ($s_i < 2f_r$). Figure. 4 illustrates the result of prefiltering processes as described by the following equation:

$$glist = \{g_1, g_2, \cdots, g_m\} \tag{6}$$

where *glist* is a linked list of grasping regions and m its number.

A very important aspect of (6) is the way how knowledge about grasping regions are represented in the extraction process, knowledge that will be used for generating grasping points.

The following equation describes the relationship between the visual features and grasp planning

$$G = f(glist, gparam, com) \tag{7}$$

where *glist*, *gparam*, and *com* are the visual features observed on the image plane and G is a grasp map of outputs defined by the relationship between fingers and the location of contact points on its corresponding grasping regions. From the grasp map G three possible solutions are derived:

$$G : \begin{cases} G_s = \{G_{s_1}, G_{s_2}, \cdots, G_{s_{is}}\} \\ G_b = \{G_{b_1}, G_{b_2}, \cdots, G_{b_{ib}}\} \\ G_r = \{G_{r_1}, G_{r_2}, \cdots, G_{r_{ir}}\} \end{cases} \tag{8}$$

where G_s, G_b, and G_r are selected, best, and rejected grasp, respectively. The *is*, *ib*, and *ir* are the number of selected, best, and rejected grasps, respectively.

For a three-finger grasps, the selected grasps (G_s) are given in the following form:

$$G_s : \begin{cases} G_{s_1} = \{(f_1, g_1), (f_2, g_6), (f_3, g_9)\} \\ G_{s_2} = \{(f_1, g_2), (f_2, g_6), (f_3, g_{10})\} \\ \vdots \\ G_{s_{is}} = \{(f_1, g_1), (f_2, g_8), (f_3, g_{12})\} \end{cases}$$

where f_i and g_i are the $i-th$ finger and grasping region, respectively.

A similar form can be given for representing the best grasps G_b and those rejected G_r.

4.2 Algorithm

The grasp planning algorithm is divided into several procedures and operates as follows:

1. *Visual features procedure*
   - *Function grouping visual features using (4)*
2. *Grasping point generation procedure*
   - *Pick three grasp regions from (6)*
   - *Determine the initial position of f_1, f_2 and f_3*
   - *Compute their friction cones using (1)*
   - *Compute the friction cones intersection of (2)*
3. *Grasping test procedure*
   - *Compute the solution friction cones using (3)*
   - *Check whether the polytopes given by (3) is bounded. If so, stop and save the selected grasps to G_s.*
   - *Else save the rejected grasps to G_r.*
4. *Quality test procedure*
   - *The last step of the algorithm consists of selecting the best grasps from a range of valid grasps from lower to upper acceptance measures by using the parameters measure given in table 1. Save to G_b.*

5 FEATURES MATCHING

Our goal is to match the grasping positions correspondence between the visual and tactile sensor features. The matching process works first getting a grasping position within its searching area and next it updates the tactile features using a tactile sensor. The size of the search windows is very important when configuring a matching process. The larger the search window, the longer it takes to process the search. The matching is conducted in the pixel domain, so the contrast is necessary for identifying edges in reference (visual) and target (tactile) image. Images with weak contrast should be avoided since the matching algorithm uses the edges based searching. The weaker the contrast, the less the amount and accuracy of edge-based information with which the searching is performed. Figure 7 shows two tables in grayscale values assigned to each block of matching.

5.1 Image Subregions

In order to identify the location of the best fitting between the tactile sensor frames and visual frames, a subregion process is performed that scales down the contour image into subregions in an efficient way so that the matching algorithm can be applied. The basic principle is similar to the image (or video) compression techniques, which defines how the component (RGB, YUV or YCrCb) can be down-sampled and digitalised to form discrete pixels. The terms 4:2:2 and 4:2:0 are often used to describe the sampling structure of the digital image. 4:2:2 means the chrominance is horizontally sub-sampled by a factor of two relative to the luminance; 4:2:0 means the chrominance is horizontally and vertically sub-sampled by a factor of two relative to the luminance. In the case of a 704x576 PAL/SECAM standard for example, the QCIF (Quarter Common Immediate Format) can be obtained by scaling down the image with a factor of 4 in the horizontal/vertical direction. For a QCIF format of size 176x144, there are 25.344 pixels in the frame. A macro block defines a 16x16 pixel area (256 pixels), so there are 99 macro blocks to process (see Figure. 6).

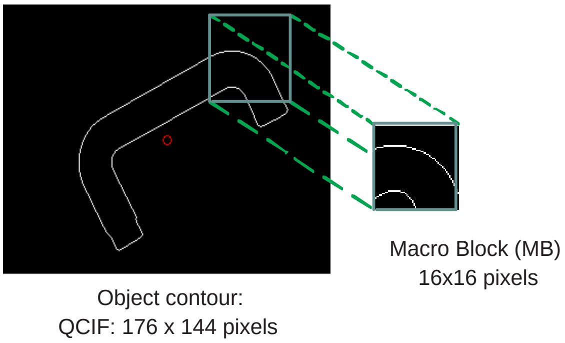


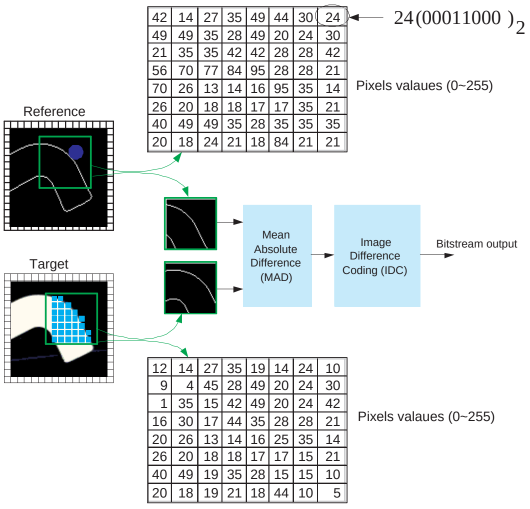
Figure 6: Image subregions.

5.2 Block-Matching Algorithm

The Mean Absolute Difference (MAD) is a well known matching criteria and widely used due to its lower computational complexity (Lu and Liou, 1997). Given two blocks represented by two set of features: $S = \{a_1, a_2, ..., a_q\}$ and $V = \{b_1, b_2, ..., b_q\}$, the corresponding features from each block are compared and their differences accumulated, as described by equation

$$MAD(dx, dy) = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} |V[i, j] - S[i + dx, j + dy]| \tag{9}$$

where $S(i, j)$ represents a $(N \times N)$ macroblock of pixel intensity in the target frame and $V(i, j)$ represents $(N \times N)$ macroblock of pixel intensity in the reference frame. (dx, dy) is a vector representing the



42	14	27	35	49	44	30	24
49	49	35	28	49	20	24	30
21	35	35	42	42	28	28	42
56	70	77	84	95	28	28	21
70	26	13	14	16	95	35	14
26	20	18	18	17	17	35	21
40	49	49	35	28	35	35	35
20	18	24	21	18	84	21	21

12	14	27	35	19	14	24	10
9	4	45	28	49	20	24	30
1	35	15	42	49	20	24	42
16	30	17	44	35	28	28	21
20	26	13	14	16	25	35	14
26	20	18	18	17	17	15	21
40	49	19	35	28	15	15	10
20	18	19	21	18	44	10	5

Figure 7: Block matching algorithm.

search area. For our application the search area is specified by $dx = (-p, p)$ and $dy = (-p, p)$ where $p = 6$ and $N = 16$.

The matching of a macroblock with another is based on the output of a cost function (9). The macroblock that results in the least cost is the one that matches the closest to current macroblock, Figure 7. For each fingertip that gets in touch with the object, the tactile features are matched to those of visual features inside a predefined searching area. A motion vector is then applied to search the features correspondence between blocks in the target frame and those in the reference frame. Figure 8 illustrates the searching method that evaluates the MAD cost function within the search area. Many other search methods can be found in (Furht, 1995).

5.3 Image Difference Coding

In order to control the grasping position, the result of the matching algorithm can be defined as error position or that so called error grasp, is then calculated using the following expression:

$$G_e(i,j) = V(i,j) - S(i+dx, j+dy) \qquad (10)$$

Since we want to guide the robot hand towards these grasping references G_{ref}, the solution consists of reducing the grasp error G_e by moving the tactile sensors towards the set of corresponding positions of grasping references. The cost of a solution is expressed as the total sum of contact displacements over the surface of the object from an initial contact configuration. If the result of matching is outside a given margin, then the grasp controller should launch a new measurement via joint angle and position sensors.

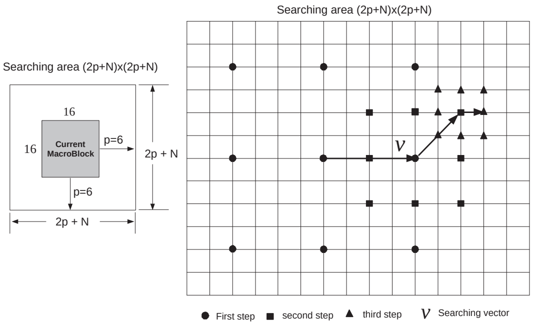
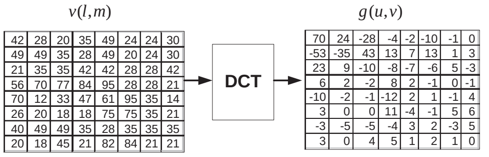


Figure 8: Searching area.

The result of block matching algorithm (see Figure 8) is a two-dimensional vector called motion vector, $v(l,m)$. The Image Difference Coding (IDC) processes these measurements and produces as output a vector image containing the parameters of grasping positions, which are compressed in a suitable format to reduce the data transmission bandwidth. The digital cosine transform (DCT) is used due to its capability of removing spatial redundancy to achieve low bit rates.



$v(l,m)$

42	28	20	35	49	24	24	30
49	49	35	28	49	20	24	30
21	35	35	42	42	28	28	42
56	70	77	84	95	28	28	21
70	12	33	47	61	95	35	14
26	20	18	18	75	75	35	21
40	49	49	35	28	35	35	35
20	18	45	21	82	84	21	21

$g(u,v)$

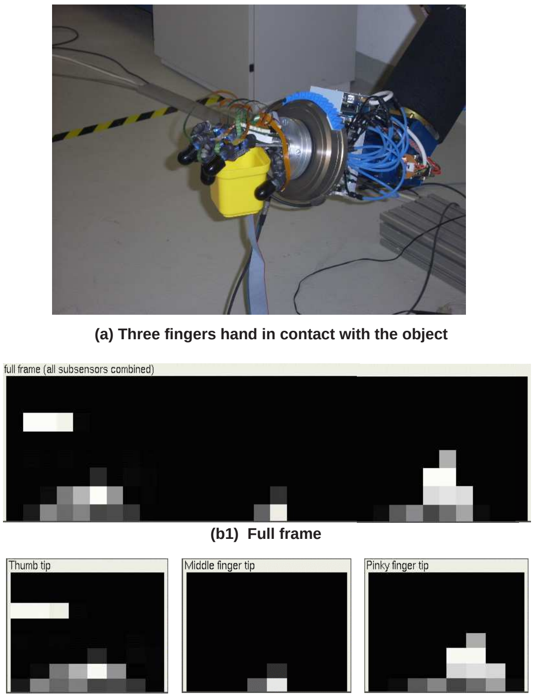
70	24	-28	-4	-2	-10	-1	0
-53	-35	43	13	7	13	1	3
23	9	-10	-8	-7	-6	5	-3
6	2	-2	8	2	-1	0	-1
-10	-2	-1	-12	2	1	-1	4
3	0	0	11	-4	-1	5	6
-3	-5	-5	-4	3	2	-3	5
3	0	4	5	1	2	1	0

Figure 9: DCT Image compression.

The DCT transforms each 8x8 block of greyscale values into a block of 8x8 spectral frequency coefficients. The energy tends to be concentrated into few significant coefficients. Other coefficients are close to zero and insignificant (see Figure 9).
Next step of IDC is to compress the frequency domain, by not transmitting (or not coding) the close-zero coefficient (insignificant coefficients) and by quantizing and coding the remaining coefficients.

6 EXPERIMENTAL RESULTS

Figure 10 illustrates our experimental system which consists of a anthropomorphic robot hand equipped with a tactile array sensor on each fingertip and a stereo vision system. The spacial resolution of the tactile sensor has 256 (16x16) sensing cells over an area of 100 square millimeter. The sensory data processing were performed using MCAGUI and SVS tools for tactile and vision data, respectively, developed at the Institute of Process Control and Robotics (Boudaba et al., 2005). Figure 10(b) shows the tactile sensor response frames. Every cell of the sensing matrix is sampled at 10 frames per second. Figure 11 and Figure 12 show the result of five grasp configurations and Table 1 resumes their parameter measures. d_1, d_2 and d_3 are distance measures of finger position f_1, f_2 and f_3 from the object's center of mass. x_1 x_2 are the coordinates of the focus point F in the plane. d is the measured distance between focus point and center of mass. R is the vector radius of the ball centered at F. The object center of mass is located at $com = (121.000, 98.000$. The angle of the friction cone, α= 17.000 for all configurations. We have implemented the grasp planning algorithms in Matlab environment for computing feasible grasping regions for three-finger hands.



(a) Three fingers hand in contact with the object

(b1) Full frame

(b2) Individual frame

Figure 10: Tactile sensor response frames.

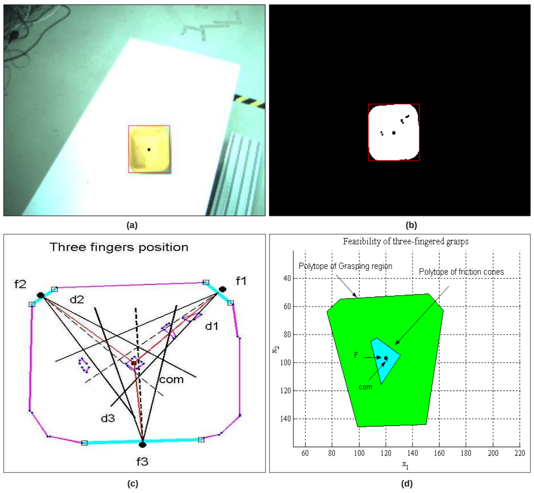


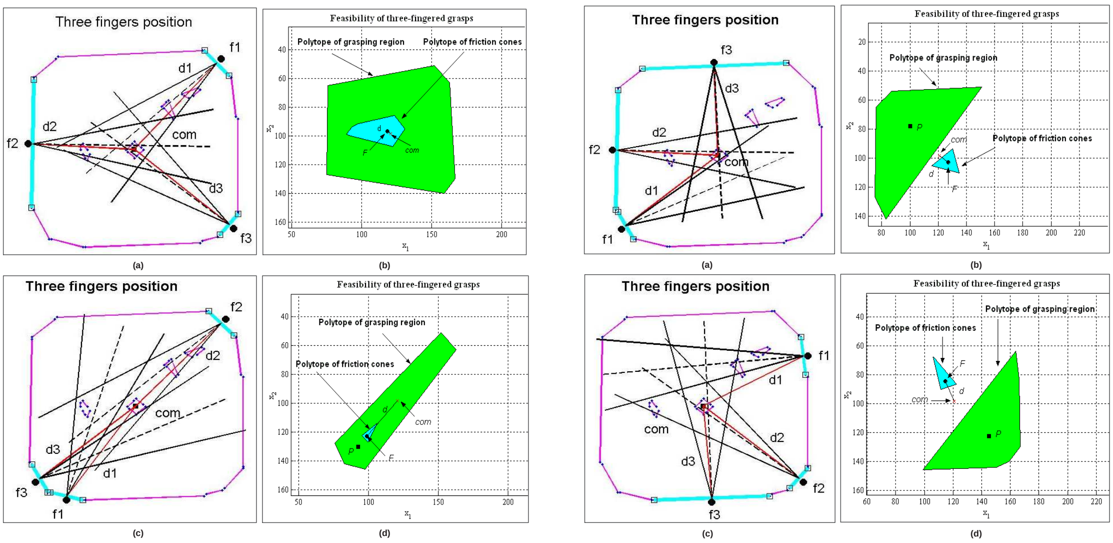
Figure 11: Grasp planning setup.

Table 1: Parameter measures of five grasp configurations.

$obj = \{GC1, GC2, GC3, GC4, GC5\}$			
d_1 d_2 d_3	$F(x_1, x_2)$	d	R
86.80 63.70 35.52	119.97 96.95	1.80	7.94
86.80 33.82 65.99	118.41 96.69	2.98	9.37
24.47 86.80 23.82	99.19 122.88	32.53	2.44
23.82 33.82 71.22	127.26 102.97	7.88	5.34
81.51 65.99 35.52	114.59 84.46	15.39	4.49

7 CONCLUSION

A framework to discuss sensor frames matching using tactile and visual features is presented in this paper. As a new approach to the grasp planning problem, the idea of vision-based grasping has been explored, within the specific context of using visual features relevant to the grasping and manipulation tasks, as complementary information to tactile data. In order to provide a suitable description of object contour, a method for grouping visual features has been proposed. Then a function defining the relationship between visual features and grasp planning has been described. A very important aspect of this method is the way knowledge about grasping regions are represented in the extraction process, which excluded all undesirable grasping points (unstable points) and all line segments that do not fit to the fingertip position. This filtering process has been used to reduce the time consumption during the process of determining suitable grasping points. For extracting the local features of the curves representing the object contour, the solution adopted is a polygonal approximation using a linking/merging algorithms.Then the force-closure condition is applied to evaluate grasping points determination. The method implemented here is currently restricted to any kind of 2D objects. In a future work, it is intended to extend our method to 3D object.The object therefore needs to be seen from different points of view which is desirable for grasp planning that performs well in the real world.



(a) Grasp Configuration(GC): 2-3

(b) Grasp Configuration(GC): 4-5

Figure 12: Result of Grasp planning with three-fingered Grasps.

ACKNOWLEDGEMENTS

The authors would like to thank Prof. Dr. H. Woern and his co-workers from the IPR institute for their support in providing the facilities and the anthropomorphic robot hand for testing the proposed approach.

REFERENCES

Allen, P., Miller, A., Oh, P., and Leibowitz, B. (1999). Integration vision, force and tactile sensing for grasping. *Int. Journal of Intell. Mechatronics*, 4(1):129–149.

Berger, A. D. and Khosla, P. K. (1991). Using tactile data for real-time feedback. *International Journal of of Robotics Research (IJR'91)*, 2(10):88–102.

Boudaba, M. and Casals, A. (2006). Grasping of planar obbjects using visual perception. In *Proc. IEEE 6th International Conference on Humanoid Robots (HUMANOIDS'06)*, pages 605–611, Genova, Italy.

Boudaba, M., Casals, A., Osswald, D., and Woern, H. (2005). Vision-based grasping point determination on objects grasping by multifingered hands. In *Proc. IEEE 6th International Conference on Field and Service Robotics (FRS'05)*, pages 261–272, Australia.

Chen, N., Rink, R. E., and Zhang, H. (1995). Edge tracking using tactile servo. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'95)*, pages 84–99.

Costa, L. and Cesar, R. (2001). *Shape Analysis and Classification Theory and Practice*. CRC Press, Florida, USA, 1st edition.

Furht, B. (1995). A survey of multimedia compression techniques and standards, part ii: Video compression. *Real-Time Imaging Journal*, 1:319–337.

Hirai, S. (2002). Kinematics of manipulation using the theory of polyhedral convex cones and its application to grasping and assembly operations. *Trans. of the Society of Inst. and Control Eng.*, 2:10–17.

Kragic, D., Miller, A., and Allen, P. (2001). Real-time tracking meets online grasp planning. In *Proc. IEEE International Conference on Robotics and Automatation (ICRA'2001)*, pages 2460–2465, Seoul, Korea.

Lee, M. H. and Nicholls, H. R. (1999). Tactile sensing for mechatronics - a state of the art servey. *Mechatronics*, 9:1–31.

Lu, J. and Liou, M. L. (1997). A simple and efficient search algorithm for block matching motion estimation. *IEEE Trans. Circuits And Systems for Video Technology*, 7:429–433.

M. Marji, P. S. (2003). A new algorithm for dominant points detection and polygonization of digital curves. *Journal of the Pattern Recognition Society*, 36:2239–2251.

Maekawa, H., Tanie, K., and Komoriya, K. (1995). Tactile sensor based manipulation of an unknown object by a multifingered hand with rolling contact. In *Proc. IEEE International Conference on Robotics and Automatation (ICRA'95)*, pages 743–750.

Rosin, P. L. (1997). Techniques for assesing polygonal approximation of curves. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 19:659–666.

Smith, C. and Papanikolopoulos (1996). Vision-guided robotic grasping: Issues and experiments. In *Proc. IEEE International Conference on Robotics and Automatation (ICRA'96)*, pages 3203–3208.

MOTION CONTROL OF AN OMNIDIRECTIONAL MOBILE ROBOT

Xiang Li and Andreas Zell
Wilhelm-Schickard-Institute, Departmant of Computer Architecture, University of Tübingen
Sand 1, 72076 Tübingen, Germany
xiang.li@uni-tuebingen.de, andreas.zell@uni-tuebingen.de

Keywords: Mobile robots and autonomous systems, system identification, actuator saturation, path following control.

Abstract: This paper focuses on the motion control problem of an omnidirectional mobile robot. A new control method based on the inverse input-output linearized kinematic model is proposed. As the actuator saturation and actuator dynamics have important impacts on the robot performance, this control law takes into account these two aspects and guarantees the stability of the closed-loop control system. Real-world experiments with an omnidirectional middle-size RoboCup robot verifies the performance of this proposed control algorithm.

1 INTRODUCTION

Recently, omnidirectional wheeled robots have received more attention in the mobile robots applications, because the omnidirecitional robots “have full mobility in the plane, which means that they can move at each instant in any direction without any reorientation ”(Campion et al., 1996). Unlike the nonholonomic robots, such as car-like robots, having to rotate before implementing any desired translation velocity, omnidirecitonal robots have higher maneuverability and are widely used in dynamic environment applications, for example, in the middle-size league of the annual RoboCup competition.

Most motion control methods of mobile robots are based on robots dynamic models (Watanabe, 1998; Liu et al., 2003; Purwin and Andrea, 2006; Tsai et al., 2006) or robots kinematic models (Muir and Neuman, 1990; Terashima et al., 2004; Rojas and Förster, 2006). A dynamic model directly describes the relationship between the forces exerted by the wheels and the robot movement, with the applied voltage of each wheel the input and the robot movement in terms of linear and angular accelerations as output. But the dynamic variations caused by the changes in the robot’s inertia moment and perturbations from the mechanic components (Scolari Conceição et al., 2005) make the controller design more complex. With the assumption that no slippage of wheels occurs, sensors have high accuracy and ground is planar enough, kinematic models are widely used in designing robots behaviors because of the simpler structures. As the inputs of kinematic models are robot wheels velocities, and outputs are the robot linear and angular velocities, the actuator dynamics of the robot are assumed fast enough to be ignored, which means the desired wheel velocities can be achieved immediately. However, the actuator dynamics limit and even degrade the robot’s performance in real situations.

Another important aspect of robot control in practice is actuator saturation. Because the commanding motor speeds of the robot’s wheels are bounded by the saturation limits, the actuator saturation can affect the robot’s performance, even make robots motion become unstable (Indiveri et al., 2006; Scolari Conceição et al., 2006).

This paper presents a motion control method for an omnidirectional robot, based on the inverse input-output linearization of the kinematic model. It takes into account not only the identified actuator dynamics but also the actuator saturation in designing a controller, and guarantees the stability of the closed-loop control system.

The remainder of this paper introduces the kinematic model of an omnidirectional middle-size Robocup robot in section 2; Path following and orientation tracking problems are solved based on the inverse input-output linearized kinematic model in section 3, where the actuator saturation is also analyzed; section 4 presents the identification of actuator dynamics and their influence on the control performance; Finally the experiment results and conclusions are discussed in sections 5 and 6, respectively.

2 ROBOT KINEMATIC MODEL

The mobile robot used in our case is an omnidirectional robot, whose base is shown in figure 1. It has three Swedish wheels mounted symmetrically with 120 degrees from each other. Each wheel is driven by a DC motor and has a same distance L from its center to the robot's center of mass R.

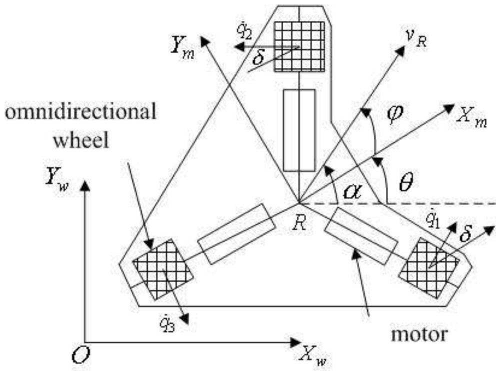


Figure 1: Kinematics diagram of the base of an omnidirectional robot.

Besides the fixed world coordinate system $[X_w, Y_w]$, a mobile robot fixed frame $[X_m, Y_m]$ is defined, which is parallel to the floor and whose origin locates at R. θ denotes the robot orientation, which is the direction angle of the axis X_m in the world coordinate system. α and φ denote the direction of the robot translation velocity v_R observed in the world and robot coordinate system, respectively. The kinematic model with respect to the robot coordinate system is given by :

$$\mathbf{v} = \begin{bmatrix} \sqrt{3}/3 & -\sqrt{3}/3 & 0 \\ 1/3 & 1/3 & -2/3 \\ 1/(3L) & 1/(3L) & 1/(3L) \end{bmatrix} \dot{\mathbf{q}}, \qquad (1)$$

where $\mathbf{v} = [\dot{x}_R^m \; \dot{y}_R^m \; \omega]^T$ is the vector of robot velocities observed in the robot coordinate system; $\dot{x}_R^m$ and $\dot{y}_R^m$ are the robot translation velocities; ω is the robot rotation velocity. $\dot{\mathbf{q}}$ is the vector of wheel velocities $[\dot{q}_1 \; \dot{q}_2 \; \dot{q}_3]^T$, and $\dot{q}_i (i = 1,2,3)$ is the i-th wheel's velocity, which is equal to the wheel's radius multiplied by the wheel's angular velocity.

Introducing the transformation matrix from the robot coordinate system to the world coordinate system as

$$^{w}R_m = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}, \qquad (2)$$

the kinematic model with respect to the world coordinate system is deduced as:

$$\dot{\mathbf{x}} = \begin{bmatrix} \frac{2}{3} cos(\theta+\delta) & -\frac{2}{3} cos(\theta-\delta) & \frac{2}{3} sin\theta \\ \frac{2}{3} sin(\theta+\delta) & -\frac{2}{3} sin(\theta-\delta) & -\frac{2}{3} cos\theta \\ \frac{1}{3L} & \frac{1}{3L} & \frac{1}{3L} \end{bmatrix} \dot{\mathbf{q}}, \qquad (3)$$

where $\dot{\mathbf{x}} = [\dot{x}_R \; \dot{y}_R \; \dot{\theta}]^T$ is the robot's velocity vector with respect to the world coordinate system; $\dot{x}_R$ and $\dot{y}_R$ are the robot translation velocities; $\dot{\theta}$ is the robot rotation velocity; δ refers to the wheel orientation in the robot coordinate system and is equal to 30 degrees.

It is important to notice that the transformation matrix in model 1 is full rank, which denotes that the translation and rotation of the robot are decoupled, and guarantees the separate control of these two movements.

For the high level control laws without considering the wheel velocities, the kinematic model

$$\dot{\mathbf{x}} = G\mathbf{v} \qquad (4)$$

is used in our control method, where the transformation matrix G is equal to $[^{w}R_m \; 0 \; ; \; 0 \; 1]$. Because G is full rank, the characteristics of decoupled movement is also kept.

3 INVERSE INPUT-OUTPUT LINEARIZATION BASED CONTROL

The trigonometric functions of angle θ in the transformation matrix G determine the nonlinearities of the kinematic model 4. Since the matrix G is full rank, this nonlinear model can be exactly linearized by introducing a simple compensator $C = G^{-1}$. The linearized system becomes $\dot{\mathbf{x}} = \mathbf{u}$ with a new input vector $\mathbf{u} = [u_1 \; u_2 \; u_3]^T$.

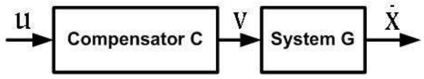


Figure 2: Linearized system by the component C.

This linear system shown in figure 2 is completely decoupled and allows the controlling of the robot's translation and rotation in a separate way. When a controller K is designed based on this simple linear system, the controller of the original system is generated as CK. The overall control loop, which consists

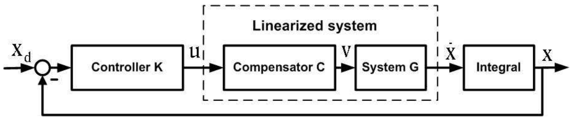


Figure 3: Closed-loop control system.

of the nonlinear system, the compensator and the controller, is shown in figure 3,

where $\mathbf{x}$ denotes the robot state vector $[x_R\ y_R\ \theta]^T$ and $\mathbf{x_d}$ is the desired state vector; x_R and y_R are robot position observed in the world coordinate system.

Based on this input-output linearized system, path following and orientation tracking problems are analyzed with respect to the robot translation and rotation control in the following subsections. The influence of actuator saturation is also accounted to keep the decoupling between the translation and rotation movements.

3.1 Path Following Control

As one high-level control problem, path following is chosen in our case to deal with the robot translation control. The path following problem is illustrated in figure 4. P denotes the given path. Point Q is the orthogonal project of R on the path P. The path coordinate system x_tQx_n moves along the path P and the coordinate axes x_t and x_n direct the tangent and normal directions at point Q, respectively. θ_P is the path tangent direction at point Q.

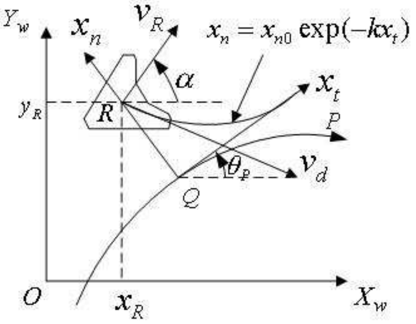


Figure 4: Illustration of the path following problem.

Based on the above definitions, the path following problem is to find proper control values of the robot translation velocity v_R and angular velocity $\dot{\alpha}$ such that the deviation distance x_n and angular error $\tilde{\theta}_R = \alpha - \theta_P$ tend to zero.

To solve this problem, a Lyapunov candidate function

$$V = \frac{1}{2}K_d x_n^2 + \frac{1}{2}K_\theta \tilde{\theta}_R^2 \quad (5)$$

can be considered, where K_d and K_θ are positive constants. The time derivation of V results in

$$\dot{V} = K_d x_n \dot{x}_n + K_\theta \tilde{\theta}_R \dot{\tilde{\theta}}_R. \quad (6)$$

Mojaev (Mojaev and Zell, 2004) presents a simple control law based on the deviation x_n, where R is controlled to move along an exponential curve and to converge to the axis X_t. The exponential curve is expressed as

$$x_n = x_{n_0} exp(-kx_t), \quad (7)$$

where x_{n_0} is the initial deviation and the positive constant k determines the convergence speed of the deviation. Differentiating (7) with respect to x_t, we get the tangent direction of the exponential curve as

$$\tilde{\theta}_R = \arctan(\frac{dx_n}{dx_t}) = \arctan(-kx_n). \quad (8)$$

Therefore, for a non-zero constant desired velocity v_d, the translation velocity of robot in the coordinate system x_tOx_n results in

$$\dot{x}_n = v_d sin\tilde{\theta}_R, \quad (9)$$

$$\dot{x}_t = v_d cos\tilde{\theta}_R. \quad (10)$$

Substituting the time derivative of $\tilde{\theta}_R$ into (6), we get

$$\dot{V} = K_d x_n \dot{x}_n + kK_\theta \arctan(-kx_n)\frac{-\dot{x}_n}{1+(kx_n)^2} < 0, \quad (11)$$

because $x_n\dot{x}_n = x_n v_d \sin(\arctan(-kx_n)) < 0$ and $\dot{x}_n \arctan(kx_n) < 0$. This solution of $\dot{V}$ guarantees the global stability of the equilibrium at $x_n = 0, \tilde{\theta}_R = 0$, which means this control law solves the path following problem.

Transforming the robot velocity into the world coordinate system, we get the control values of the linearized system as

$$u_1 = v_d \cos\alpha, \quad (12)$$

$$u_2 = v_d \sin\alpha, \quad (13)$$

where $\alpha = \tilde{\theta}_R + \theta_P$.

The input of controller 12 and 13 is the deviation distance between point R and the given path, which normally can be directly obtained by the sensors on the robot. Moreover, the deviation converges smoothly to zero with the speed controlled by parameter k, which can be chosen according to the performance requirement.

3.2 Orientation Tracking

Unlike a car-like wheeled robot, the orientation of an omnidirectional robot can be different from the direction of the robot translation velocity by any angle

φ. This relationship is denoted as $\alpha = \theta + \varphi$. That means the robot orientation can track any angle when the robot is following a given path. Based on the linearized model, the orientation tracking task is to find a suitable u_3, which is equal to the robot rotation velocity ω, such that

$$\lim_{t\to\infty}(\theta_d(t) - \theta(t)) = 0, \tag{14}$$

where $\theta_d(t)$ is the desired orientation.

As the system between input variable u_3 and output variable θ is an integrator, a commonly used PD controller can be designed to fulfill the orientation tracking task.

3.3 Actuator Saturation

Based on the inverse input-output linearization, the translation and rotation of an omnidirectional robot can be easily achieved in a separate way. This linearization is with respect to the input-output relationship, which requires the internal parts having sufficient capability to achieve the desired inputs. However, the power of the robot's motors is bounded and the actuators will saturate when the commanding velocities are too large. The presence of actuator saturation can influence the decoupling between robot translation velocity and rotation velocity, such that the system performance and stability is severely impacted. Therefore, it is necessary to deal with the actuator saturation in the controller design.

For our omnidirectional robot, the maximal velocity of each wheel is limited by $\dot{q}_m$, namely $|\dot{q}_i| \le \dot{q}_m$. Substituting the above control values from equations (12) (13) and u_3 into the inverse kinematic models (2) and (1), the wheel velocities are computed as:

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \begin{bmatrix} v_d cos(\alpha - \theta - \delta) + Lu_3 \\ -v_d cos(\alpha - \theta + \delta) + Lu_3 \\ v_d sin(\theta - \alpha) + Lu_3 \end{bmatrix}, \tag{15}$$

To achieve orientation tracking based on the above path following control, the desired translation velocity' magnitude V_d is assumed to be less than $\dot{q}_m$. Substituting $\dot{q}_m$ into (15), the lower and upper boundary of each wheel's velocity (L_{b_i} and U_{b_i}) can be calculated from the following three inequalities,

$$\begin{array}{l} |v_d cos(\alpha - \theta - \delta) + Lu_3| < \dot{q}_m \\ |-v_d cos(\alpha - \theta + \delta) + Lu_3| < \dot{q}_m \\ |v_d sin(\theta - \alpha) + Lu_3| < \dot{q}_m, \end{array} \tag{16}$$

Then the dynamic boundary values of u_3 are computed as

$$\begin{array}{l} l_b = max(l_{b_1}, l_{b_2}, l_{b_3}) \\ u_b = min(u_{b_1}, u_{b_2}, u_{b_3}), \end{array} \tag{17}$$

where l_b and u_b are the low and up boundary.

Considering the saturation function

$$x_2 = \begin{cases} u_b, & \text{if} \quad x_1 > u_b \\ x_1, & \text{if} \quad l_b \le x_1 \le u_b \\ l_b, & \text{if} \quad x_1 < l_b, \end{cases} \tag{18}$$

and its gain characteristics illustrated in figure 5, we can take the saturation function as a dynamic gain block k_a, which has maximum value one and converges to zero when the input saturates. Then the closed-loop system of controlling the robot orientation is as shown in figure 6, in which a PD controller is used to control the robot orientation converging to the ideal θ_d,

$$\omega = k_1(e_\theta + k_2\dot{e}_\theta), \tag{19}$$

where k_1 and k_2 are the proportional and derivative gains, respectively. It can be obtained that the closed-loop has only one pole $\frac{-k_a k_1}{1 + k_a k_1 k_2}$ and one zero $-1/k_2$. Therefore, when k_2 is negative and k_1 is chosen such that the pole is negative too, the stability of the closed-loop system can be guaranteed whenever k_a decreases.

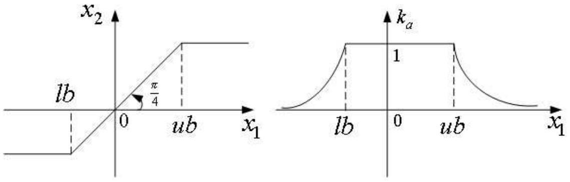


Figure 5: Saturation function and its gain characteristics.

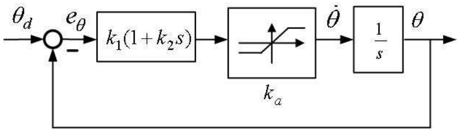


Figure 6: Closed-loop of robot orientation control.

4 ACTUATOR DYNAMICS

The results in the last section are only practical when we assume that the low level actuator dynamics is faster than the kinematics, or the delay of actuator dynamics can be ignored. It is necessary to analyze the actuator dynamics and take it into account when designing a controller. In the following subsections, the actuator dynamics is identified based on the observed input-output data, and its influence on the robot motion control is presented.

4.1 Actuator Dynamics Identification

The system identification problem is to estimate a model based on the observed input-output data such that a performance criterion is minimized. Because the full rank transformation matrix in the low level dynamics model (1) denotes the outputs $\dot{x}_R^m$, $\dot{y}_R^m$ and ω are not relevant, we identify the actuator models for these three values. The inputs of the actuator models are required velocity values($\dot{x}_{R_c}^m$, $\dot{y}_{R_c}^m$ and ω_c), and the outputs are corresponding measured values. As one commonly used parametric model, ARMAX is chosen as the identified model, which has the following structure

$$A(z)y(t) = B(z)u(t-n_k) + C(z)e(t), \tag{20}$$

$$A(z) = 1 + a_1 z^{-1} + \ldots + a_{n_a} z^{-n_a}, \tag{21}$$

$$B(z) = 1 + b_1 z^{-1} + \ldots + b_{n_b} z^{-n_b+1}, \tag{22}$$

$$C(z) = 1 + c_1 z^{-1} + \ldots + c_{n_c} z^{-n_c}. \tag{23}$$

n_k denotes the delay from input $u(t)$ to output $y(t)$. $e(t)$ is white noise. z is the shift operator resulting in $q^{-1}u(t) = u(t-1)$. n_a, n_b and n_c are the orders of polynomials $A(z)$, $B(z)$ and $C(z)$, respectively. To choose the optimal parameters of this model, we use the prediction error method, which is to find the optimal n_k and parameters of $A(z)$, $B(z)$ and $C(z)$ such that the prediction error E is minimized, namely

$$[A(z), B(z), C(z), nk]_{opt} = argmin \sum_{t=1}^{N} E^2 \tag{24}$$

$$E = y_o(t) - A^{-1}(z)(B(z)u(t-nk) + C(z)e(t)), \tag{25}$$

where $y_o(t)$ denotes the measured output data.

The system identification toolbox of Matlab has been used to identify the actuator dynamics model. Figures 7 8 and 9 show the optimal parameters and comparison between models outputs and measured outputs with respect to the actual inputs.

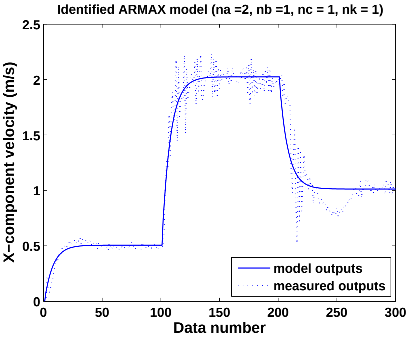


Figure 7: Identified model for $\dot{x}_R^m$.

To coincide with the robot's continuous model, the identified models are transformed from discrete ones into continuous ones using 'zoh'(zero-order hold) method,

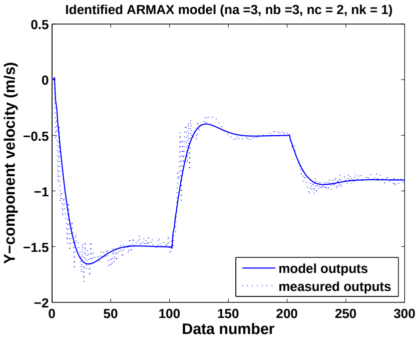


Figure 8: Identified model for $\dot{y}_R^m$.

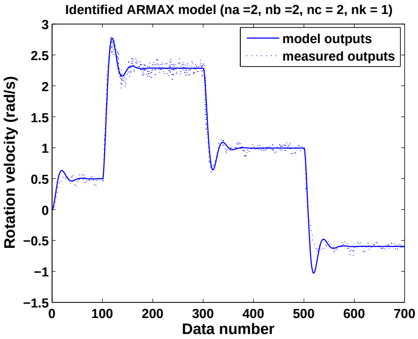


Figure 9: Identified model for ω.

$$\dot{x}_R^m = \frac{8.7948(s+58.47)}{(s+73.66)(s+6.897)} \dot{y}_{R_c}^m, \tag{26}$$

$$\dot{y}_R^m = \frac{2.4525(s+48.83)(s+6.185)}{(s+28.45)(s^2+6.837s+25.97)} \dot{y}_{R_c}^m \tag{27}$$

$$\omega = \frac{1.667(s+45.37)}{(s^2+6.759s+76.11)} \omega_c. \tag{28}$$

4.2 Actuator Influence

With consideration of the actuator, the whole structure of the control system is shown in figure 10,

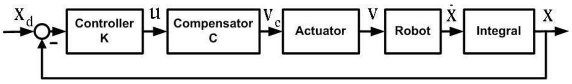


Figure 10: Closed-loop control system including actuator dynamics.

where $\mathbf{V}_c = [\dot{X}_{Rc}^m \; \dot{y}_{Rc}^m \; \omega_c]$ is the commanding robot velocity vector with respect to the robot coordinate system. Because the poles of the actuators dynamics (26) and (27) have negative real parts, these two systems are stable. That means there exits a finite short time t^*, after which the real velocities $\dot{X}_R^m$ and $\dot{Y}_R^m$ can converge to the desired ones $\dot{X}_{Rc}^m$ and $\dot{Y}_{Rc}^m$, and the inputs u_1 and u_2 begin to take effect. Therefore, the above path following law can also guarantee the robot approach to the reference path, although during t^* the deviation distance x_n and angular error $\tilde{\theta}_R$ may increase.

In the orientation tracking control, as the dynamic system (28) adds another two poles to the closed-loop system, shown in figure 11, the controller parameters decided in the above section may result the system losing the stability.

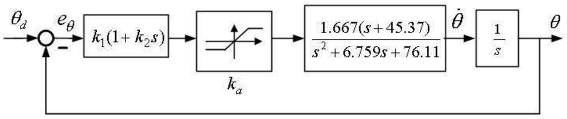


Figure 11: Closed-loop of robot orientation control including actuator dynamics.

By setting the positions of poles and zeros of the closed-loop system with the locus technique, we obtain that the conditions $k_1 > 0$ and $k_2 > 0.0515$ can guarantee the system stability, even when the actuators saturate. Figure 12 shows the root locus of an open-loop system in the critical situation with $k_2 = 0.0515$, where all the poles of closed-loop system locate in the left-half plane whatever positive value $K_a K_1$ is. Otherwise, when k_2 is less than 0.0515, the root locus may cross the imaginary axis, and the poles of closes-loop system may move to the right-half plane when k_a goes to zero.

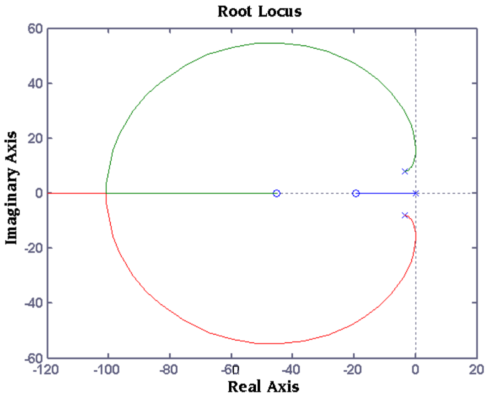


Figure 12: Root locus of open-loop model.

5 EXPERIMENT

The control algorithm discussed above has been tested in our robot laboratory having a half-field of the RoboCup middle size league. The omnidirectional robot used is shown in figure 13.

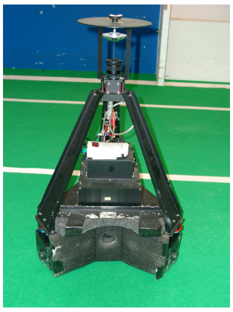


Figure 13: The real omnidirectional robot.

a AVT Marlin F-046C color camera with a resolution of 780 × 580 is assembled pointing up towards a hyperbolic mirror, which is mounted on the top of the omnidirectional robot, such that a complete surrounding map of the robot can be captured. A self-localization algorithm described in (Heinemann et al., 2004) based on the 50 Hz output signal of the camera gets the robot's position in the play field in real time. The wheels are driven by three $60W$ Maxon DC motors and the maximum wheel velocity is $1.9m/s$. Three wheel encoders measure the real wheel velocities, which are steered by three PID controllers.

An eight-shaped path is adopted as the reference path, whose geometrical symmetry and sharp changes in curvature make the test challenging. With a scale variable s, the chosen eight-shaped path is as

$$\begin{aligned} x_r &= 1.8sin(2s) \\ y_r &= 1.2sin(s), \end{aligned} \tag{29}$$

The robot was controlled to follow the eight-shaped path with a constant translation velocity $v_d = 1m/s$, and the parameters of our control algorithm were chosen as $k = 2.5,\ k_1 = 4.15,\ k_2 = 3$. The first experiment selected the path tangent direction θ_p as the desired robot orientation. Figures 14, 15, 16 and 17 show us that the proposed control method steers the robot center R converging to the given path and the robot orientation tracking the desired ones with acceptable errors, where the actuator saturation did not appear. In order to check the influence of the actuator saturation, the second experiment selected the desired robot orientation as

$$\theta_d = \theta_P + 0.9c_P v_d^2, \tag{30}$$

where c_P is the path curvature at point P. The results illustrated in figures 18, 19, 20 and 21 show us that the robot center R converges to the given path, even though the wheels velocities come in the saturation when the path turns sharply.

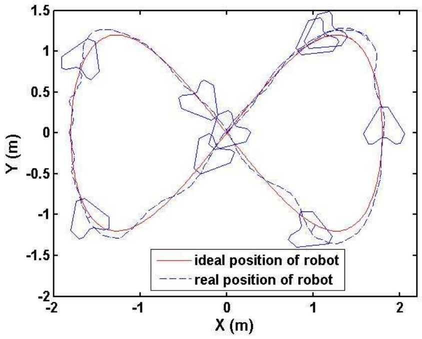


Figure 14: Reference path and robot path.

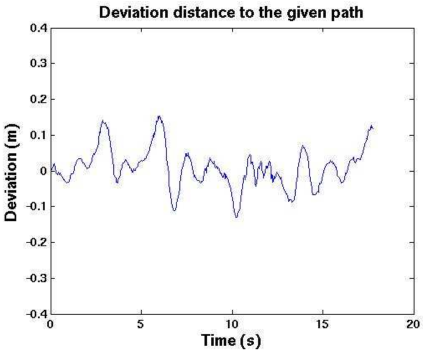


Figure 15: Distance error.

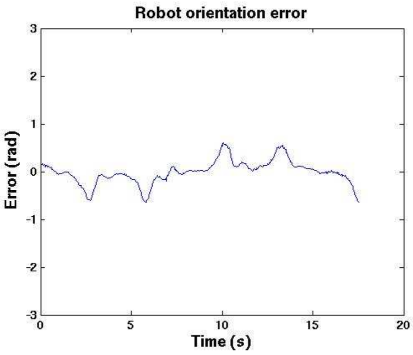


Figure 16: Orientation error.

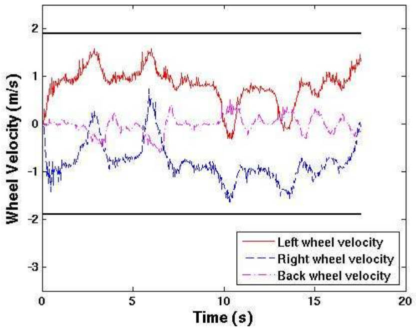


Figure 17: Real wheel velocities.

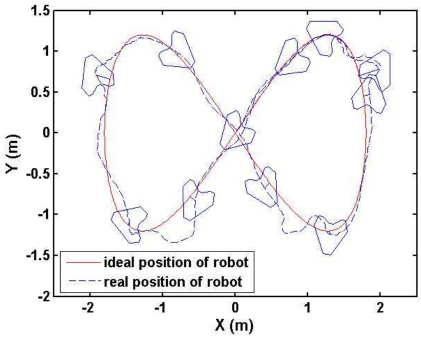


Figure 18: Reference path and robot path.

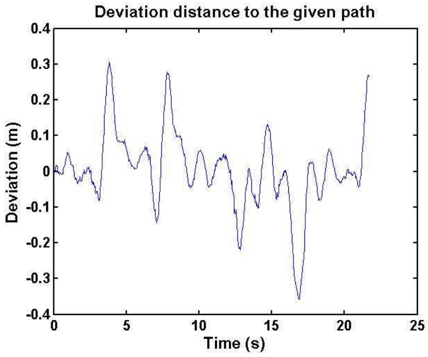


Figure 19: Distance error.

6 CONCLUSION

In this paper a new motion control method for an omnidirectional robot is presented. This approach is based on the inverse input-output linearized robot kinematic model, which completely decoupled the robot translation and rotation. The robot translation is steered to follow a reference path, and the robot rotation is controlled to track the desired orientation. Because the actuator dynamics and saturation can greatly affect the robot performance, they are taken into account when designing the controller. With the Lyapunov stability theory, the global stability of the path following control law has been proven. The locus technique is used to analyze and choose the suitable parameters of the PD controller, such that the robot orientation can converge to the desired one even

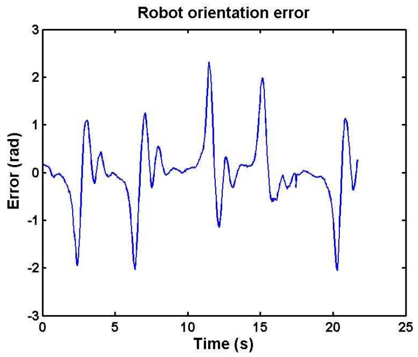


Figure 20: Orientation error.

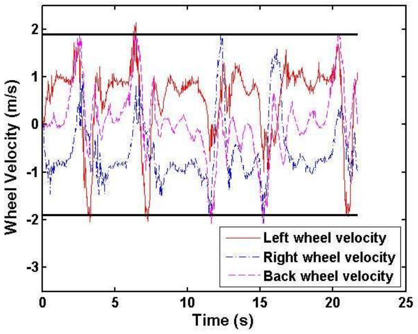


Figure 21: Real wheel velocities.

when the wheels velocities saturate.

In real-world experiments, the robot was controlled to follow an eight-shaped curve with a constant translation velocity of $1m/s$, and to track sharp changing orientations. The result shows the effectiveness of the proposed control method in the case of both actuator saturation and non-saturation.

REFERENCES

Campion, G., Bastin, G., and D'Andréa-Novel, B. (1996). Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. In *IEEE Transactions on Robotics and Automation*, volume 12, pages 47–62.

Heinemann, P., Rueckstiess, T., and Zell, A. (2004). Fast and accurate environment moddelling using omnidirectional vision. In *Dynamic Perception*. Infix.

Indiveri, G., Paulus, J., and Plöger, P. G. (2006). Motion control of swedish wheeled mobile robots in the presence of actuator saturation. In *10th annual RoboCup International Symposium*.

Liu, Y., Wu, X., Zhu, J. J., and Lew, J. (2003). Omnidirectional mobile robot controller design by trajectory linearization. In *ACC'03, Proceeding of the 2003 American Control Conference*.

Mojaev, A. and Zell, A. (2004). Tracking control and adaptive local navigation for nonholonomic mobile robot. In *Proceedings of the IAS-8 conference*.

Muir, P. F. and Neuman, C. P. (1990). Kinematic modeling for feedback control of an omnidirectional wheeled mobile robot. In *Autonomous Robot Vehicles*. Springer-Verlag.

Purwin, O. and Andrea, R. D. (2006). Trajectory generation and control for four wheeled omnidirectional vehicles. volume 54(1), pages 13–22.

Rojas, R. and Förster, A. G. (2006). *Holonomic Control of a Robot with an Omni-directional Drive*. BöttcherIT Verlag, Bremen.

Scolari Conceição, A., j. Costa, P., and Moreira, A. (2005). Control and model identification of a mobile robot's motors based in least squares and instrumental variable methods. In *MMAR'05, 11st International Conference on Metgids abd Models in Automation and Robotics*.

Scolari Conceição, A., Moreira, A., and j. Costa, P. (2006). Trajectory tracking for omni-directional mobile robots based on restrictions of the motor's velocities. In *SYROCO'06, 8th International IFAC Symposium on Robot Control*.

Terashima, K., Miyoshi, T., Urbano, J., and Kitagawa, H. (2004). Frequency shape control of omni-directional wheelchair to increase user's comfort. In *ICRA'04, Proceedings of the 2004 IEEE International Conference on Robotics and Automation*.

Tsai, C.-C., Huang, H.-C., Wang, T.-S., and Chen, C.-M. (2006). System design, trajectory planning and control of an omnidirectional mobile robot. In *2006 CACS Automatic Control Conference*.

Watanabe, K. (1998). Control of an omnidirectional mobile robot. In *KES'98, 2th International Conference on Knowledge-Based Intelligent Electronic Systems*.

IMPROVEMENT OF THE VISUAL SERVOING TASK WITH A NEW TRAJECTORY PREDICTOR
The Fuzzy Kalman Filter

C. Pérez, N. García, J. M. Sabater, J. M. Azorín and O. Reinoso
Miguel Hernández University, Avda. de la Universidad S/N, Elche, Spain
carlos.perez@umh.es

L. Gracia
Technical University of Valencia, Camino Vera S/N, Valencia, Spain
luigraca@isa.upv.es

Keywords: Visual servoing, fuzzy systems, vision / image processing, *Kalman* filter.

Abstract: Visual Servoing is an important issue in robotic vision but one of the main problems is to cope with the delay introduced by acquisition and image processing. This delay is the reason for the limited velocity and acceleration of tracking systems. The use of predictive techniques is one of the solutions to solve this problem. In this paper, we present a Fuzzy predictor. This predictor decreases the tracking error compared with the classic *Kalman* filter (KF) for abrupt changes of direction and can be used for an unknown object's dynamics. The Fuzzy predictor proposed in this work is based on several cases of the *Kalman* filtering, therefore, we have named it: Fuzzy *Kalman* Filter (FKF). The robustness and feasibility of the proposed algorithm is validated by a great number of experiments and is compared with other robust methods.

1 INTRODUCTION

During the last few years, the use of visual servoing and visual tracking has been more and more common due to the increasing power of algorithms and computers.

Visual servoing and visual tracking are techniques that can be used to control a mechanism according to visual information. This visual information is available with a time delay, therefore, the use of predictive algorithms are widely extended (notice that prediction of the object's motion can be used for smooth movements without discontinuities).

The *Kalman* filter (Kalman, 1960) has become a standard method to provide predictions and solve the delay problems (considered the predominant problem of visual servoing) in visual based control systems (Corke, 1998), (Dickmanns and V., 1988) and (Wilson and Bell, 1996).

The time delay is one of the bigger problems in this type of systems. For practically all processing architectures, the vision system requires a minimum delay of two cycles, but for on-the-fly processing, only one cycle of the control loop is needed (Chroust and Vincze, 2003).

Authors of (Chroust and Vincze, 2001) demonstrate that steady-state *Kalman* filters ($\alpha\beta$ and $\alpha\beta\gamma$ filters) performs better than the KF in the presence of abrupt changes in the trajectory, but not as good as the KF for smooth movements. Some research works about the motion estimation are presented in (S. Soatto and Perona, 1997) and (Z. Duric and Rivlin, 1996). Further, some motion understanding and trajectory planning based on the *Frenet-Serret* formula are described in (J. Angeles and Lopez-Cajun, 1988), (Z. Duric and Rosenfeld, 1998) and (Z. Duric and Davis, 1993). Using the knowledge of the motion and the structure, identification of the target dynamics may be accomplished.

To solve delay problems, taking into account these considerations, we propose a new prediction algorithm, the *fuzzy Kalman filter* (FKF). This filter minimizes the tracking error and works better than the classic KF because it decides what of the used filters ($\alpha\beta^{slow}/\alpha\beta^{fast}$ (Chroust and Vincze, 2003), $\alpha\beta\gamma$, *Kv*, *Ka* and *Kj*) must be employed. The transition between them is smooth avoiding discontinuities.

These five filters should be used in a combination because: The *Kalman* filter is considered one of the reference algorithms for position prediction (but we must consider the right model depending on the object's dynamics: velocity−acceleration−jerk). When the object is outside the image plane, the best prediction is given by steady-state filters (αβ/αβγ depending on the object's dynamics: velocity−acceleration). Obviously, considering more filters and more behaviour cases, FKF can be improved but computational cost of additional considerations can be a problem in real-time execution. These five filters are considered by authors as the best consideration (solution taking into account the prediction quality and the computational cost). This is the reason to combine these five filters to obtain the FKF.

This paper is focused on the new FKF filter and is structured as follows: in section 2 we present the considered dynamics, the considered dynamics is a Jerk model with adaptable parameters obtained by KFs (Nomura and T., 2000), (Li and Jilkov, 2000) and (Mehrotra and Mahapatra, 1997). In section 3, we present the block diagram for the visual servoing task. This block diagram is widely used in several works like (Corke, 1998) or (Chroust and Vincze, 2003). Section 4 presents the basic idea applied in our case (see (Wang, 1997b) and (Wang, 1997a)), but the main work done is focused in one of the blocks described in section 3, the Fuzzy *Kalman* Filter (FKF) is described in section 5.
In section 6, we can see the results with simulated data. These results show that FKF can be used to improve the high speed visual servoing tasks. This section is organized in two parts: in the first one (Subsection 6.1), the analysis of the FKF behaviour is focussed and in the second one (Subsection 6.2) their results are compared those with achieved by Chroust and Vince (Chroust and Vincze, 2003) and with CPA (Tenne and Singh, 2002) algorithm (algorithm used for aeronautic/aerospace applications). Conclusions and future work are presented in section 7.

2 THE DYNAMICS OF A MOVING OBJECT

The object's movement is not known (a priori) in a general visual servoing scheme. Therefore, it is treated as an stochastic disturbance justifying the use of a KF as a stochastic observer. The KF algorithm presented by Kalman (Kalman, 1960) starts with the system description given by 1 and 2.

$$x_{k+1} = F \cdot x_k + G \cdot \xi_k \tag{1}$$

$$y_k = C \cdot x_k + N \cdot \eta_k \tag{2}$$

where $x_k \in \Re^{nx1}$ is the state vector and $y_k \in \Re^{mx1}$ is the output vector. The matrix $F \in \Re^{nxm}$ is the so-called system matrix witch describes the propagation of the state from k to $k+1$ and $C \in \Re^{mxn}$ describes the way in which the measurement is generated out of the state x_k. In our case of visual servoing m is 1 (because only the position is measured) and $n = 4$. The matrix $G \in \Re^{nx1}$ distributes the system noise ξ_k to the states and η_k is the measurement noise. In the KF the noise sequences η_k and ξ_k are assumed to be gaussian, white and uncorrelated. The covariance matrices of ξ_k and η_k are Q and R respectively (these expressions consider 1D movement). A basic explanation for the assumed gaussian white noise sequences is given in (Maybeck, 1982).

In the general case of tracking, the usual model considered is a constant acceleration model (Chroust and Vincze, 2003), but in our case, we consider a constant jerk model described by matrices F and C are:

$$F = \begin{bmatrix} 1 & T & T^2/2 & T^3/6 \\ 0 & 1 & T & T^2/2 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}; C = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}$$

where T is the sampling time. This model is called a *constant jerk model* because it assumes that the jerk $(dx^3(t)/dt^3)$ is constant between two sampling instants.
F and C matrices are obtained from expression 3 to 7:

$$\frac{a - a_i}{t - t_i} = \frac{\Delta a}{\Delta t} = J_0 \tag{3}$$

$$x(t) = x_i + v_i(t - t_i) + \frac{1}{2} a_i (t - t_i)^2 + \frac{1}{6} J_i (t - t_i)^3 \tag{4}$$

$$v(t) = v_i + a_i(t - t_i) + \frac{1}{2} J_0 (t - t_i)^2 \tag{5}$$

$$a(t) = a_i + J_0(t - t_i) \tag{6}$$

$$J(t) = J_0 \tag{7}$$

where, x is the position, v is the velocity, a is the acceleration and J is the jerk. So the relation between them is:

$$x(t) = f(t);\ \dot{x}(t) = v(t);\ \ddot{x}(t) = a(t);\ \dddot{x}(t) = J(t)$$

3 DESCRIPTION OF THE CONTROL SYSTEM

The main objective of the visual servoing is to bring the target to a position of the image plane and to keep it there for any object's movement. In figure 1 we can see the visual control loop presented by Corke in (Corke, 1998). The block diagram can be used for a moving camera and for a fixed camera controlling the motion of a robot. Corke use a KF to incorporate a feed-forward structure. We incorporate the FKF algorithm in the same structure (see figure 2) but reordering the blocks for an easier comprehension.

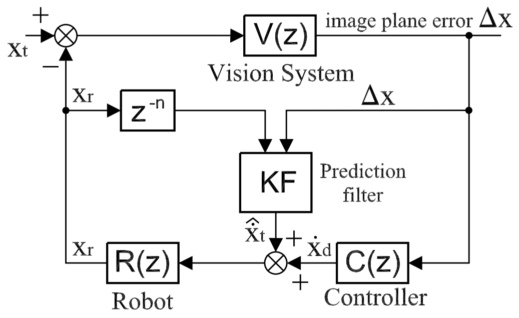


Figure 1: Operation diagram using KF presented by Corke.

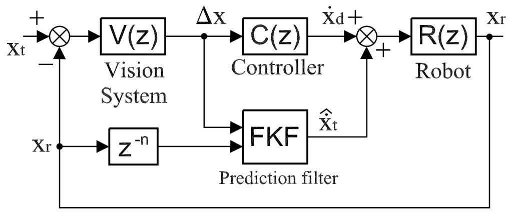


Figure 2: Operation diagram using FKF.

V(z) in figure 2 represents the camera behaviour, which is modeled as a simple delay: $V(z) = k_v \cdot z^{-2}$ (see (Corke, 1998), (Hutchinson and Corke, 1996), (Vincze and Hager, 2000), (Vincze and Weiman, 1997) and (Vincze, 2000)). C(z) is the controller (A simple proportional controller is implemented in experiments presented in this paper). R(z) is the robot (for this work: $R(z) = z/z-1$) and the *Prediction filter* generates the feedforward signal by prediction the position of the target. The variable for been minimized is Δx (generated by the vision system) that represents the deviation of the target respect to the desired position (error). The controller calculates a velocity signal $\dot{x}_d$ which moves the robot in the right direction to decrease the error. Using this approach, no path planning is needed (the elimination of this path planning is important because it decreases the computational load (Corke, 1998)).

The transfer function of the robot describes the behaviour from the velocity input to the position reached by the camera, which includes a transformation in the image plane. Therefore, the transfer function considered is (Chroust and Vincze, 2003):

$$R(z) = \frac{z}{z-1}$$

The FKF block is explained in the next sections (sections 4 and 5).

4 THEORETICAL BACKGROUND OF THE FUZZY *KALMAN* FILTER (FKF)

The most common fuzzy inference process used is known as Mamdani's fuzzy inference method, but on the other hand, we can find a so-called *Sugeno*, or *Takagi-Sugeno-Kang*, method of fuzzy inference. It was introduced in 1985 (Sugeno, 1985) and is similar to the Mamdani's method in many respects. The first two parts of the fuzzy inference process, fuzzifying the inputs and applying the fuzzy operator, are exactly the same. The main difference between Mamdani and *Sugeno* is that the *Sugeno* output membership functions are either linear or constant (for more information see (Passino and S., 1988)).

For *Sugeno* regulators, we have a linear dynamic system as the output function so that the i^{th} rule has the form:

If $\tilde{z_1}$ is $\tilde{A}_1^j$ and $\tilde{z_2}$ is $\tilde{A}_2^k$ and, ..., and $\tilde{z_p}$ is $\tilde{A}_p^l$ Then $\dot{x^i}(t) = U_i x(t) + V_i u(t)$
where $x(t) = [x_1(t), x_2(t), ..., x_n(t)]^T$ is the state vector, $u(t) = [u_1(t), u_2(t), ..., u_m(t)]^T$, U_i and V_i are the state and input matrices and $z(t) = [z_1(t), z_2(t), ..., z_p(t)]^T$ is the input to the fuzzy system, so:

$$\dot{\mathrm{x}}(\mathrm{t}) = \frac{\sum_{i=1}^{R} (U_i x(t) + V_i u(t)) \mu(z(t))}{\sum_{i=1}^{R} (\mu(z(t))}$$

or

$$\dot{\mathrm{x}}(\mathrm{t}) = \left(\sum_{i=1}^{R} (U_i \xi_i(z(t)) \right) x(t) + \left(\sum_{i=1}^{R} (V_i \xi_i(z(t)) \right) u(t)$$

where

$$\xi^{\mathrm{T}} = [\xi_1, ..., \xi_{\mathrm{R}}] = \frac{1}{\sum_{i=1}^{R} \mu_i} [\mu_1, ..., \mu_R]$$

Our work is based on this idea and these expressions (see (Passino and S., 1988) for more details). We have mixed the Mamdani's and the *Sugenos*'s idea because we have implemented an algorithm similar to *Sugeno* but not for linear systems. We obtain a normalized weighting of several non linear recursive expressions. The system works like we can see in figure 3 (see section 5).

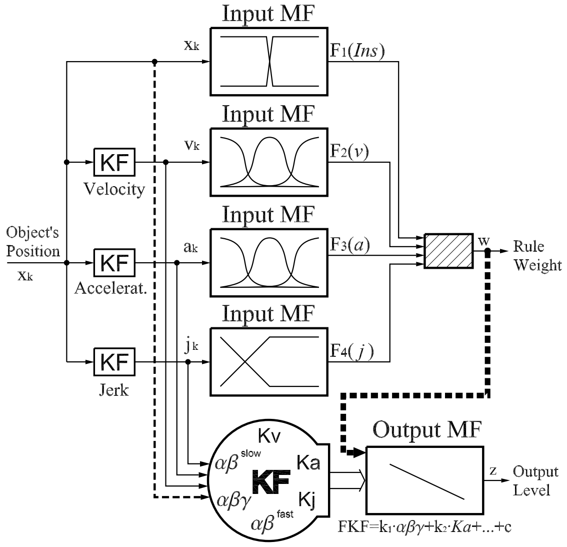


Figure 3: Fuzzy *Kalman* Filter proposed – FKF.

5 THE FUZZY *KALMAN* FILTER (FKF)

We have developed a new filter that mixes different types of *Kalman* filters depending on the conditions of the object's movement. The main advantage of this new algorithm is the non-abrupt change of the filter's output.

Consider the nonlinear dynamic system

$$\dot{x} = f_1(x,u); \quad y = g_1(x,u)$$

as each one of the filters used. The application of the fuzzy regulator in our case produces the next space-state expression:

$$\sum_{i=1}^{N} f_i(x,u) \cdot \omega(x,u)$$

where

$$\omega(\mathrm{x},\mathrm{u}) = \frac{\mu_i(x,u)}{\sum_{i=1}^{N} \mu_j(x,u)}$$

The final system obtained has the same structure than filters used:

$$\dot{x} = f_2(x,u); \quad y = g_2(x,u)$$

Figure 3 shows the FKF block diagram. In this figure, we can see that the general input is the position sequence of the target (x_k). Using this information, we estimate the velocity, acceleration and jerk of the target in three separate KFs (Nomura and Naito present the advantages of this hybrid technique in (Nomura and T., 2000)). This information is used as 'Input MF' to obtain F_1(*Ins*), F_2(*v*), F_3(*a*) and F_4(*j*). These MF inputs are the fuzzy membership functions defined in figure 4. The biggest KF block (rounded) shown in his figure is a combination of all used algorithms in the fuzzy filter ($\alpha\beta^{slow}$ and $\alpha\beta^{fast}$ (Chroust and Vincze, 2003), $\alpha\beta\gamma$, *Kv*, *Ka* and *Kj*). This block obtains the output of all specified filters. The 'Output MF' calculates the final output using the R_i rules.

Now, we present the rules (R_i) considered for the fuzzy filter:

R_1: IF object IS inside AND velocity IS low AND acceleration IS low AND jerk IS low THEN FKF=*Kv*

R_2: IF object IS inside AND velocity IS medium AND acceleration IS low AND jerk IS low THEN FKF=*Kv*

R_3: IF object IS outside AND velocity IS low AND acceleration IS low AND jerk IS low THEN FKF=$\alpha\beta^{slow}$

R_4: IF object IS outside AND velocity IS medium AND acceleration IS low AND jerk IS low THEN FKF=$\alpha\beta^{fast}$

R_5: IF object IS inside AND velocity IS high AND acceleration IS low AND jerk IS low THEN FKF=*Kv*

R_6: IF object IS inside AND acceleration IS medium AND jerk IS low THEN FKF=$0.2 \cdot \alpha\beta\gamma + 0.8 \cdot Ka$

R_7: IF object IS outside AND acceleration IS medium AND jerk IS low
THEN FKF=$0.8 \cdot \alpha\beta\gamma + 0.2 \cdot Ka$

R_8: IF object IS inside AND acceleration IS high AND jerk IS low
THEN FKF=*Ka*

R_9: IF object IS outside AND acceleration IS high AND jerk IS low
THEN FKF=$\alpha\beta\gamma$

R_{10}: IF jerk IS high THEN FKF=Kj

These rules have been obtained empirically, based on the authors experience using the Kalman filter in different applications.

Notice that rule R_{10} (when jerk is high) shows that the best filter considered is Kj and it does not depend on the object's position (inside or outside) velocity/acceleration value (low, medium or high).

We have used a product inference engine, singleton fuzzifier and centre average defuzzifier. Figure 4 presents the fuzzy sets definition where (u_{max}, v_{max}) is the image size, $\mu_{vel} = \mu_{acc} = 2m/s$, $\sigma_{vel} = \sigma_{acc} = 0.5$, $c_{vel} = c_{acc} = 1$, $d_{vel} = d_{acc} = 3$, $i_{vel} = i_{acc} = 1$ and $j_{vel} = j_{acc} = 1$ (these values have been empirically obtained).

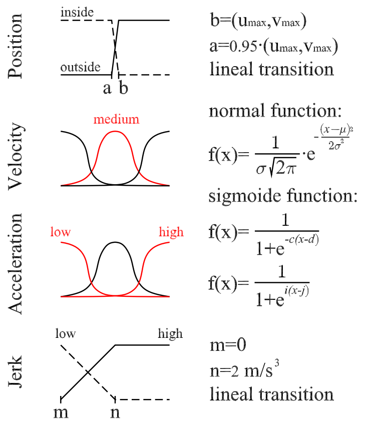


Figure 4: Parameter definition of the fuzzy system.

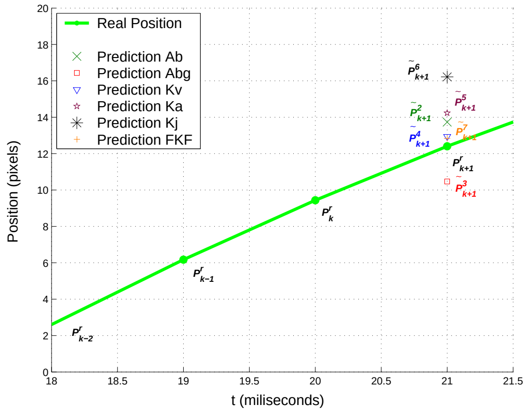


Figure 5: Real position *vs.* prediction.

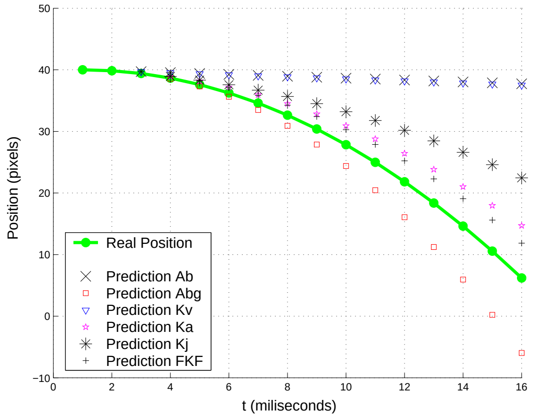


Figure 6: Prediction of a smooth trajectory.

6 RESULTS

This section is composed by two different parts: first (section 6.1), we analyze the prediction algorithm presented originally in this paper (FKF block diagram shown in figure 3) and second (section 6.2), some simulations of the visual servoing scheme (see figure 2) are done including the FKF algorithm.

6.1 Fuzzy *Kalman* Filter (FKF) Results

In figure 5, we show the effectiveness of our algorithm's prediction compared with the classical KF methods. In this figure, we can see positions P^r_k (actual object position), P^r_{k-1} (object position in $k-1$) and P^r_{k-2} (object position in $k-2$). Next real position of the object will be P^r_{k+1}, and points from $\widetilde{P}^1_{k+1}$ to $\widetilde{P}^6_{k+1}$, represent the prediction obtained by each single filter. The best prediction is given by the FKF filter. This experiment is done for a parabolic trajectory of an object affected by the gravity acceleration. (See figures 5 and 6).

We have done a lot of experiments for different movements of the object and we have concluded that our FKF algorithm works better than the others filters compared (filters compared are: αβ, αβγ, *Kv*, *Ka*, *Kj* and CPA -see section 6.2- with our FKF). Figure 6 shows the real trajectory and the trajectory predicted for each filter. For this experiment, we have used the first four real positions of the object as input for all filters and they predict the trajectory using only this information. As we can see in this figure, the best prediction is again the FKF.

6.2 Visual Servoing Control Scheme Results

To prove the control scheme presented in figure 2, we have used the object motion shown in figure 7 (up). This target motion represents a ramp-like motion between $1 < t < 4$ seconds and a sinusoidal motion for $t > 6$ seconds. This motion model is corrupted with a noise of σ=1 pixel. This motion is used by Stefan Chroust and Markus Vincze in (Chroust and Vincze, 2003) to analyze the *switching Kalman filter* (SKF).

For this experiment, we compare the proposed filter (FKF) with a well known filter, the *Circular Prediction Algorithm* (CPA) (Tenne and Singh, 2002). In figure 7 (down), we can see the results of FKF and CPA algorithms. For changes of motion behaviour, the FKF produce less error than CPA. For the change in t=1, the FKF error is [+0.008,-0] and [+0.015,-0.09] for the CPA. For the change in t=4, FKF error =

Table 1: Numerical comparative for dispersion value of all filters implemented (bounce of a ball experiment).

Init. pos.	αβ	αβγ	*Kv*	*Ka*	*Kj*	*FKF*
40	0.619	0.559	0.410	0.721	0.877	0.353
40(bis)	0.547	0.633	0.426	0.774	0.822	0.340
50	0.588	0.663	0.439	0.809	0.914	0.381
70	0.619	0.650	0.428	0.700	0.821	0.365
90	0.630	0.661	0.458	0.818	0.857	0.343
150	0.646	0.682	0.477	0.848	0.879	0.347

[+0,-0.0072] and CPA error = [+0.09,-0.015]. For the change in t=6, FKF error = [+0.022,-0] and CPA error = [+0.122,-0.76]. For the region $6 < t < 9$ (sinusoidal movement between 2.5m and 0.5m) both algorithms works quite similarly: FKF error = [±0.005] and CPA error = [±0.0076]. CPA filter works well because it is designed for movements similar to a sine shape, but we can compare this results with the SKF filter proposed in (Chroust and Vincze, 2003) and SKF works better (due to the AKF (Adaptive Kalman Filter) effect). Therefore, the FKF filter proposed works better than CPA for all cases analyzed but comparing FKF with SKF, FKF is better for t=1, t=4 and t=6 but not for $6 < t < 9$ (sinusoidal movement).

Figure 9 shows the zoom region $0 < t < 2$ and $-0.02 < \Delta x_p < 0.02$ of the same experiment. In this figure, we can see the fast response of the FKF proposed.

6.3 Experimental Results

Experimental results are obtained for this work using the following setup: Pulnix GE series high speed camera (200 frames per second), Intel PRO/1000 PT Server Adapter card, 3.06GHz Intel processor PC computer, *Windows XP Professional* O.S. and *OpenCV* blob detection library.

For this configuration, the bounce of a ball on the ground is processed to obtain data shown in figure 10. Results of this experiment are presented in table 1. In this table, we can see the dispersion of several filters. The FKF dispersion is less than αβ, αβγ, *Kv*, *Ka* and *Kj* although FKF is a combination of them. This table contains data from this particular experiment (the bounce of a ball on the ground). For this experiment, the position of the ball is introduced to the filters to prove the behaviour of them. The filter proposed (FKF) is the best analyzed.

In figure 11 we can see some frames of the experiment 'bounce of a ball on the ground'. For each frame the center of gravity of the tennis ball is obtained.

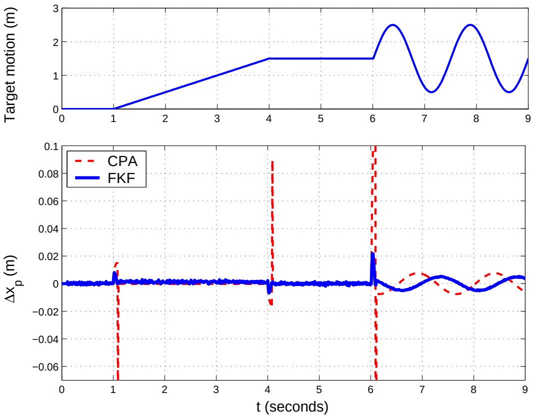


Figure 7: Simulation result for tracking an object.

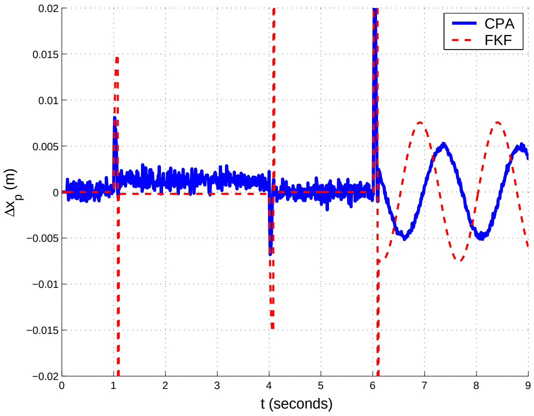


Figure 8: Zoom of the simulation.

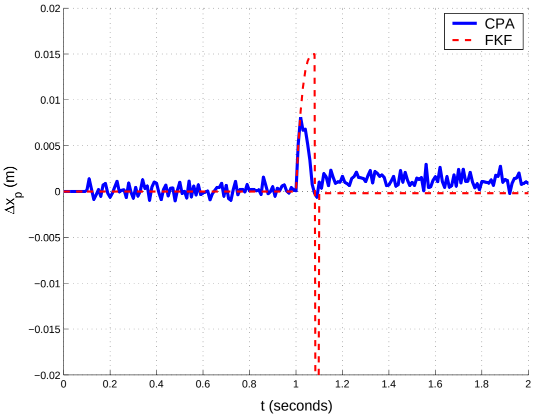


Figure 9: Zoom between 0 and 2 seconds.

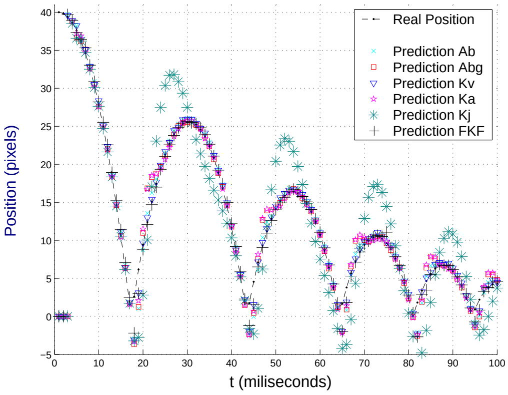


Figure 10: Bounce of the ball on the ground. Data.

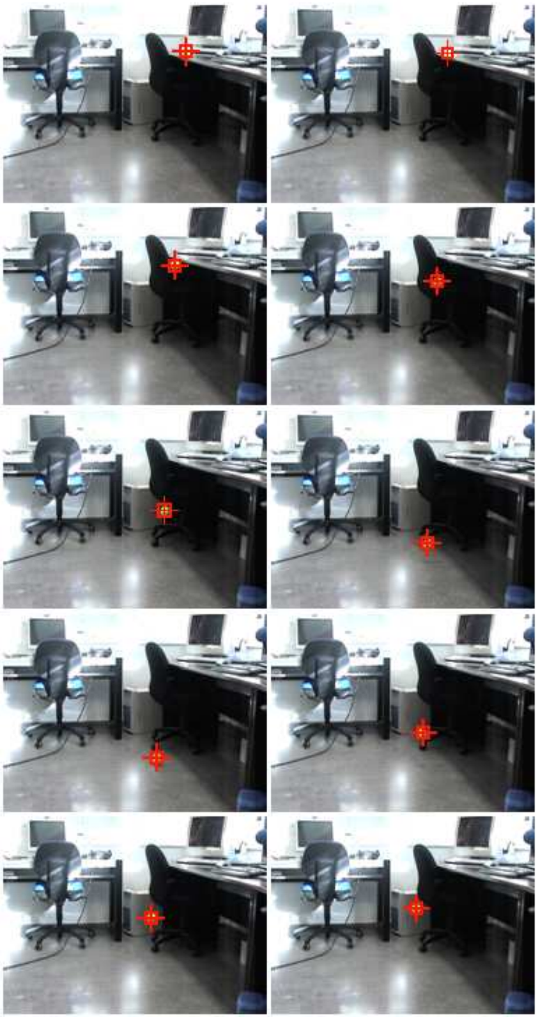


Figure 11: Bounce of the ball on the ground. Frames.

7 CONCLUSIONS AND FUTURE WORK

In section 6.1 (figures 5 and 6), we can see the quality of the new filter presented (FKF) which shows good behaviour for smooth and discontinuous motions. The object's position is estimated even when it is inside the image plane and when it is outside the image plane. Therefore, combine classic filters (KF) when inside and steady-state filters ($\alpha\beta/\alpha\beta\gamma$) when outside.

We have compared our filter with $\alpha\beta$, $\alpha\beta\gamma$, Kv, Ka and Kj in experiments of pure prediction. We have compared too, our filter with *Circular Prediction Algorithm* (CPA) in this paper reproducing the same experiment as (Chroust and Vincze, 2003) for a direct comparison with the work done by Chroust and Vincze. The filter proposed works very well but not better than SKF for all conditions, therefore, the addition of a AKF action can improve the filter behaviour (future work).

The FKF is evaluated with a ramp-like and sinosoidal motions. Δx_p is reduced in all tests done and the overshoot is decreased significantly.

Results presented in this paper are obtained for $C(z) = K_P$. Other controllers like PD, PID, ... will be implemented in future work.

ACKNOWLEDGEMENTS

This work is supported by the *Plan Nacional de I+D+I 2004-2007*, DPI2005-08203-C02-02 of the Spanish Government (*Técnicas Avanzadas de Teleoperación y Realimentación Sensorial Aplicadas a la Cirugía Asistida por Robots*).

REFERENCES

Chroust, S. and Vincze, M. (2003). Improvement of the prediction quality for visual servoing with a switching kalman filter. *I. J. Robotic Res.*, 22(10-11):905–922.

Chroust, S., Z. E. and Vincze, M. (2001). Pros and cons of control methods of visual servoing. In *In Proceedings of the 10th International Workshop on Robotics in the Alpe-Adria-Danube Region*.

Corke, P. I. (1998). *Visual Control of Robots: High Performance Visual Visual Servoing*. Research Studies Press, Wiley, New York, 1996 edition.

Dickmanns, E. D. and V., G. (1988). Dynamic monocular machine vision. In *Applications of dinamyc monoclar machine vision*. Machine Vision and Applications.

Hutchinson, S., H. G. D. and Corke, P. (1996). Visual servoing: a tutorial. In *Transactions on Robotics and Automation*. IEEE Computer Society.

J. Angeles, A. R. and Lopez-Cajun, C. S. (1988). Trajectory planning in robotics continuous-path applications. In *Journal of Robotics and Automation*. IEEE Computer Society.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*. IEEE Computer Society.

Li, X. and Jilkov, V. (2000). A survey of maneuvering target tracking: Dynamic models. In *Signal and Data Processing of Small Targets*. The International Society for Optical Engineering.

Maybeck, P. S. (1982). *Stochastic Models, Estimation and Control*. Academic Press, New York.

Mehrotra, K. and Mahapatra, P. R. (1997). A jerk model for tracking highly maneuvering targets. In *Transactions on Aerospace and Electronic Systems*. IEEE Computer Society.

Nomura, H. and T., N. (2000). Integrated vsual servoing sysem to grasp industrial parts moving on conveyer by controlling 6dof arm. In *Internacional Conference on Systems, Man. and Cybernetics*. IEEE Computer Society.

Passino, K. M. and S., Y. (1988). *Fuzzy Control*. Addison-Wesley, Ohio, USA.

S. Soatto, R. F. and Perona, P. (1997). Motion estimation via dynamic vision. In *IEEE Transactions on Automatic Control*. IEEE Computer Society.

Sugeno, M. (1985). *Industrial applications of fuzzy control*. Elsevier Science Publications Company.

Tenne, D. and Singh, T. (2002). Circular prediction algorithms-hybrid filters. In *American Control Conference*. IEEE Computer Society.

Vincze, M. (2000). Real-time vision, tracking and control-dynamics of visual servoing. In *International Conference on Robotics and Automation*. IEEE Computer Society.

Vincze, M. and Hager, G. D. (2000). *Robust Vision for Vision-Based Control of Motion*. SPIE Press / IEEE Press, Bellingham, Washington.

Vincze, M. and Weiman, C. (1997). On optimizing window-size for visual servoing. In *International Conference on Robotics and Automation*. IEEE Computer Society.

Wang, L.-X. (1997a). *Course In Fuzzy Systems and Control, A.* Prentice Hall.

Wang, L.-X. (1997b). *Course in Fuzzy Systems and Control Theory*. Pearson US Imports & PHIPEs. Pearson Higher Education.

Wilson, W. J., W. H. C. C. and Bell, G. S. R. (1996). Relative end-effector control using cartesian position based visual servoing. In *IEEE Transactions on Robotics and Automation*. IEEE Computer Society.

Z. Duric, E. R. and Davis, L. (1993). Egomotion analysis based on the frenet-serret motion model. In *Proceedings of the 4th International Conference on Computer Vision*. IEEE Computer Society.

Z. Duric, E. R. and Rosenfeld, A. (1998). Understanding the motions of tools and vehicles. In *Proceedings of the Sixth International Conference on Computer Vision*. IEEE Computer Society.

Z. Duric, J. A. F. and Rivlin, E. (1996). Function from motion. In *Transactions on Pattern Analysis and Machine Intelligence*. IEEE Computer Society.

OBSTACLE DETECTION IN MOBILE OUTDOOR ROBOTS
A Short-term Memory for the Mobile Outdoor Platform RAVON

H. Schäfer, M. Proetzsch and K. Berns
Robotics Research Lab, University of Kaiserslautern, P.O. Box 3049, D-67653 Kaiserslautern, Germany
b_schaef@informatik.uni-kl.de, proetzsch@informatik.uni-kl.de

Keywords: Obstacle Detection, Obstacle Avoidance, Behaviour-based Navigation, Mobile Robotics, Short-term Memory, Distributed Minimal World Model, Algorithms, Biologically Inspired Robotics.

Abstract: In this paper a biologically inspired approach for compensating the limited angle of vision in obstacle detection systems of mobile robots is presented.
Most of the time it is not feasible to exhaustively monitor the environment of a mobile robot. In order to nonetheless achieve safe navigation obstacle detection mechanisms need to keep in mind certain aspects of the environment. In mammals this task is carried out by the creature's short-term memory. Inspired by this concept an absolute local map storing obstacles in terms of representatives has been introduced in the obstacle detection and avoidance system of the outdoor robot RAVON. That way the gap between the fields of vision of two laser range finders can be monitored which prevents the vehicle from colliding with obstacles seen some time ago.

1 INTRODUCTION

In recent years complex mobile robotic systems have been developed for autonomous navigation in various environments. The more complex a robot's working place gets the more sophisticated its obstacle detection and avoidance facilities need to designed. First of all, sensor equipment must be appropriate for the requirements of given scenarios. In most cases such sensors have a limited field of vision and therefore cannot cover the complete area around the robot. One possibility to overcome this limitation is mounting several devices of the same kind. However, this leads to higher weight and costs and the evaluation of more data might exceed the computational capacity.

In mammals the short-term memory is used to compensate for the relatively small field of vision. The hindlegs of a cat for example have to avoid obstacles some time after the cat has visually perceived the hindrance (McVea and Pearson, 2006). Clearly the cat has to keep in mind certain information about its local environment in order to safely navigate complex terrain.

In this paper an approach for adopting the principle of a local short-term obstacle memory for mobile outdoor robots is presented. The implemented system is used to gather additional information for the behaviour-based control system of the robot RAVON (Robust Autonomous Vehicle for Outdoor Navigation, see Figure 1). For now data of two laser scanners – mounted at the front and the rear of the robot – is evaluated and inserted into the obstacle memory.

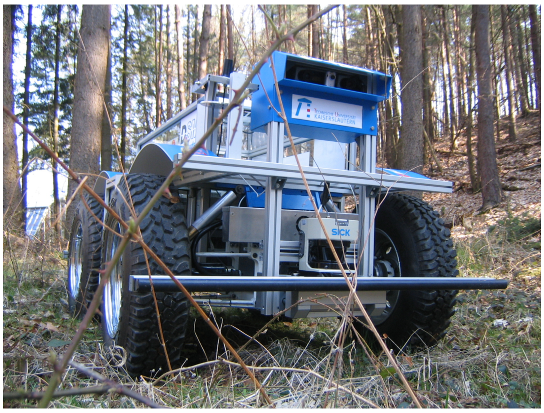


Figure 1: The outdoor robot RAVON in rough terrain.

The approach, however, is kept in a generic way such that additional sensor systems as e.g. stereo camera systems can easily be integrated.

The outline of the paper is as follows: In the next section work related to the topic of this paper is described. Afterward we give some details on the outdoor robot RAVON which is used for validating the presented approach. Section 4 describes the concept of the short-time memory including some details about the obstacle detection methods. Section 5 shows an experiment used for evaluating the presented approach. Finally, we conclude with a summary and directions for future work.

2 STATE OF THE ART

Obstacle avoidance in outdoor terrain is a topic of several publications. Mostly there is the distinction between reactive obstacle avoidance and building up complete geometric maps. Reactive approaches like (Badal et al., 1994) compute steering vectors according to the proximity of obstacles in the current view. However, for vehicles supporting agile steering manoeuvres like sideward motion this neglects the problem of possible collisions outside the field of view. Other similar approaches ((Kamon et al., 1996), (Laubach and Burdick, 1999)) add some kind of state to the obstacle avoidance system but do not explicitly consider positions of hidden obstacles.

Work describing building up maps which contain information about terrain elevation ((Shiller, 2000), (Bonnafous et al., 2001)) also neglect the need to enlarge the virtual field of vision. A high degree of computation is used to evaluate the traversability of the detected terrain region and to calculate a feasible trajectory. However, in outdoor terrain given paths cannot be followed precisely due to disturbances. Therefore, an evaluation of possibly dangerous obstacles needs to be undertaken during the driving manoeuvre.

The approach presented here can be seen in between the completely reactive and the mapping approach. A short-term memory keeping only the relevant information deals as the source for a behaviour-based system keeping the robot away from currently relevant obstacles.

3 VEHICLE DESCRIPTION

The platform used to examine obstacle avoidance is the four wheeled off-road vehicle RAVON (see Figure 1). It measures 2.35 m in length and 1.4 m in width and weighs 400 kg. The vehicle features a four wheel drive with independent motors yielding maximal velocities of 3 m/s. In combination with its off-road tires, the vehicle can climb slopes of 100% inclination predestining it for the challenges in rough terrain. Front and rear axis can be steered independently which supports agile advanced driving manoeuvres like double Ackerman and parallel steering.

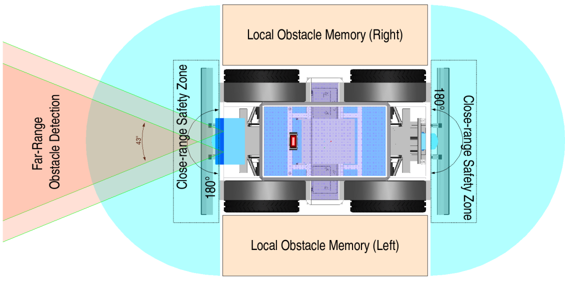

Local Obstacle Memory (Right)
Far-Range Obstacle Detection
43°
Close-range Safety Zone
180°
180°
Close-range Safety Zone
Local Obstacle Memory (Left)

Figure 2: Regions monitored by the obstacle avoidance facilities.

In order to navigate in a self-dependent fashion, RAVON has been equipped with several sensors. For self localisation purposes, the robot uses its odometry, a custom design inertial measurement unit, a magnetic field sensor, and a DGPS receiver. The sensor data fusion is performed by a Kalman filter (Schmitz et al., 2006) which calculates an estimated pose in three dimensions.

In order to protect the vehicle in respect to obstacles, several safety regions are observed by different sensor systems (Schäfer and Berns, 2006) (see Fig. 2). First of all, hindrances can be detected using the stereo camera system mounted at the front of the vehicle. This obstacle detection facility is complemented with two laser range finders (field of vision: 180 degrees, angular resolution: 0.5 degrees, distance resolution: about 0.5 cm) monitoring the environment nearby the vehicle. Data from both sources of proximity data is used for obstacle avoidance by appropriate behaviours, the fusion of which is performed inside the behaviour network (Schäfer et al., 2005). In case of emergency, the system is stopped on collision by the safety bumpers which are directly connected to the emergency stop to ensure maximal safety. In the future, the compression of the spring system shall be used to detect occluded obstacles in situations where geometric obstacle detection cannot be used.

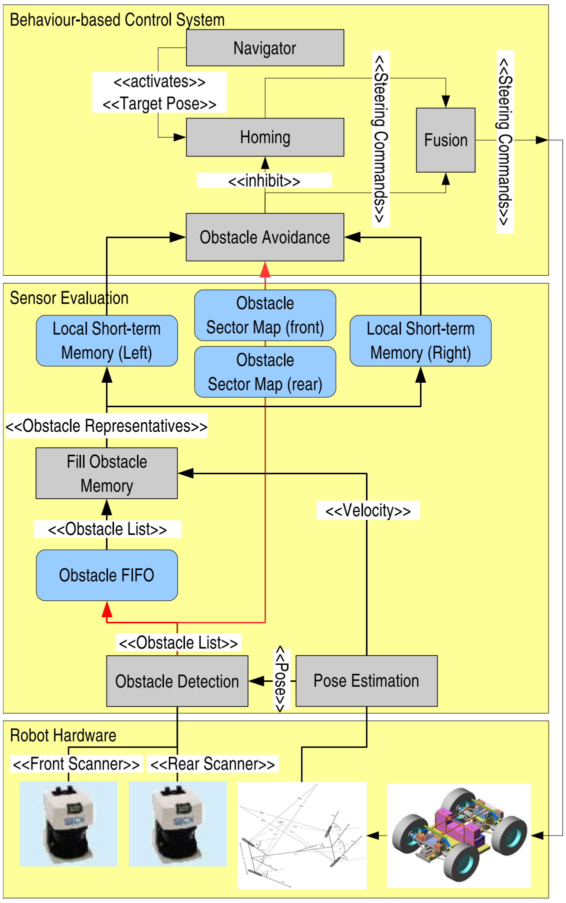


Figure 3: Top-level concept of RAVON's Obstacle Avoidance Facility.

4 A SHORT-TIME MEMORY FOR RAVON

In order to compensate for the blind angles of the sensor systems local short-term memories shall successively be introduced. Any of these can be regarded as a virtual sensor system which can seamlessly be integrated into RAVON's control system as described in (Schäfer et al., 2005). That way step by step blind regions can be covered with increasing accuracy as overlapping virtual sensors yield more and more additional information. In this paper the validation of the general approach and connected technical facilities are presented by the example of closing the gap between the two laser range finders (See Figure 2).

Listing 1: Declarations.

```
# Raw sensor information from the
# laser range finders
scanner_data

# FIFO containing the candidate obstacles
obstacle_fifo

# The maximal length of the queue
# assures that the obstacle lists
# have a certain age when retrieved.
# Furthermore the scans are discarded
# if the vehicle is not moving to prevent
# the accumulation of outdated information.
max_number_of_scans

# Stores the current robot pose in the
# robot's absolute working coord. system
current_robot_pose

# Stores the current velocity of the robot
current_velocity

# Stores representatives in the blind
# angle to the left of the robot.
short_term_memory_left

# Stores representatives in the blind
# angle to the right of the robot.
short_term_memory_right

# Threshold of the maximal euclidean distance
# between the current vehicle pose and the
# one at which a scan was taken.
progress_threshold
```

Figure 3 illustrates the structure of the laser-scanner-based part of the obstacle avoidance system. The sensor flow[1] of the main loop which is dealing with the short-term memory is outlined in Listing 2 (For explanation of the particular variables used see Listing 1). The first step of the main loop is the detection of obstacles from the laser range data which is carried out in the Obstacle Detection facility. The current vehicle pose in the robot's absolute working coordinate system is applied to the resulting obstacle lists in order to yield an absolute reference map through which the robot can virtually be navigated. After that the lists are casted into sector maps which are used by the `Obstacle Avoidance` behaviours to compute evasive steering commands if necessary.

For the extension described in this paper a FIFO queue was introduced which holds in stock the obstacle lists of several subsequent scans. In every sense cycle older obstacle lists are examined for ob-

[1] The control concept underlying all robot projects at the Robotics Research Lab strictly separates the main loop into a buttom up sense and a top down control cycle which are alternately invoked.

stacle representatives which may have migrated into the blind angles. Older in this sense can be interpreted in two ways: a) the scan was taken a certain time ago (*age criterion*[2]) or b) the scan was taken a certain distance ago (*progress criterion*[3]).

Listing 2: Main Loop.

```
Sense ()
  # Detect obstacles from range information
  DetectObstacles (scanner_data ,
                   obstacle_list_fifo)

  if (current_velocity != 0) do
    # remove representatives which are no
    # longer in the scanner's blind angle
    Trim (short_term_memory_left , current_robot_pose)
    Trim (short_term_memory_right , current_robot_pose)

    # check age criterion (a)
    while (Size (obstacle_list_fifo) >
           max_number_of_scans) do
      obstacle_list = First (obstacle_list_fifo)
      FillObstacleMemory (short_term_memory_left ,
                          short_term_memory_right ,
                          obstacle_list ,
                          current_robot_pose)
      RemoveFirst (obstacle_list_fifo)
    od

    # check progress criterion (b)
    obstacle_list = First (obstacle_list_fifo)
    while (Distance (Pose (obstacle_list),
                     current_robot_pose)
           > progress_threshold) do
      FillObstacleMemory (short_term_memory_left ,
                          short_term_memory_right ,
                          obstacle_list ,
                          current_robot_pose)
      RemoveFirst (obstacle_list_fifo)
      obstacle_list = First (obstacle_list_fifo)
    od
  else
    # discard older scans if not moving
    while (Size (obstacle_list_fifo) >
           max_number_of_scans) do
      RemoveFirst (obstacle_list_fifo)
    od
  esle
esneS
```

If one of the two criteria applies to an obstacle list `FillObstacleMemory` checks the obstacle representatives and places them into the corresponding short-term memory. In order to prevent the short-term memory from overfilling `Trim` (See Listing 4) removes obstacle representatives which are no longer in the robot's blind angle. Furthermore the obstacle list FIFO has a limited maximal size. That way only a limited number of scans is keept in mind and is examined which minimises memory consumption and computational expenses.

As the `Obstacle Sector Maps` the short-term memories are evaluated by the `Obstacle Avoidance` component which computes evasive steering commands and inhibits the `Homing` behaviours if applicable. The steering commands from the `Homing` facility and the `Obstacle Avoidance` are fused according to the behaviour architecture described in (Proetzsch et al., 2005). If no obstacles are present `Homing` successively navigates along a trajectory given in subsequent target poses provided by the `Navigator` component.

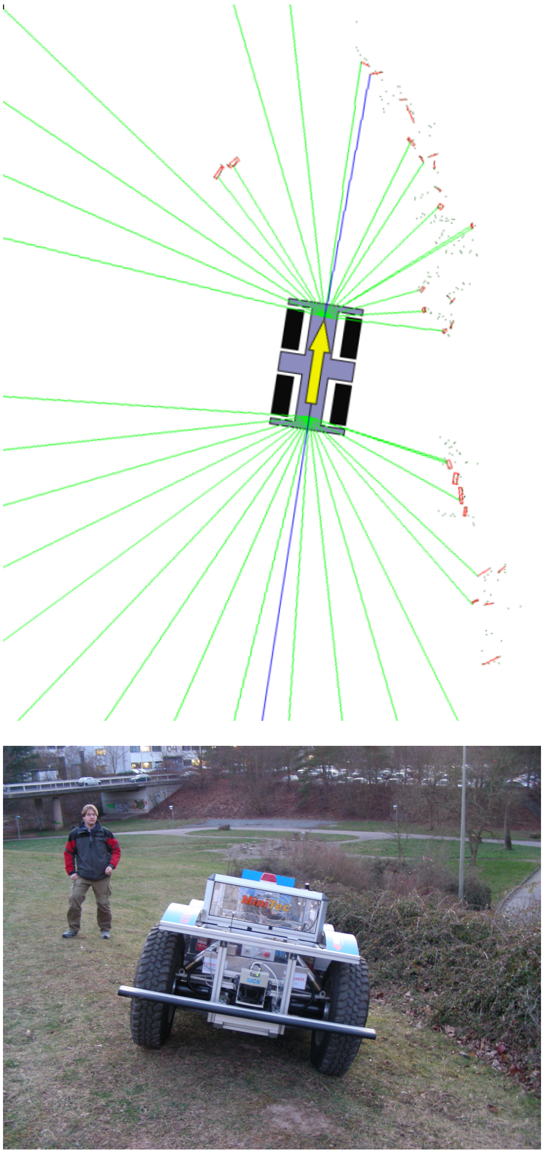


Figure 4: Screenshot showing clustering of detected range values (top) and photo showing the scenario (bottom).

4.1 Obstacle Detection

The range data from the laser scanners is first of all clustered according to point-wise adjacency in order to filter artefacts and entities which are likely to represent flexible vegetation (See Figure 4). In this figure

[2] The age of a scan is modelled implicitly by the maximal number of scans in the queue.

[3] The *progress_threshold* assures that no large gaps occur in the obstacle memory at higher speeds as scans are examined earlier.

the current laser scanner data (dark points) are interpreted as clusters (red boxes) according to their distance. Each of the clusters is then evaluated concerning its dimension and its number of cluster representatives. If both criteria are true clusters are interpreted as obstacles.

In order to have a uniform interpretation of near obstacles which can be used by obstacle avoidance behaviours the sensor coverage area is divided into sectors. In Figure 4 the closest obstacle in each of these sectors is indicated by a green line. In Figure 4 the human to the left and the bushes to the right of the robot have clearly been detected as obstacles. Note that the bushes are subject to severe noise and that they are therefore detected as a large number of small fluctuating objects. For that reason obstacles cannot be correlated over time which is a problem for local map building as representatives accumulate over time if the robot is moving slowly. Using an occupancy grid would be a solution to this problem but it lacks accuracy because of rasterisation. Furthermore the consideration of objects moving independently from the vehicle – which shall be introduced in the future – would be complicated a lot.

4.2 Short-term Memory

As already indicated above obstacles detected in laser rage data consist of points which represent the boundary of a rigid object. Thus in every sensor cycle a list of such point clusters[4] is generated and stored in a FIFO queue for further processing. To every obstacle list the obstacle detection facility attaches the current robot pose in an absolute working coordinate system.

In later sense cycles older obstacle lists are retrieved from the FIFO queue in order to be examined for obstacle representatives which might now be in the blind angle of the laser scanners. Note that in this context the algorithm drops the abstraction of obstacles in favour of simplicity. This does not result in a loss of information as all the robot can actually do if it detects a remembered obstacle at its flank is try to keep away from it by adding a translational component to the steering command. Which representative belongs to what obstacle is in that sense irrelevant.

To fill the obstacle memory routine `FillObstacleMemory` (See Listing 3) examines all obstacles in a provided obstacle list. In order to determine the relation between the vehicle at its current pose and the scan points registered earlier, all representatives are first of all transferred into the absolute working coordinate system of the robot using the pose previously attached to the obstacle list (Subroutine `ToAbsoluteContext`). Intuitively explained this is as if you would virtually drive a robot model through an absolute landscape while checking for points in the blind regions of the robot. That way it is not necessary to transform the representatives whenever the robot changes its pose.

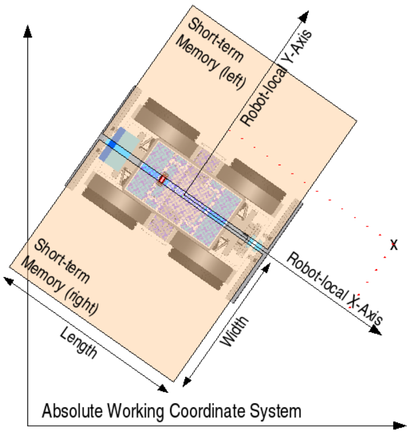


Figure 5: Schema of how to determine whether an obstacle representative resides in the blind region of the vehicle.

Listing 3: Fill the Obstacle Memory.

```
FillObstacleMemory (short_term_memory_left ,
                    short_term_memory_right ,
                    obstacle_list ,
                    current_robot_pose)
  forall (obstacle in obstacle_list) do
    forall (representative in obstacle) do
      absolute_representative =
        ToAbsoluteContext (Pose (obstacle_list),
                           representative)

      y = PointOnLine (XAxis (current_robot_pose),
                       absolute_representative)
      x = PointOnLine (YAxis (current_robot_pose),
                       absolute_representative)

      if (Abs (y) < Width (short_term_memory)/2 AND
          Abs (x) < Length (short_term_memory)/2) {
        if (ToRightOfRobot (current_robot_pose ,
                            absolute_representative))
          Append (short_term_memory_right ,
                  absolute_representative)
        else
          Append (short_term_memory_left ,
                  absolute_representative)
        fi
      od
    od
  od
yromeMelcatsbOlliF
```

Once in an absolute context the coordinates can reside where they are. With the subroutine

[4] In the following this list shall be referred to as *obstacle list*.

`PointOnLine` the distance of the representative to the X-axis and the Y-axis of the robot-local coordinate system is computed (See Figure 5). This yields the robot-local coordinate of the representative which offers a straightforward check whether the representative resides in a blind area of the robot. Depending whether the representative is rather to the left or the right of the robot it is added to the corresponding short-term memory. Routine `Trim` (See Listing 4) works in analogy just the other way around to sweep representatives from the short-term memories which are no longer in the blind angle of the scanners.

Listing 4: Trim the Short-term Memory.

```
Trim (short_term_memory, current_robot_pose)
  forall (representative in short_term_memory) do

    y = PointOnLine (XAxis (current_robot_pose),
                            representative)
    x = PointOnLine (YAxis (current_robot_pose),
                            representative)

    # Check whether the representative is
    # still in the blind angle
    if (Abs (y) > Width (short_term_memory)/2 OR
        Abs (x) > Length (short_term_memory)/2)
        Remove (representative, short_term_memory)
    fi
  od
mirT
```

Note that the robot pose used during the procedures outlined above currently relies on odometry only. This is due to the fact that odometry is locally coherent in comparison to data fused from odometry, GPS, the inertial measurement unit and the magnetic field sensor. Eventhough globally more accurate, the latter three introduce certain local jumps into the pose, which are a problem for precise local navigation. The pose drift immanent in odometry on the other hand is not an issue for the approach presented in this paper as it is enough to be rather precise over the distance of a vehicle length. When the robot has passed the obstacle it is erased from the short-term memory anyway. In the future visual odometry shall additionally be called into service in order to gain more accurate pose information. In particular on slippery surfaces the authors expect this combination to be a lot more robust than banking on odometry only.

5 EVALUATION IN EXPERIMENTS

The short-term memory described before was implemented and integrated into the control system of RAVON. In order to evaluate the approach several test runs have been executed. First of all autonomous navigation along predefined tracks has been executed. In this case the vehicle follows the given direction to the target point as long as there is no hindrance. In case of obstacles, however, the straight motion is interrupted by avoidance behaviours. This effect remains active even if the obstacle is in the blind side region. Therefore the robot gets back on track only as soon as all hindrances are left behind.

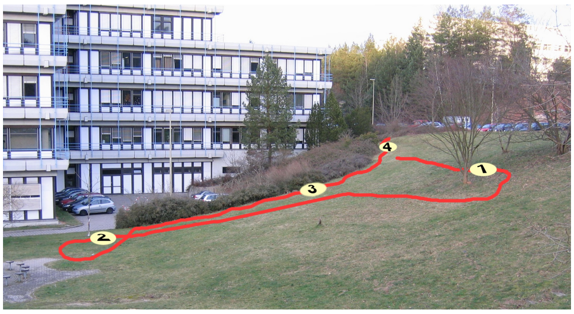


Figure 6: Image of the test scenario with overlaid trace.

For clarification of the properties of the short-term memory in this section a test run is presented where the robot is remote controlled. In this experiment the robot is driven through a locally flat environment. All the time the behaviour-based control system evaluates sensor information and overrides the given commands if necessary in order to avoid collisions.

The scenario for this run is hilly grass land as demonstrated in Figure 6. The image shows some trees to be avoided as well as some bushes with unknown load bearing surface. For clarification the approximate path of the vehicle is overlaid. Four markers indicate positions referred to in the following description.

Using the localisation system the trace of the vehicle can be plotted as presented in Figure 7. Some of the trees and bushes are integrated in order to point out the sensor data and vehicle reaction. Figure 8 shows the scanner data as well as clusters and sector values (see Section 4.1) for the four positions indicated by markers in the trace plot. Additionally data of the obstacle memory at the side of the vehicle is depicted as blue lines.

Position 1 shows a situation where the robot was steered into the direction of a tree. After an evading manoeuvre the robot is now located next to the tree. The sensor data shows that the obstacle is at the edge of the side region and would eventually be hidden. However, the blue lines indicate data stored in the obstacle memory. Therefore the robot continues preserving a minimal distance to the obstacle. At

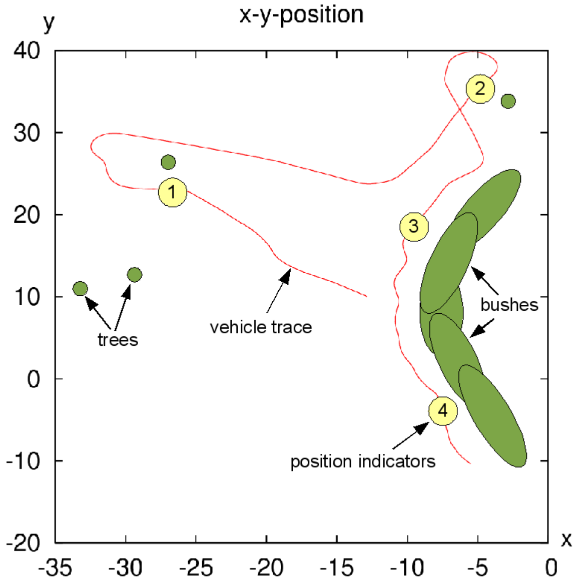


Figure 7: Top view position diagram of the experiment.

position 2 the robot is in a similar situation. A tree which was detected by the laser range finder at the front changed its relative position such that it is located at the right side of the robot.

Positions 3 and 4 show the situation where the robot is manually steered to the left. Due to the obstacles detected and stored in the short-term memory the evading behaviours force the robot away from the hindrances. This interaction results in a smooth following of the bushes.

The experiments showed a major improvement of the reaction to obstacles. Behaviours regarding the short-term memory prevent the robot of driving into the direction of hidden obstacles. For static obstacles this leads to a system reacting on hindrances which can be located anywhere around the vehicle.

6 CONCLUSIONS AND FUTURE WORK

In this paper we presented an approach for virtually extending the coverage of obstacle detection sensor systems. The formulated algorithm uses information about the vehicle motion to propagate sensor data into the blind zone of the robot. Due to a massive reduction of stored data the system features low memory usage. Additionally there is no need for calculating the correspondence between consecutive data making the approach applicable for realtime scenarios.

Next steps involve the integration of a stereo cam-

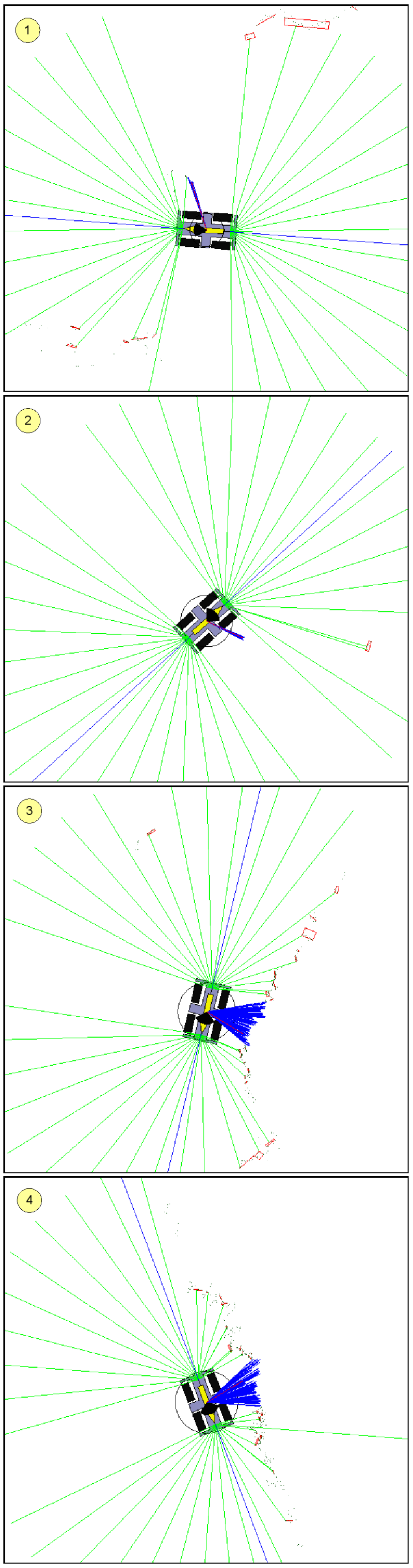


Figure 8: Sensor data visualisation during the experiment.

era system. Due to the device's limited field of view the presented algorithm leads to an eminent extension of the supervised area surrounding the vehicle. Additionally the accuracy of the local positioning system will be enhanced by visual ego motion estimation.

REFERENCES

Badal, S., Ravela, S., Draper, B., and Hanson, A. (1994). A practical obstacle detection and avoidance system. In *IEEE Workshop on Applications of Computer Vision*.

Bonnafous, D., Lacroix, S., and Simon, T. (2001). Motion generation for a rover on rough terrain. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Kamon, I., Rivlin, E., and Rimon, E. (1996). A new range-sensor based globally convergent navigation algorithm for mobile robots. In *IEEE International Conference on Robotics and Automation (ICRA)*.

Laubach, S. L. and Burdick, J. W. (1999). An autonomous sensor-based path-planner for planetary microrovers. In *IEEE Int. Conf. on Robotics and Automation*.

McVea, D. and Pearson, K. (2006). Long-lasting memories of obstacles guide leg movements in the walking cat. *The Journal of Neuroscience*.

Proetzsch, M., Luksch, T., and Berns, K. (2005). Fault-tolerant behavior-based motion control for offroad navigation. In *20th IEEE International Conference on Robotics and Automation (ICRA)*, Barcelona, Spain.

Schäfer, H. and Berns, K. (2006). Ravon - an autonomous vehicle for risky intervention and surveillance. In *International Workshop on Robotics for risky intervention and environmental surveillance - RISE*.

Schäfer, H., Proetzsch, M., and Berns, K. (2005). Extension approach for the behaviour-based control system of the outdoor robot ravon. In *Autonome Mobile Systeme*.

Schmitz, N., Proetzsch, M., and Berns, K. (2006). Pose estimation in rough terrain for the outdoor vehicle ravon. In *37th International Symposium on Robotics (ISR)*.

Shiller, Z. (2000). Obstacle traversal for space exploration. In *IEEE International Conference on Robotics and Automation*.

METHODOLOGY FOR LEARNING VISUAL REACTIVE BEHAVIOURS IN ROBOTICS THROUGH REINFORCEMENT AND IMAGE-BASED STATES

Pablo Quintía, José E. Domenech, Cristina Gamallo and Carlos V. Regueiro
Facultad de Informática, Universidad de A Coruña, 15071 A Coruña, Spain
pablo.quintiavidal@gmail.com, pintzio@gmail.com, cristinagamallo@gmail.com, cvazquez@udc.es

Keywords: Reinforcement Learning, Mobile Robotics, Artificial Vision, Visual Reactive Behaviours, Motivation.

Abstract: This article describes the development of a methodology for the learning of visual and reactive behaviours using reinforcement learning. With the use of artificial vision the environment is perceived in 3D, and it is possible to avoid obstacles that are invisible to other sensors that are more common in mobile robotics. Reinforcement learning reduces the need for intervention in behaviour design, and simplifies its adjustment to the environment, the robot and the task. The designed methodology is intended to be general; thus, in order to change the desired behaviour, only the reinforcement and the filtering of the image need to be changed. For the definition of the reinforcement a laser sensor is used, and for the definition of the states a fixed 3x3 grid is used. The behaviours learned were wall following, object following, corridor following and platform following. Results are presented with a Pioneer 2 AT. A Gazebo 3D simulator was used for the Learning and testing phase, and a test of the wall following behaviour was carried out in a real environment.

1 INTRODUCTION

The design and implementation of reactive behaviours for the control of autonomous mobile robots has been shown to be one of the most efficient ways of carrying out low-level tasks. These require a very short response time and continuous interaction with the environment, which is almost totally unknown, complex and dynamic. Thus arises the challenge of specifying how each one should be implemented.

One of the most promising techniques is reinforcement learning (RL) (Millán et al., 2002; Wyatt, 1995), one of the principal advantages of which is that it minimizes interaction with the designer, since only the set of states and actions and the reinforcement has to be established. There is no need to identify all the situations in which the robot may find itself, nor the action to be implemented in each of them; only whether the result of the action is acceptable or not need be stated.

This work describes the design and implementation of four visual behaviours using a single camera, with the aim of studying the feasibility of the project and enabling a simple, economical implementation. The results obtained are generalisable, with the possible exception of the discrimination between floor and obstacles. This perception would be more robust and efficient if in-depth information were used.

We now comment on related work and go on to describe the TTD(λ) - Q-learning algorithm and its application to the four behaviours chosen. We then show the results obtained in simulation. For this phase the multirobot simulator Player/Stage/Gazebo was chosen as it is highly generalized, it supports the Pioneer 2-AT mobile robot, and simulates in 3-D (Gazebo). Lastly, we finish off with a conclusions and future work section.

2 RELATED WORK

Only a small number of studies have used vision as the principal sensorial input for visual behaviours in a mobile robot. This is probably due to the high cost associated with processing visual information (Nakamura and Asada, 1995).

In some works, reinforcement is used to learn visual behaviours that are similar to wall following (e.g.

servoing and wandering), but which are simpler as there is no need to maintain a distance from the wall the robot is following. Gaskett et al. (Gaskett et al., 2000) use an improved version of Q-learning ("Advantage Learning") which handles continuous states and actions thanks to neural networks.

Another implementation of visual servoing behaviour can be found in (Martínez-Marín and Duckett, 2005). The algorithm used is another variant of Q-learning that permits real-time learning. Unlike in the present work, the state of the agent is defined by the position of the camera (inclination and angle of turn) focused on the objective, and not by the image. Thus, active vision is required, along with a perfect identification of the objective, which is not always possible. It is also difficult to use the same system to implement more than one behaviour simultaneously.

A similar, but more complete, general approach was taken in (Boada et al., 2002). This system learns basic behaviours (watching and orientation) for controlling a pan-tilt unit and the robot, and combines them to obtain complex ones (approach). Nevertheless, they need to detect and identify the objective, which is very difficult with a wall, a corridor or a platform. Once again, states were not defined directly by the image.

Ruiz et al. (Ruiz et al., 2005) have implemented two visual behaviours. Their approach is based on the detection of straight segments in the image, which to a certain degree limits its mode of use. Moreover, control is totally heuristic, with no type of learning. Our approach is more general and is not conditioned by the characteristics of the environment.

3 REINFORCEMENT LEARNING

Reinforcement learning (RL) is based on the use of a qualification (reinforcement) of the agent's actions by the environment. The reinforcement does not indicate the correct action (supervised learning), only whether it has been satisfactory or not, and is not usually immediate in time. The situations of the agent are usually codified into discrete states (s) in which various actions (a) can be implemented (may be different for each state).

A simple and intuitive definition of reinforcement has been sought, as we believe that it is one of the main advantages of this type of algorithm. Reinforcement indicates *only* those situations in which it is highly probable that the robot has ceased to implement the task correctly (i.e. the robot is excessively far from a wall), or has definitively carried it out badly (the robot collides with an element in its environment or is excessively close to one).

Learning consists of approximating a quality function $Q(s,a)$. The optimal action in each state is the one that maximizes its evaluations. The algorithm chosen for this work is *TTD(λ) - Q-learning*, due to it being based on the basic and simply *Q-Learning* algorithm but applied to several states obtaining a faster convergence time. The updating equation for the valuations is:

$$\Delta Q(s_t,a_t) = \alpha[r_t + \gamma max_a Q(s_{t+1},a) - Q(s_t,a_t)]e_t(s) \ , \tag{1}$$

for all $s_t \in S_n$ where α is the learning coefficient, γ is the reinforcement discount coefficient, S_n is the set of the n-last visited states and e_t is the eligibility trace of the state:

$$e_t(s) \quad = \quad \begin{cases} \gamma\lambda e_{t-1}(s) & \text{if } s \neq s_t, \\ 1 & \text{if } s = s_t, \end{cases} \tag{2}$$

where λ is a decay parameter. Initial Q-values $\in [-0,9,-1,0]$.

One drawback of *Q-learning* is the need to strike a balance between exploration and exploitation. In order to do so, the *Softmax* method (Moreno et al., 2004) has been applied, where the probability of taking the action a_i in the state s at time t is:

$$Pr(s,a_i) = \frac{e^{Q_t(s,a_i)/T_s}}{\sum_{j=1}^{n} e^{Q_t(s,a_j)/T_s}} \ , \tag{3}$$

where $\{a_1,....,a_n\}$ is the set of possible actions in the state s and T_s is the temperature associated to state s.

With temperature it is possible to regulate the probability distribution between actions, and thus, the balance between exploration and exploitation. Initially we start from a high temperature (greater exploration of actions) and this is progressively reduced throughout the learning in order to principally select those actions with the best evaluation.

Temperature is modified for each state in accordance with the equation:

$$T(t) \quad = \quad \begin{cases} T_0 e^{-\frac{t}{t_k} ln \frac{T_0}{k}} & \text{if } t \leq t_k, \\ k & \text{if } t > t_k, \end{cases} \tag{4}$$

where t is the number of times the current state has appeared, T_0 is the "initial" temperature, k is the minimum temperature (the state does not explore more actions) and t_k is the number of appearances that are required for the temperature to reach k.

Table 1 shows a summary of the values used for the parameters of the TTD(λ) - Q-learning algorithm.

4 BEHAVIOURS

4.1 Common Aspects

On a reactive behaviour and with RL, each state must contain all the information for selecting the next action. After several tests the camera, which uses a 80 wide-angle lens, (Fig. 1) was placed 53 cm over the robot and inclined 42 degrees towards the floor. Using these values there is no need to change the position of the camera for the different behaviours.

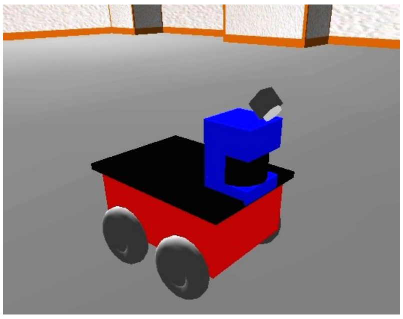


Figure 1: Position of the camera and the laser in the Gazebo simulator.

The position of the laser depends on the reinforcement wanted but, with the exception of the platform following, the laser was placed over the front of the robot, centred and parallel to the floor. For the platform following behaviour it was inclined 45 degrees towards the floor in order to detect the limits of the platform. A SICK LMS 200 laser sensor was used.

4.1.1 State Space

As was to be expected, the definition of the state space was critical in the development of this work, since it has to respond to criteria that are at times conflicting. On one hand, the space must be small, as convergence time in RL increases exponentially with the number of states. On the other hand, "perceptual aliasing" must be avoided; that is, the system should not classify two situations in which the robot must execute very different commands in the same state.

Table 1: Values used in this work for the TTD(λ) - Q-learning algorithm.

Par.	Description	Value
α	Learning coefficient	0.2
γ	Reinforcement discount coefficient	0.99
T_0	Initial temperature	0.07
k	Lower temperature limit	0.009
t_k	State exploration limit	500
λ	Decay parameter	0.9
n	Number of last visited states	3

Lastly, due to the quantity of information generated by a camera, the image needs to be processed in order to reduce the amount of pertinent information (Nehmzow, 1999). In order to resolve these problems, a simple, computationally efficient methodology has been employed, which can be run in real time on a mobile robot (Regueiro et al., 2006). This approach is easily generalisable.

Firstly, with the exception of object following, the image is processed with a Sobel edge enhancement filter (Fig. 2(b)) to highlight the pertinent information: obstacles (positive and negative) on the floor. This floor detection process is highly sensitive to changes in lighting and textures. Nevertheless, it can be improved in different ways: with stereo vision, by calibrating the camera to detect the ground plane or by applying Machine-Learning techniques for ground detection (Michels et al., 2005). For the object following behaviour the image is processed to detect a specific colour (red was chosen). Then for each pixel its red value is analysed, and if this exceeds a specified threshold it is set to white, otherwise it is set to black.

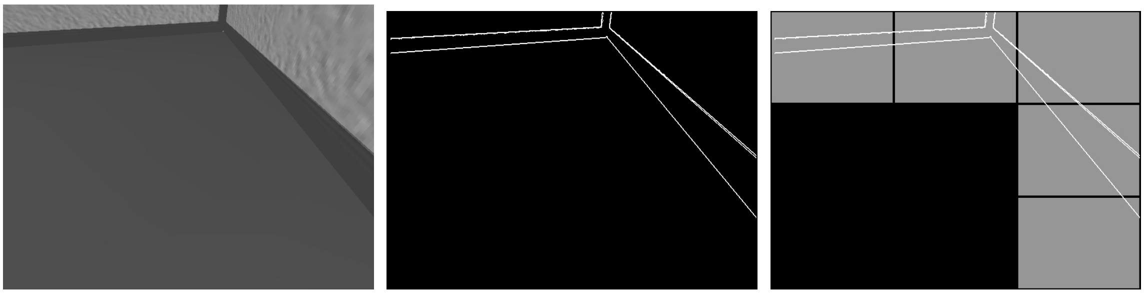
Secondly, the image is divided into a grid made up of 3 rows and 3 columns (Fig. 2(c)) for codification. A cell is considered occupied if the percentage of edge pixels reaches a given threshold. This step is essential for avoiding "perceptual aliasing". Thus defined, the state space is 2^9, and in order to reduce it, it is supposed that if a cell in one of the columns is occupied, all those cells above it are also occupied (Fig. 2(c)). Hence the number of possible states is $(3+1)^3$; i.e. 64. The state space may be further reduced, but drastic modifications would be needed in the codification, which would be difficult to justify.

4.1.2 Action Space

The action space was chosen in order to allow a flexible behaviour, and at the same time avoid a series of actions that would leave the robot immobile and without negative reinforcement. One constraint of Q-learning is that actions must be discrete. Hence, the action space chosen is as in Table 2

4.2 Wall Following Behaviour

It has been shown that wall following behaviour is one of the most useful when robots need to move reactively and safely through their environment (Regueiro et al., 2002). One of its advantages is that it only uses local information, and it makes use of the topological structure of the environment.



(a) (b) (c)

Figure 2: Determination of state with an image: a) original image; b) edge pixels (Sobel filter); and c) final codified state (showing the codification grid and the free and occupied cells).

4.2.1 Definition of Reinforcement

The defined reward is always negative (-1), spurious in time, and has two different components:

1. Close reinforcement: if any of the laser beams detects an obstacle at 25 cm or less.
2. Far reinforcement: if the right-hand beam detects a distance greater of 1 m.

The reinforcement has been defined in two ways:

1. Instant reinforcement: the reinforcement is applied when it finds a non desired situation. Used with the close reinforcement.
2. Delayed reinforcement: the reinforcement is applied when a undesired situation persists over a period of time. Used with the "far reinforcement".

With delayed reinforcement the need to always have a wall on the right side is relaxed and thus a more robust behaviour is possible.

Table 2: Action space chosen.

Act. n.	Linear speed m/s	Angular speed rad/s
0	0.4	0.2
1	0.4	0
2	0.4	-0.2
3	0.2	0.8
4	0.2	0.6
5	0.2	0.4
6	0.2	0.2
7	0.2	0
8	0.2	-0.2
9	0.2	-0.4
10	0.2	-0.6
11	0.2	-0.8
12	0.05	0.8
13	0.05	-0.8

4.2.2 Results

The environment used is shown in Fig. 4(a).

Wall following behaviour belongs to the class of continuous tasks which persist over time. This means that they are not divided naturally into episodes, as would be desirable for application of a reinforcement algorithm. In order to avoid this drawback a reinforcement is generated after each serious system error (collision or excessive distancing) and the robot executes a recovery function which returns the robot to a safe state.

Fig. 3 shows the reinforcement received during the learning phase in TTD(λ). Each point represents the reinforcement accumulated in the previous 400 learning cycles. Thus, the first point indicates that the reinforcement received since the onset of the experiment up until 400 cycles has been -12, and so forth. The learned Q-values are stored for their subsequent testing.

As can be seen in the diagrams, the agent learns the task in 2,000 learning cycles with TTD(λ) as opposed to 80,000 cycles obtained with Q-Learning algorithm and 8x4 grid (Regueiro et al., 2006).

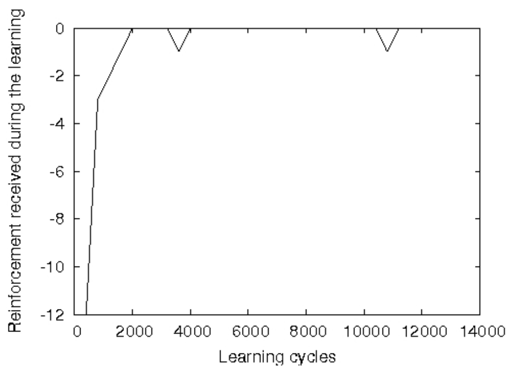


Figure 3: Reinforcement accumulated during the learning of wall following behaviour.

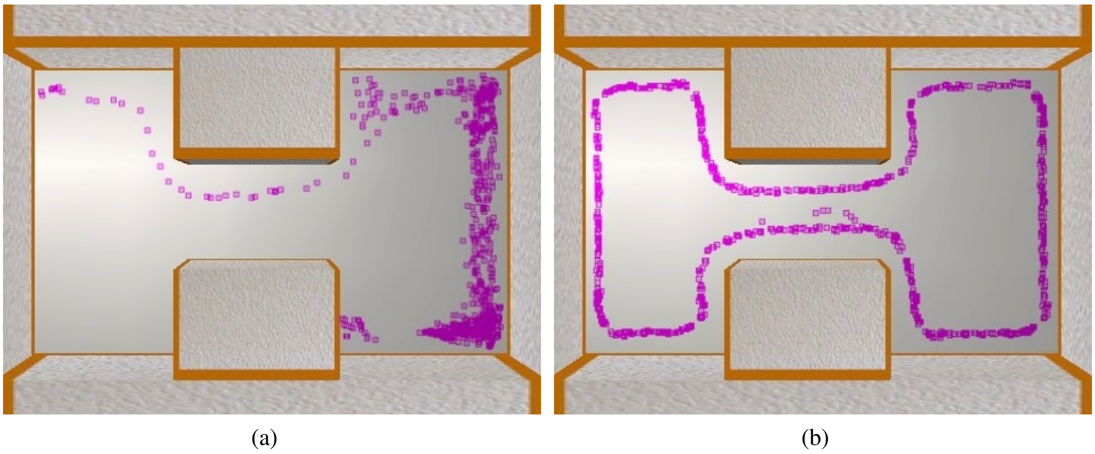


Figure 4: Wall following behaviour: a) Trajectories obtained during initial stages; and b) Trajectories obtained during final stages.

4.3 Corridor Following Behaviour

The corridor following behaviour is one of the simplest behaviours to learn; nevertheless, when working with RL it has problems with decision selection at crossroads. It simply makes no decision, as could be expected from a reactive behaviour, thus the robot cannot be ordered to go straight on or to turn at a junction. Several positive reinforcements were used to make the robot go straight on, but without success.

4.3.1 Definition of Reinforcement

Two classes of reinforcements were defined:

1. Critical reinforcement: if any of the laser beams detects an obstacle at 25 cm or less the reward is -1.
2. Speed reinforcement: used to favour the use of straight and fast actions or/and penalise slow and turning actions with values in $[-0.4, 0.3]$.

There were no significant variations in the final behaviours with the use of speed reinforcement. We have to mention that even when defining an erroneous reinforcement the robot learned the correct behaviour.

4.3.2 Results

The environment used is shown in Fig. 6(a).

As has previously been mentioned, the final behaviours show a tendency to turn to right or left. The example in Fig. 6(b) shows a tendency to turn left; if not to straight on and then turn right. That is the reason for the difference between turns to the left, where the robot turns smoothly and passes close to the left wall, and the right turns, where the robot tries to go straight until the wall is directly in front of it.

Fig. 5 shows the reinforcement received during the learning phase in TTD(λ). As can be seen in the diagrams, the agent learns the task in 800 learning cycles.

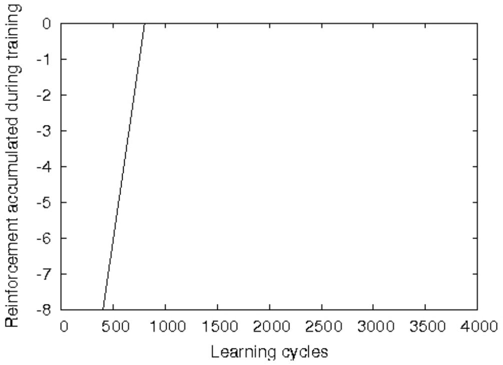


Figure 5: Reinforcement accumulated during the learning of corridor following.

4.4 Platform Following Behaviour

The platform following behaviour is very similar to the corridor following behaviour, but the laser sensor points 45 degrees towards the ground, as it has to detect uneven features in the floor. This behaviour also has the same problem at crossroads.

4.4.1 Definition of Reinforcement

A simple calibration of the laser is needed at the beginning of the execution. The laser measures the distance of the front beam (number 90) and this distance is used to calculate the expected distance for the rest of the beams (values between 50 and 130). If a laser

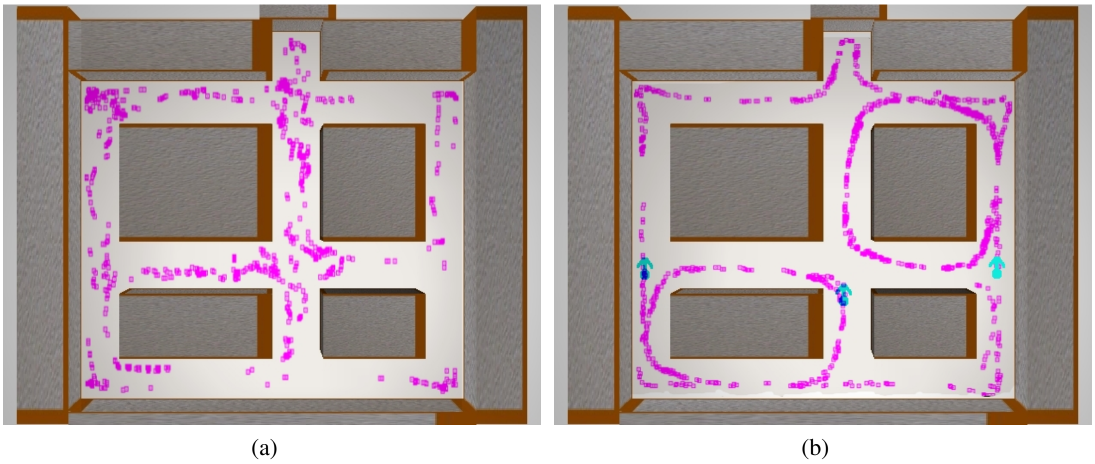


Figure 6: Corridor following behaviour: a) Trajectories obtained during initial stages; and b) Trajectories obtained during final stages.

beam value is greater than the expected distance plus 15 cm, then the reward is -1.

4.4.2 Results

The environment used is shown in Fig. 8(a).

Fig. 7 shows the reinforcement received during the learning phase in TTD(λ). As can be seen in the diagrams, the agent learns the task in 2,000 learning cycles.

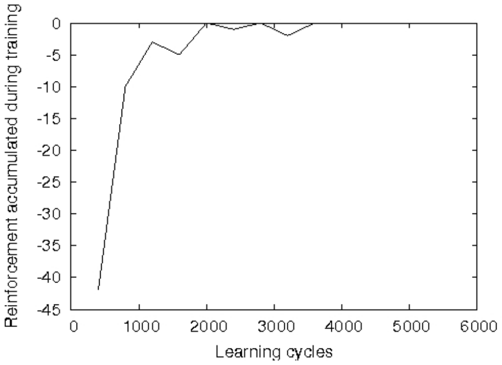


Figure 7: Reinforcement accumulated during the learning of platform following behaviour.

4.5 Object Following Behaviour

The main change for this behaviour was the filtering of the image. Here, instead of using a Sobel filter the image is preprocessed with a simple colour filter. Red was chosen and a red box is attached to the prey, another Pioneer 2-AT executing a follow line behaviour. The prey has a follow line behaviour, the path of the pray is shown in Fig. 10(a).

4.5.1 Definition of Reinforcement

The defined reward is negative (-1.0) when any of the following conditions is accomplished:

1. Close reinforcement: if any of the laser beams detects an obstacle at 30 cm or less.
2. Image reinforcement: if the prey is not viewed by the camera.

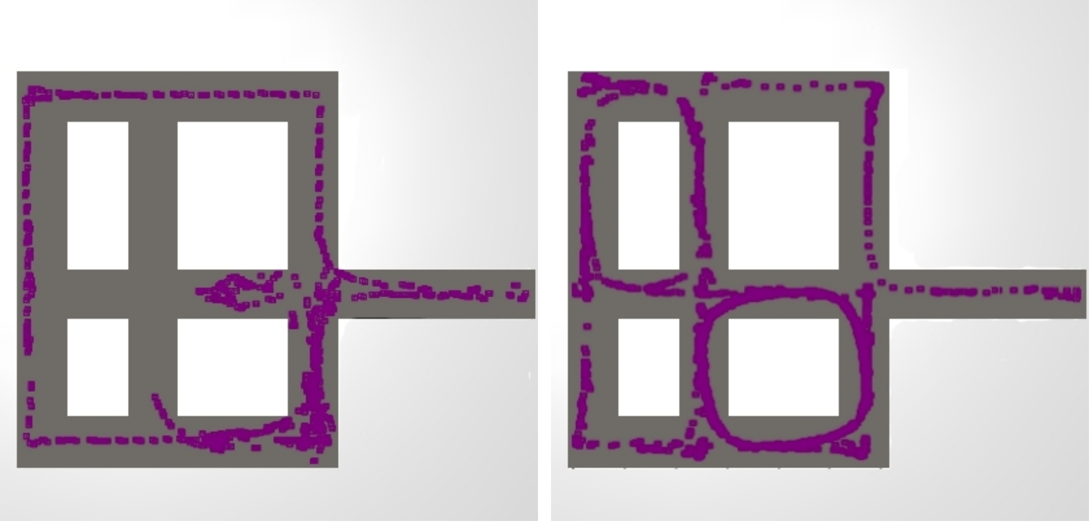
4.5.2 Results

Fig. 9 shows the reinforcement received during the learning phase in TTD(λ). As can be seen in the diagrams, the agent learns the task in 10,500 learning cycles. The learning is slower for this behaviour due to the fact that the path of the prey is very long, an entire lap taking approximately 7 minutes, so a number of difficult situations appeared very sporadically.

As can be seen in Fig. 10(b) the path of the final behaviour is smoother than the path of the prey. The average distance between the robot and a prey moving at constant speed of 0.2 m/s is 0.9 m. We have tested several speeds for the prey: if the speed is slower than 0.35 m/s the robot can follow it (it should be remembered that the maximum speed of the robot is 0.4 m/s, see table 2).

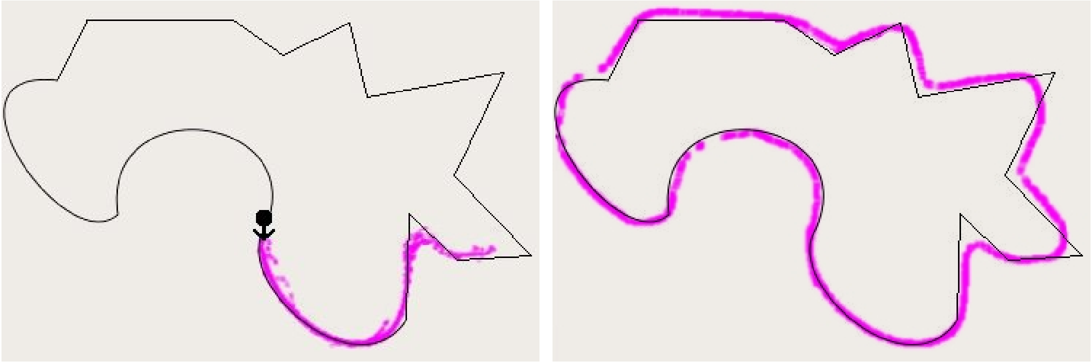
4.6 Test in a Real Robot

Finally we have tested the behaviour with minor changes on a real environment: a long corridor, narrower than learning environment, with many new features, such as trash baskets, radiators, plants, glass walls, open and closed doors. This real environment



(a) (b)

Figure 8: Platform following behaviour: a) Trajectories obtained during initial stages; and b) Trajectories obtained during final stages.



(a) (b)

Figure 10: Object following behaviour: a) Trajectories obtained during initial stages; and b) Trajectories obtained during final stages.

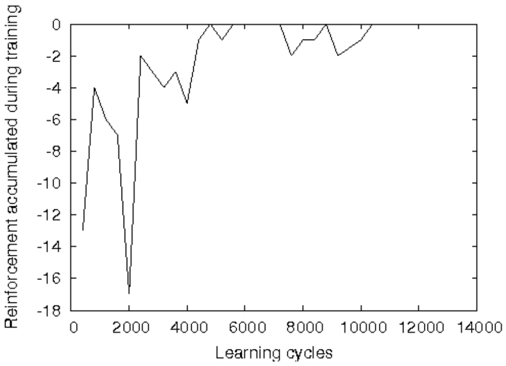


Figure 9: Reinforcement accumulated during the learning of object following behaviour.

also has both natural and artificial lighting. As can be seen by examining the robot's path on Fig. 11, our behaviour works perfectly.

5 CONCLUSIONS AND FUTURE WORK

In this work four visual reactive behaviours for the Pioneer 2 AT robot have been implemented with RL algorithms. All behaviours use the same state and action spaces, only reinforcement changes. A rangefinder laser is used to define the reinforcement of each behaviour. Learning and testing were carried

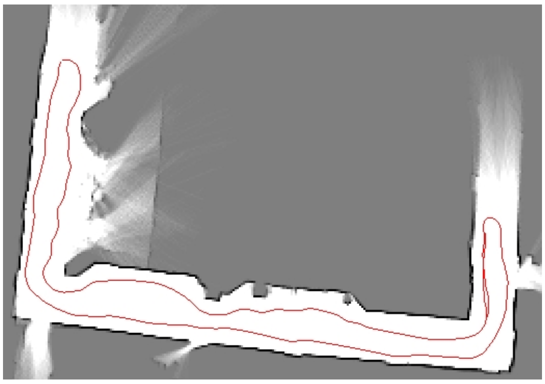


Figure 11: Robot's path carrying out the visual wall following behaviour on a real environment (right wall). The map is a certainty grid created with a LMS200 laser (white pixels are free space, black obstacles and grey uncertainty).

out on the 3-D Gazebo simulator. A wall following test in a real environment is also shown.

Using a fisheye camera the environment is perceived in 3D, and it is possible to avoid obstacles that are invisible to other sensors which are more common in mobile robotics (laser or ultrasounds).

In the proposed methodology the states are defined directly from the image after a simple preprocessing (Sobel or colour filtering) with no calibration process. With a 3x3 grid, we can define a state space of only 64 states. It has a certain degree of "perceptual aliasing", but RL algorithm converges. We have also tested grids of different dimensions with similar results but greater convergence time. A delicate balance need be struck between reducing the number of states and avoiding "perceptual aliasing".

The proposed codification and methodology is general, not specific for the task, and has proved to be efficient and valid, and easy to adapt to distint behaviours. The system works with different types of reinforcement and filtering.

Various tests were carried out to verify the robustness of the learned behaviours. We used obstacles that were not detected by the laser device, and walls with gaps. In both cases the system generalized perfectly and the results were optimal. If the gaps in the walls were large (over 40 cm) a large number of new states appeared with respect to the training process, and the final result deteriorated.

Future lines of work include on-line real robot learning, the integration of several behaviours (e.g. follow objects and avoid obstacles) and establishing a mechanism for automatically defining the states of RL (e.g. neural networks).

ACKNOWLEDGEMENTS

This paper was supported in part by the Xunta de Galicia and Spanish Government under Grants PGIDIT04-TIC206011PR, TIN2005-03844, and department colaboration grant (B.O.E. 16-06-2006).

REFERENCES

Boada, M., Barber, R., and Salichs, M. (2002). Visual approach skill for a mobile robot using learning and fusion of simple skills. *Robotics and Autonomous Systems*, 38:157–170.

Gaskett, C., Fletcher, L., and Zelinsky, A. (2000). Reinforcement learning for a vision based mobile robot. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, volume 1, pages 403–409.

Martínez-Marín, T. and Duckett, T. (2005). Fast reinforcement learning for vision-guided mobile robots. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 4170–4175.

Michels, J., Saxena, A., and Ng, A. (2005). High speed obstacle avoidance using monocular vision and reinforcement learning. In *Proc. Int. Conf. on Machine Learning*.

Millán, J. R., Posenato, D., and Dedieu, E. (2002). Continuous-action Q-Learning. *Machine Learning*, 49:247, 265.

Moreno, D., Regueiro, C., Iglesias, R., and Barro, S. (2004). Using prior knowledge to improve reinforcement learning in mobile robotics. In *Towards Autonomous Robotic Systems (TAROS)*.

Nakamura, T. and Asada, M. (1995). Motion sketch: Acquisition of visual motion guided behaviors. In *IJCAI*, pages 126–132.

Nehmzow, U. (1999). Vision processing for robot learning. *Industrial Robot*, 26(2):121–130.

Regueiro, C., Domenech, J., Iglesias, R., and Correa, J. (2006). Acquiring contour following behaviour in robotics through q-learning and image-based states. In *Proc. XV Int. Conf. on Computer and Information Sciences Engineering (CISE)*, pages 403–409.

Regueiro, C., Rodríguez, M., Correa, J., Moreno, D., Iglesias, R., and Barro, S. (2002). A control architecture for mobile robotics based on specialists. In Leondes, C., editor, *Intelligent Systems: Technology and Applications*, volume 6, pages 337–360. CRC Press.

Ruiz, J., Montero, P., Martín, F., and Matellán, V. (2005). Vision based behaviors for a legged robot. In *Proc. Workshop en Agentes Físicos (WAF)*, pages 59–66.

Wyatt, J. (1995). Issues in putting reinforcement learning onto robots. In *10th Biennal Conference of the AISB*, Sheffield, UK.

FORMAL VERIFICATION OF SAFETY BEHAVIOURS OF THE OUTDOOR ROBOT RAVON

Martin Proetzsch, Karsten Berns
Robotics Research Lab, University of Kaiserslautern, Germany
proetzsch@informatik.uni-kl.de, berns@informatik.uni-kl.de

T. Schuele, K. Schneider
Reactive Systems Group, University of Kaiserslautern, Germany
schuele@informatik.uni-kl.de, schneider@informatik.uni-kl.de

Keywords: Behaviour-based control, formal verification, outdoor robotics.

Abstract: This paper presents an approach to the formal verification of safety properties of the behaviour-based control network of the mobile outdoor robot RAVON. In particular, we consider behaviours that are used for the computation of the projected vehicle's velocity from obstacle proximity sensor data and inclination information. We describe how this group of behaviours is implemented in the synchronous language Quartz in order to be formally verified using model checking techniques of the Averest verification framework. Moreover, by integrating the automatically generated and verified code into the behaviour network, it can be guaranteed that the robot slows down and stops as required by the given safety specifications.

1 INTRODUCTION

More and more applications like unmanned space travelling, autonomous farming, civil protection, and humanitarian demining require autonomous vehicles navigating in unstructured natural terrain. The diversity as well as missing information for physical models of outdoor scenarios call for a flexible control architecture not requiring a complete knowledge of the environment to achieve robust locomotion. In this context, behaviour-based control networks have proven suitable for appropriate reaction on external influences.

One of the advantages of behaviour-based architectures is emergence: The combination of behaviours leads to proper reactions not directly explainable by the individual components. On the other hand, this feature also poses problems concerning predictability, making it difficult to reason about the correctness of the overall system. However, safety critical applications require proofs that guarantee that given specifications are met. In particular, the correctness of reactive control layers is mandatory for the final behaviour of the robot.

In circuit design, formal verification has already become a standard to avoid design errors. Since all possible input traces of a system are considered by formal verification methods, it is guaranteed that the checked specifications hold under all circumstances. In particular, different kinds of model checking (Schneider, 2003; Schuele and Schneider, 2006) are popular verification methods due to the high degree of automation.

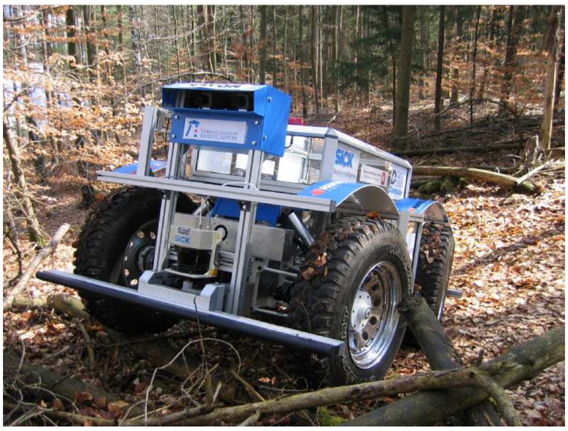


Figure 1: RAVON in rough outdoor terrain.

In this paper, we consider the formal verification of a part of the behaviour-based control network of the mobile outdoor platform RAVON (**R**obust **A**utonomous **V**ehicle for **O**ff-road **N**avigation, see Fig. 1), namely RAVON's control system that is responsible for slowing down and stopping the vehicle. It is clear that this part is highly safety critical, and therefore, it is very important to guarantee its correctness. To this end, we have implemented this part of the behaviour-based control network in the synchronous programming language Quartz (Schneider, 2001b; Schneider, 2006). We then used the formal verification methods of the Averest framework (Schneider and Schuele, 2005) to check the correctness of our implementation with respect to given safety conditions.

The application of formal methods to verify the correctness of a robot system is not new: In (Diethers et al., 2003), the model checker HyTech was used to analyse a robot program based on skill primitive nets. While HyTech considers hybrid automata as system models to model continuous values of physical properties, our approach is based on discrete transition systems that benefit directly from symbolic model checking techniques. The use of symbolic model checking for the formal verification of a robot system has been reported in (Sharygina et al., 2004). In contrast to our approach, however, the verification is not integrated with code generation.

The use of synchronous languages (Benveniste et al., 2003) for the implementation of safety critical control systems of robots is also not new: In (Sowmya et al., 2002), a controller for a mobile robot (Rug Warrior) has been implemented in the synchronous language Esterel (Berry, 1998). In (Kim and Kang, 2005), the core of the Samsung Home Robot SHR100 has been re-engineered to be verified by means of the Esterel framework. In contrast to our approach, the systems considered in (Sowmya et al., 2002) and (Kim and Kang, 2005) were quite small control programs with only a few control states.

In contrast to the previous work in this area, our approach considers verification as well as (verified) code generation. Moreover, the considered system is not simply a small control program, but a behaviour-based control network with difficult interdependencies.

The outline of the paper is as follows: In the next two sections, we give some details on the outdoor robot RAVON and its behaviour-based control system. In Section 4, we describe the verification framework Averest and give some basics about synchronous languages and model checking. Section 5 contains the results of the verification. Finally, we conclude with a summary and directions for future work.

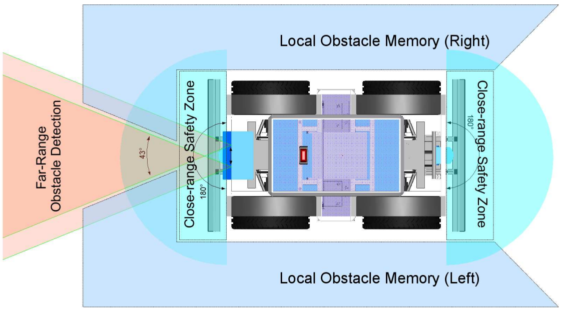

Local Obstacle Memory (Right)
Far-Range Obstacle Detection
43°
Close-range Safety Zone
180°
Close-range Safety Zone
180°
Local Obstacle Memory (Left)

Figure 2: Regions monitored by the obstacle detection and avoidance facilities.

2 THE OUTDOOR ROBOT RAVON

RAVON is a four wheeled off-road vehicle measuring 2.35 m in length and 1.4 m in width and weighting 400 kg. The vehicle features a four wheel drive with independent motors yielding maximal velocities of 3 m/s. In combination with its off-road tires, the vehicle can climb slopes of 100% inclination predestining it for the challenges in rough terrain. Front and rear axis can be steered independently which supports agile advanced driving manoeuvres like double Ackerman and parallel steering.

In order to navigate in a self-dependent fashion, RAVON has been equipped with several sensors. For self localisation purposes, the robot uses its odometry, a custom design inertial measurement unit, a magnetic field sensor, and a DGPS receiver. The sensor data fusion is performed by a Kalman filter (Schmitz et al., 2006) which calculates an estimated pose in three dimensions. Due to gravity measurements of the inertial measurement unit, the control system receives quite precise absolute data for the roll and pitch angle of the vehicle. These are fed into behaviours that are responsible for supervising whether the vehicle might tip over due to critical inclination.

In order to protect the vehicle in respect to obstacles, several safety regions are observed by different sensor systems (Schäfer and Berns, 2006) (see Fig. 2). First of all, hindrances can be detected using the stereo camera system mounted at the front of the vehicle. The stereo camera's narrow field of vision is compensated by local obstacle memories to either side of the robot realising a short-term representation of detected obstacles. This obstacle detection facility is complemented with two laser range finders (field of vision: 180 degrees, angular resolution: 0.5 degrees, distance resolution: about 0.5 cm) monitoring the environment nearby the vehicle. Data from both sources of proximity data is used for obstacle avoid-

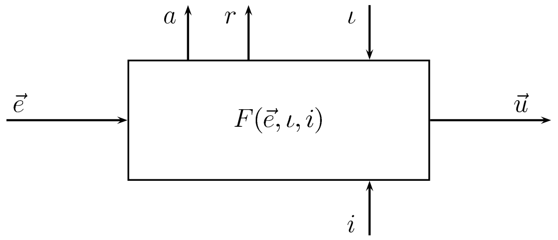


Figure 3: Basic behaviour module.

ance by appropriate behaviours, the fusion of which is performed inside the behaviour network.In case of emergency, the system is stopped on collision by the safety bumpers which are directly connected to the emergency stop to ensure maximal safety.

3 BEHAVIOUR-BASED CONTROL SYSTEM OF RAVON

This section introduces the components used for building up the behaviour-based network controlling RAVON.

3.1 Behaviour Module

The fundamental unit of the proposed control architecture is the behaviour module (see Fig. 3). Each atomic behaviour is wrapped into such a module with a defined interface. Behaviours can be described as three-tuples of the form

$$\mathcal{B} = (r, a, F) \tag{1}$$

where r is the target rating function, a is the activity function, and F is the transfer function of the behaviour. Additionally each behaviour receives an input vector $\vec{e}$, an activation ι, and an inhibition i and generates an output vector $\vec{u}$.

More precisely behaviours receive data needed for fulfilling their work via the sensor input $\vec{e} \in \mathfrak{R}^n$ which can be composed of sensory data or information from other behaviours. The output vector $\vec{u} \in \mathfrak{R}^m$ transmits data generated by the behaviour. This output describes the influence a behaviour can have on the environment or on other behaviours.

Each behaviour owns an input determining its activation $\iota \in [0,1]$. In this notation $\iota = 0$ indicates deactivation and $\iota = 1$ a fully activated behaviour. Values between 0 and 1 refer to a partially activated behaviour. Activation can be used to adjust the relevance of competing behaviours. The inverse effect is achieved by inhibition $i \in [0,1]$ which is used to reduce the activation of a behaviour: $i = 1$ refers to full inhibition, $i = 0$ to no inhibition.

Information about the activity of a behaviour is provided by the output $a \in [0,1]$. The maximal activity is described by $a = 1$, inactivity by $a = 0$. It is defined by the activity function

$$a(\vec{e},\iota,i) = a_{\text{int}}(\vec{e}) \cdot \iota \cdot (1-i) \tag{2}$$

where $a_{\text{int}}(\vec{e}) \in [0,1]$ is an internal function representing the intended activity of the behaviour.

The target rating $r \in [0,1]$ deals as an indicator for the contentment of a behaviour. A value of $r = 0$ indicates that the behaviour is content with the actual state, while $r = 1$ shows maximal dissatisfaction.

The output vector $\vec{u}$ of a behaviour is determined using its transfer function $F(\vec{e},\iota,i)$ where

$$F : \mathfrak{R}^n \times [0,1]^2 \to \mathfrak{R}^m, \quad F(\vec{e},\iota,i) = \vec{u}$$

This function provides the intelligence of a behaviour, calculating actions depending on input values and internal representations. This can be a reactive respond to input values but also a more complex calculation as a state machine or sophisticated algorithms. Both reactive and deliberative behaviours can be implemented that way.

3.2 Example Behaviour `Roll Stop`

In this section a behaviour reacting on high roll is described in order to exemplify the behaviour properties described before. As the behaviour wants to stop the vehicle, the output $\vec{u}$ is a velocity of zero. Therefore the transfer function is:
$\vec{u} = v_{\text{out}} = 0$
This velocity has an effect if the activity a rises:
$a = \texttt{Threshold}(\texttt{roll}) \cdot \iota \cdot (1-i)$
Here $\texttt{Threshold}(\texttt{roll})$ is a function returning 0 or 1 depending on the roll value being below or above a given threshold. The activity is scaled by the activation ι and the inhibition i as stated above. Similarly the target rating r is
$r = \texttt{Threshold}(\texttt{roll})$
In case of normal roll angles the behaviour is content ($r = 0$) while for high roll angles it is dissatisfied ($r = 1$).

3.3 Fusion Behaviour Module

In case of competing behaviours so called fusion behaviours (see figure 4) are used for coordination. The underlying assumption of the fusion of output values is that behaviours having a high activity deserve a higher influence on the control than those with lower

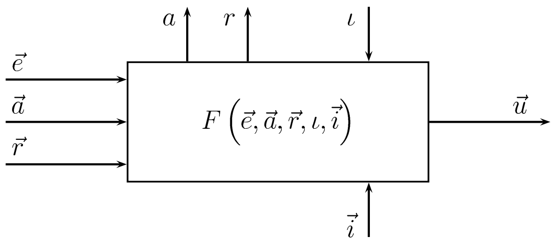


Figure 4: Fusion behaviour module.

activity. The interface of fusion behaviours implements a refinement of usual behaviours. For each of the competing behaviours $\mathcal{B}_i$ the activity (indicated by $\vec{a}$), target rating (indicated by $\vec{r}$) and output vector $\vec{u}$ is provided. The output vector is fed into the fusion behaviour as $\vec{e}$. Additionally there is a fusion of inhibiting behaviours by the inhibition inputs $\vec{i}$. The transfer function then is the fusion function $f(\vec{a},\vec{e})$ which processes these input values to a merged output control vector $\vec{u}$.

The fusion function can have several implementations, in this work the weighted fusion is used: Here the control values are weighted with the activity of the corresponding behaviour, leading to a fusion function f_{weighted}, where

$$\vec{u} = f_{\text{weighted}}(a_0,\vec{u}_0,\dots,a_{n-1},\vec{u}_{n-1}) = \frac{\sum\limits_{j=0}^{n-1} a_j \cdot \vec{u}_j}{\sum\limits_{k=0}^{n-1} a_k} \tag{3}$$

The activity is set according to the weighted input activities, the activation, and the maximally activated inhibiting behaviour:

$$a = \frac{\sum\limits_{j=0}^{n-1} a_j^2}{\sum\limits_{k=0}^{n-1} a_k} \cdot \iota \cdot (1 - i_m) \text{ where } i_m = \max_l(i_l)$$

The target rating of a fusion behaviour indicates its goal to satisfy highly activated input behaviours and is calculated as follows:

$$r = \frac{\sum\limits_{j=0}^{n-1} a_j \cdot r_j}{\sum\limits_{k=0}^{n-1} a_k}$$

The weighted fusion function provides a subtle gradation of coordinating behaviour control outputs regarding their activity.

3.4 Behaviour Network of Ravon

The behaviour network implemented on RAVON is shown in Fig. 5. Here the flow of control data is marked as blue drawn through line while data derived from sensors as well as interpreted sensor data (e.g. the activity output of behaviours used as inhibition input for fusion behaviours) is marked as red dashed line.

The behaviour network comprises three control chains affecting desired rotation, sideward motion, and velocity of the vehicle. The rotational and the sideward components are influenced by obstacle avoidance behaviours. For safety reasons the velocity is adjusted according to obstacle proximity and critical vehicle inclination. In this behaviour network the following types of behaviours are used:

- Emergency Stop: Stop due to laser scanner data, positive or negative obstacles detected by the camera system; independently used for forward and backward motion.

- Slow Down: Reduce velocity due to obstacle proximity (laser scanner, positive/negative obstacles); independently used for forward and backward motion.

- Keep Distance Rotational: Turn away from obstacles (laser scanner, positive/negative obstacles); independently used for both sides of the vehicle.

- Keep Distance Sideward: Accomplish sideward motion due to obstacles at the side; independently used for both sides of the vehicle.

- Evasion: Evade obstacles at the front by arbitrating between the keep distance behaviours.

- Point Access: Accessing a given position.

- Point Access Ranking: Perform ranking manoeuvres accounting for kinematic constraints.

- Trace Back: Follow just driven path backwards in order to escape dead ends.

The advantage of this approach is the emergent vehicle behaviour leading to unforeseen, but suitable reaction on several external influences at a time. However, especially the maintenance of vehicle and person safety requires methods for guaranteeing fundamental characteristics of the vehicle motion e.g. in critical situations. Therefore, it is necessary to formally verify the behaviour network with respect to a given set of specifications.

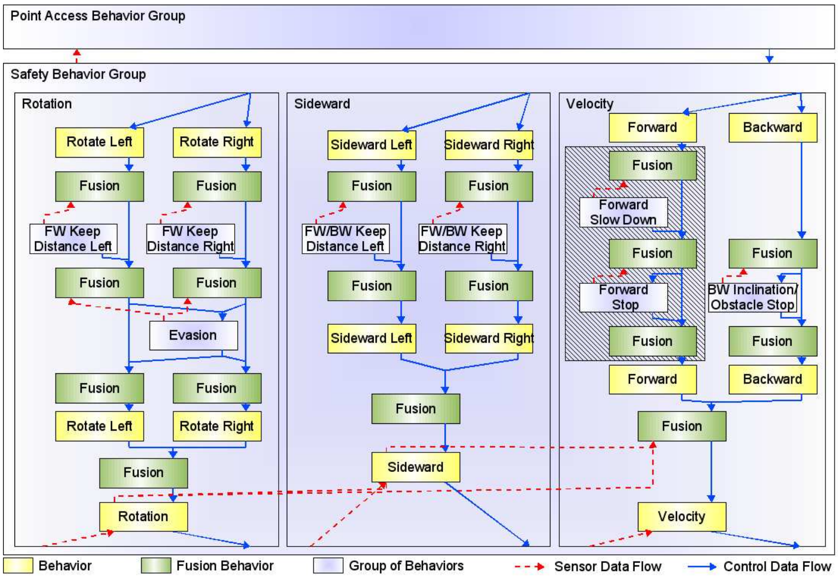


Figure 5: Behaviour network of RAVON.

4 THE AVEREST SYSTEM

In this section, we describe the Averest[1] framework (Schneider and Schuele, 2005; Schneider and Schuele, 2006) that provides tools for verifying temporal properties of synchronous programs (Benveniste et al., 2003; Halbwachs, 1993) as well as for compiling these programs to equivalent hardware and software systems. In particular, many formal verification techniques, including model checking of temporal properties of finite and infinite state systems are available. In Averest, a system is described using the Esterel–like synchronous programming language Quartz (Schneider, 2001a), and specifications can be given in temporal logics such as LTL and CTL (Schneider, 2003). Currently, Averest consists of the following tools:

- ruby: a compiler for translating Quartz programs to finite and infinite state transition systems
- beryl: a symbolic model checker for finite and infinite state transition systems
- topaz: a code generator to convert transition systems into hardware and/or software

Figure 6 shows the typical design flow. A given Quartz program is first translated to a symbolically represented transition system in Averest's Interchange Format AIF that is based on XML. The AIF description can then be used for verification and code generation. Moreover, there are interfaces to third–party tools, e.g. other model checkers such as SMV (McMillan, 1992). In the remainder of this section, we describe some background information on Averest focusing on the verification of behaviour networks.

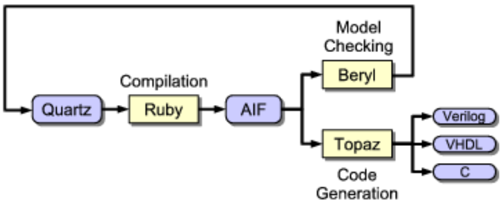


Figure 6: Averest design flow.

The basic paradigm of synchronous languages (Benveniste et al., 2003; Halbwachs, 1993) is the distinction between *micro and macro steps* in a program. From a programmer's point of view, micro steps do not take time, whereas macro steps take one unit of time. Hence, consumption of time is explicitly programmed by partitioning the program into macro steps. This programming model, referred to as perfect synchrony (Benveniste et al., 2003; Halbwachs, 1993), together with a deterministic form of concurrency allows the compilation of *multi–threaded synchronous programs* to deterministic single–threaded code. A distinct feature of synchronous languages is their *detailed formal semantics* that is usually given by means of transition rules in structural operational semantics. This makes synchronous languages attractive for safety–critical applications where formal verification is mandatory.

After translating a Quartz program to a transition system, it can be verified using symbolic model

[1] http://www.averest.org

checking techniques. For that purpose, the specifications are given as a set of temporal logic formulas that describe the desired properties of the system. The most frequently used properties are safety and liveness properties. Intuitively, a safety property states that a condition invariantly holds on a given path of the transition system. Similarly, a liveness property states that a condition holds at least once on a given path. As an example for a safety property, the formula AGφ states that φ holds on all possible computation paths of the system. An example for a liveness property is the formula EFφ, stating that there exists a path such that φ eventually holds.

A breakthrough in formal verification was achieved in the early nineties, where it was observed that finite sets can be efficiently represented by means of binary decision diagrams (BDDs), a canonical normal form for propositional logic formulas (Bryant, 1986). The development of BDDs was a cornerstone for *symbolic model checking* procedures based on fixpoint computations (Burch et al., 1990) (see textbooks like (Clarke et al., 1999; Schneider, 2003) for more details). With sophisticated implementations and refinements of symbolic model checking, it has become possible to verify systems of industrial size, and to detect errors that can hardly be found using simulation (Clarke and Wing, 1996).

5 FORMAL VERIFICATION OF THE BEHAVIOUR NETWORK

In case of RAVON, the most important behaviour network property is the control of the vehicle velocity due to obstacles or critical inclination. The behaviours affecting the control in this respect are marked hatched in Fig. 5. In order to formally verify this part of the behaviour network, the corresponding behaviours have been implemented in the synchronous language Quartz. In this way, the correctness of every single behaviour can be shown by means of a symbolic model checker. By forming a group of the mentioned behaviours, it is even possible to verify specifications concerning the overall behaviour of the complete velocity control part of the behaviour network. Moreover, the Quartz code can be exported to C code and can be wrapped into a module that replaces the original (unverified) code. As the output of the verified module is directly transferred to the vehicle motion actuators, it can be guaranteed that slowing down and stopping has the intended effect.

The verified and synthesized parts of RAVON's behaviour network are depicted in in Fig. 7. It shows the following structure: If none of the slow-down and stop behaviours is active, the velocity given by higher layers (*ci_velocity*) is piped through the three fusion behaviours without change to *co_velocity*. As soon as one of the mentioned behaviours becomes active (in case the inclination rises above a threshold or the obstacle distance becomes too low), the active behaviour uses its activity to inhibit the fusion behaviour which is above it. At the same time, it proposes a velocity of zero to the underlying fusion behaviour. This fusion behaviour calculates a weighted sum (using the input activities) of the input velocity values. The more active a behaviour becomes, the less active the fusion behaviour above is. Therefore, the influence of the slow-down behaviour rises and the velocity output of the underlying fusion behaviour decreases. This mechanism is implemented on two layers here.

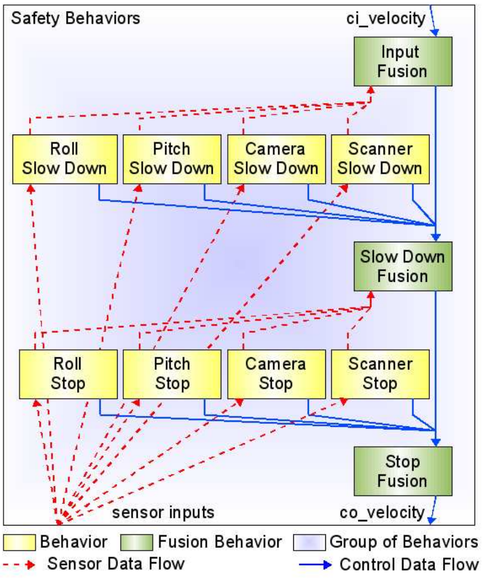


Figure 7: Structure of the verified part of RAVON's behaviour network.

For this behaviour network, we checked eight specifications including the ones we list below. In this context, uppercase words indicate system parameters, `ci_` indicates controller inputs, `co_` controller outputs, and `si_` sensor inputs. Numbers are marked as unsigned integers (`0u`).

- The output velocity is never higher than the input velocity:

```
A G (co_velocity <= ci_velocity);
```

- In case of no near obstacle and tolerable inclination, the output velocity equals the input velocity. Therefore, the vehicle is not slowed down without a reason:

```
A G ((si_camera_distance
  >= MAX_VELOCITY_OBSTACLE_DISTANCE)
```

```
    & (si_scanner_distance
     >= MAX_VELOCITY_OBSTACLE_DISTANCE)
    & (si_roll
     <= MAX_VELOCITY_INCLINATION)
    & (si_pitch
     <= MAX_VELOCITY_INCLINATION)
    -> co_velocity == ci_velocity);
```

- If very high inclination or very near obstacles occur, the output velocity is set to zero, i.e., the vehicle stops:

```
A G (si_roll >= STOP_INCLINATION
 -> co_velocity == 0u);
A G (si_pitch >= STOP_INCLINATION
 -> co_velocity == 0u);
A G (si_camera_distance
  <= STOP_OBSTACLE_DISTANCE
 -> co_velocity == 0u);
A G (si_scanner_distance
  <= STOP_OBSTACLE_DISTANCE
 -> co_velocity == 0u);
```

- In case of a rising roll angle, the vehicle slows down. Similar specifications hold for the pitch angle and for near obstacles:

```
A G (ci_velocity == MAX_VALUE
 & si_camera_distance
  > MIN_VELOCITY_OBSTACLE_DISTANCE
 & si_scanner_distance
  > MIN_VELOCITY_OBSTACLE_DISTANCE
 & si_pitch
  < MIN_VELOCITY_INCLINATION
 & si_roll
  > MAX_VELOCITY_INCLINATION
 & si_roll
  < MIN_VELOCITY_INCLINATION
 -> co_velocity < MAX_VALUE
  & co_velocity > 0u );
```

In order to avoid vacuous specifications, formulas of the type $\mathsf{AG}(\varphi \rightarrow \psi)$ are always complemented with $\mathsf{EF}\varphi$ (not shown here). In this way, it is guaranteed that the antecedent of the implication is not always false. All of our specifications can be expressed in the temporal logic CTL. Hence, we can use state-of-the-art model checking techniques to verify these properties. Using global model checking, the model checkers perform a complete traversal of the reachable states and thereby check whether the given specifications hold. Using local model checking, the model checkers perform some sort of an induction proof to avoid a complete state space traversal.

BDD-based symbolic model checkers do usually not support floating point numbers. Therefore, integers with a given bitwidth are used for the implementation in Quartz instead of floating point numbers. In order to integrate the verified module in the control system, a conversion from floating point to fixpoint numbers is performed for control and sensor values.

In our case study, global model checking was able to check the specifications up to a sufficiently large bitwidth. We used CadenceSMV as backend of Averest. In the following table, we list experimental results for some bitwidths. For each bitwidth, we list the number of reachable states, the runtime that was necessary to verify the system, and the number of required BDD nodes for the entire verification of the eight specifications. All experiments were performed on a PentiumIV with 512 MByte main memory.

Table 1: Experimental results of the verification process.

bits	states	runtime (s)	BDD nodes
3	1024	0.95	453
4	8192	1.59	10012
5	65536	2.54	10063
6	524288	3.87	10521
7	4194204	5.54	15399
8	33554432	8.85	49500

In the process of implementing the behaviour network, the question arose if the fusion behaviours could implement a maximum fusion function (i.e. the most active behaviour has full influence) instead of the weighted fusion function. By means of formal verification, it was possible to show that in this case not all specifications were valid. Depending on the implementation, there was either no slowing down of the vehicle (but only abrupt stopping) or it was possible that the velocity output was higher than the velocity input. Experimental changes concerning the structure and implementation of the behaviour network can therefore be performed with immediate feedback about the correct properties stated in the specifications.

6 CONCLUSIONS AND FUTURE WORK

We presented an approach to the formal verification of a behaviour-based control network. Without the need of testing, it is guaranteed that the specified properties are valid for all possible input traces. Of course, it is necessary to verify more parts of the system or even the complete behaviour-based network. Due to the enormous number of states, this inevitably leads to the need of improving the model checking approach. Therefore, methods like modular model checking and abstraction will have to be analysed in this respect. The uniformity of the behaviours is seen to be an advantage in this context that can be exploited by tailored verification and abstraction techniques.

ACKNOWLEDGEMENTS

The research work presented in this paper is funded by the German Federal State of Rhineland-Palatinate within the excellence cluster ”Dependable Adaptive Systems and Mathematical Modeling”.

REFERENCES

Benveniste, A., Caspi, P., Edwards, S., Halbwachs, N., Le Guernic, P., and de Simone, R. (2003). The synchronous languages twelve years later. *Proceedings of the IEEE*, 91(1):64–83.

Berry, G. (1998). The foundations of Esterel. In Plotkin, G., Stirling, C., and Tofte, M., editors, *Proof, Language and Interaction: Essays in Honour of Robin Milner*. MIT.

Bryant, R. (1986). Graph-based algorithms for Boolean function manipulation. *IEEE Transactions on Computers*, C-35(8):677–691.

Burch, J., Clarke, E., McMillan, K., Dill, D., and Hwang, L. (1990). Symbolic model checking: 10^{20} states and beyond. In *Symposium on Logic in Computer Science (LICS)*, pages 1–33, Washington, D.C. IEEE Computer Society.

Clarke, E., Grumberg, O., and Peled, D. (1999). *Model Checking*. MIT, London, England.

Clarke, E. and Wing, J. (1996). Formal methods: State of the art and future directions. Technical Report CMU-CS-96-178, Carnegie Mellon University. ftp://reports.adm.cs.cmu.edu/usr/anon/1996/CMU-CS-96-178.ps.

Diethers, K., Firley, T., Krger, T., and Thomas, U. (2003). A new framework for task oriented sensor based robot programming and verification. In *International Conference on Advanced Robotics (ICAR)*, pages 1208–1214, Coimbra,Portugal. IEEE Computer Society.

Halbwachs, N. (1993). *Synchronous programming of reactive systems*. Kluwer.

Kim, M. and Kang, K. (2005). Formal construction and verification of home service robots: A case study. In Peled, D. and Tsay, Y.-K., editors, *International Symposium on Automated Technology for Verification and Analysis (ATVA)*, volume 3707 of *LNCS*, pages 429–443, Taipei, Taiwan. Springer.

McMillan, K. (1992). The SMV system, symbolic model checking - an approach. Technical Report CMU-CS-92-131, Carnegie Mellon University.

Schäfer, H. and Berns, K. (2006). Ravon - an autonomous vehicle for risky intervention and surveillance. In *International Workshop on Robotics for risky intervention and environmental surveillance - RISE*.

Schmitz, N., Proetzsch, M., and Berns, K. (2006). Pose estimation in rough terrain for the outdoor vehicle ravon. In *37th International Symposium on Robotics (ISR)*.

Schneider, K. (2001a). Embedding imperative synchronous languages in interactive theorem provers. In *Conference on Application of Concurrency to System Design (ACSD)*, pages 143–156, Newcastle upon Tyne, UK. IEEE Computer Society.

Schneider, K. (2001b). *Exploiting Hierarchies in Temporal Logics, Finite Automata, Arithmetics, and μ-Calculus for Efficiently Verifying Reactive Systems*. Habilitation Thesis. University of Karlsruhe.

Schneider, K. (2003). *Verification of Reactive Systems – Formal Methods and Algorithms*. Texts in Theoretical Computer Science (EATCS Series). Springer.

Schneider, K. (2006). The synchronous programming language Quartz. Internal Report (to appear), Department of Computer Science, University of Kaiserslautern.

Schneider, K. and Schuele, T. (2005). Averest: Specification, verification, and implementation of reactive systems. In *Conference on Application of Concurrency to System Design (ACSD)*, St. Malo, France. participant’s proceedings.

Schneider, K. and Schuele, T. (2006). A framework for verifying and implementing embedded systems. In Straube, B. and Freibothe, M., editors, *Methoden und Beschreibungssprachen zur Modellierung und Verifikation von Schaltungen und Systemen*, pages 242–247, Dresden, Germany. GI/ITG/GMM, Fraunhofer Institut fr Integrierte Schaltungen, ISBN 3-9810287-1-6.

Schuele, T. and Schneider, K. (2006). Bounded model checking for infinite state systems. *Formal Methods in System Design (FMSD)*. DOI 10.1007/s10703-006-0019-9.

Sharygina, N., Browne, J., Xie, F., Kurshan, R., and Levin, V. (2004). Lessons learned from model checking a NASA robot controller. *Formal Methods in System Design (FMSD)*, 25(2-3):241–270.

Sowmya, A., So, D., and Tang, W. (2002). Design of a mobile robot controller using Esterel tools. *Electronic Notes in Theoretical Computer Science (ENTCS)*, 65(5). Workshop on Synchronous Languages, Applications, and Programming (SLAP).

A MODIFIED IMPULSE CONTROLLER FOR IMPROVED ACCURACY OF ROBOTS WITH FRICTION

Stephen van Duin, Christopher D. Cook, Zheng Li and Gursel Alici
Faculty of Engineering, University of Wollongong, Northfields Avenue, Gwynnville, Australia
svanduin@uow.edu.au, ccook@uow.edu.au, zheng@elec.uow.edu.au, gursel@uow.edu.au

Keywords: Impulsive control, static friction, limit cycle, stick-slip, impulse shape, friction model, accuracy.

Abstract: This paper presents a modified impulse controller to improve the steady state positioning of a SCARA robot having characteristics of high non-linear friction. A hybrid control scheme consisting of a conventional PID part and an impulsive part is used as a basis to the modified controller. The impulsive part uses short width torque pulses to provide small impacts of force to overcome static fiction and move a robot manipulator towards its reference position. It has been shown that this controller can greatly improve a robot's accuracy. However, the system in attempting to reach steady state will inevitably enter into a small limit cycle whose amplitude of oscillation is related to the smallest usable impulse. It is shown in this paper that by modifying the impulse controller to adjust the width of successive pulses, the limit cycle can be shifted up or down in position so that the final steady state error can be even further reduced.

1 INTRODUCTION

Precision robot manufacturers continually strive to increase the accuracy of their machinery in order to remain competitive. The ability of a robot manipulator to position its tool centre point to within a very high accuracy, allows the robot to be used for more precise tasks. For positioning of a tool centre point, the mechanical axes of a robot will be required to be precisely controlled around zero velocity where friction is highly non-linear and difficult to control.

Non-linear friction is naturally present in all mechanisms and can cause stick-slip during precise positioning. In many instances, stick-slip has been reduced or avoided by modifying the mechanical properties of the system; however this approach may not always be practical or cost effective. Alternatively, advances in digital technology have made it possible for the power electronics of servomechanisms to be controlled with much greater flexibility. By developing better controllers, the unfavourable effects of non-linear friction may be reduced or eliminated completely.

Impulse control has been successfully used for accurate positioning of servomechanisms with high friction where conventional control schemes alone have difficulty in approaching zero steady state error. Static and Coulomb friction can cause a conventional PID controller having integral action (I), to overshoot and limit cycle around the reference position. This is a particular problem near zero velocities where friction is highly non linear and the servomechanism is most likely to stick-slip. Despite the above difficulties, PID controllers are still widely used in manufacturing industries because of their robustness to parameter uncertainty and unknown disturbances.

Stick-slip can be reduced or eliminated by using impulsive control near or at zero velocities. The impulsive controller is used to overcome static friction by impacting the mechanism and moving it by microscopic amounts. By combining the impulsive controller and conventional controller together, the PID part can be used to provide stability when moving towards the reference position while the impulse controller is used to improve accuracy for the final positioning where the error signal is small.

By applying a short impulse of sufficient force plastic deformation occurs between the asperities of mating surfaces resulting in permanent controlled movement. If the initial pulse causes insufficient movement, the impulsive controller produces additional pulses until the position error is reduced to a minimum.

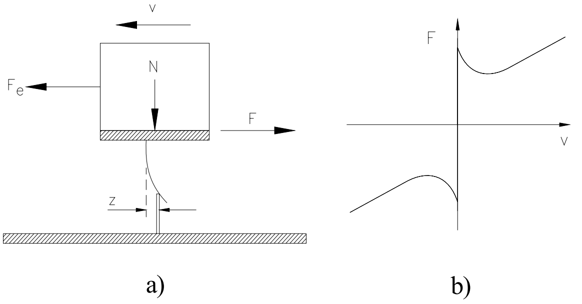


Figure 1: Bristle model; Figure a) shows the deflection of a single bristle. Figure b) shows the resulting static friction model for a single instance in time.

A number of investigators have devised impulsive controllers which achieve precise motion in the presence of friction by controlling the height or width of a pulse. Yang and Tomizuka (Yang et al, 1988) applied a standard rectangular shaped pulse whereby the height of the pulse is a force about 3 to 4 times greater than the static friction to guarantee movement. The width of the pulse is adaptively adjusted proportional to the error and is used to control the amount of energy required to move the mechanism towards the reference positioning. Alternatively, Popovic (Popovic et al, 2000) described a fuzzy logic pulse controller that determines both the optimum pulse amplitude and pulse width simultaneously using a set of membership functions. Hojjat and Higuchi (Hojjat et al, 1991) limited the pulse width to a fixed duration of 1ms and vary the amplitude by applying a force about 10 times the static friction. Rathbun et al (Rathbun et al, 2004) identify that a flexible-body plant can result in a position error limit cycle and that this limit cycle can be eliminated by reducing the gain using a piecewise-linear-gain pulse width control law.

In a survey of friction controllers by Armstrong-Hélouvry (Armstrong- Hélouvry et al, 1994), it is commented that underlying the functioning of these impulsive controllers is the requirement for the mechanism to be in the stuck or stationary position before subsequent impulses are applied. Thus, previous impulse controllers required each small impacting pulse to be followed by an open loop slide ending in a complete stop.

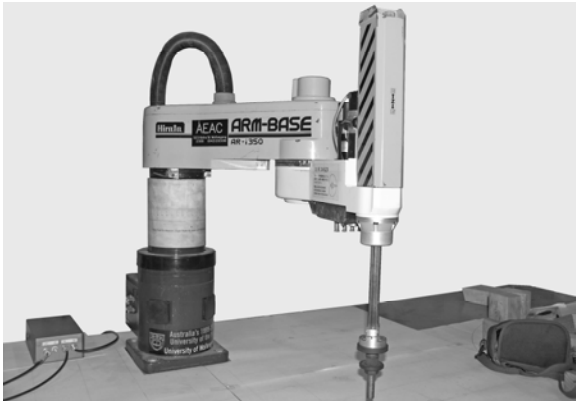


Figure 2: The Hirata SCARA robot.

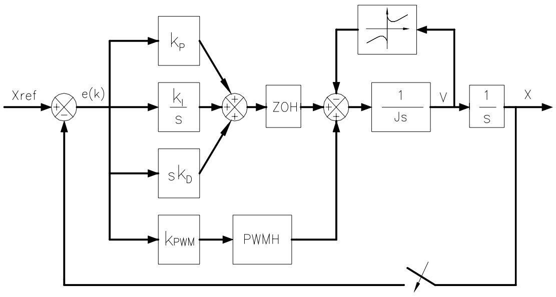


Figure 3: Block diagram of the experimental system controller.

In this paper, a hybrid PID + Impulsive controller is used to improve the precision of a servomechanism under the presence of static and Coulomb friction. The design and functioning of the controller does not require the mechanism to come to rest between subsequent pulses, making it suitable for both point to point positioning and speed regulation. The experimental results of this paper show that the shape of the impulse can be optimised to increase the overall precision of the controller. It is shown that the smallest available movement of the servomechanism can be significantly reduced without modification to the mechanical plant.

2 MODELLING AND EXPERIMENTAL SYSTEM

2.1 Friction Model

On a broad scale, the properties of friction are both well understood and documented. Armstrong-Hélouvry (Armstrong- Hélouvry et al, 1994) have surveyed some of the collective understandings of how friction can be modelled to include the complexities of mating surfaces at a microscopic level. Canudas de Wit (Canudas de Wit et al, 1995) add to this contribution by presenting a new model that more accurately captures the dynamic phenomena of rising static friction (Rabinowicz, 1958), frictional lag (Rabinowicz, 1958), varying break away force (Johannes et al, 1973), (Richardson et al, 1976), dwell time (Kato et al, 1972), pre-sliding displacement (Dahl, 1968), (Dahl, 1977), (Johnson, 1987) and Stribeck effect (Olsson, 1996). The friction interface is thought of as a contact between elastic bristles. When a tangential force is applied, the bristles deflect like springs which give rise to the friction force (Canudas de Wit et al, 1995); see Figure 1(a). If the effective applied force F_e exceeds the bristles force, some of the bristles will be made to slip and permanent plastic movement occurs between each of the mating surfaces. The set of equations governing the dynamics of the bristles are given by (Olsson, 1996):

$$\frac{dz}{dt} = v - \frac{|v|}{g(v)} z \tag{1}$$

$$g(v) = \frac{1}{\sigma_0}\left(F_C + \left(F_s - F_C\right)e^{-(v/v_s)^2}\right) \tag{2}$$

$$F = \sigma_0 z + \sigma_1(v)\frac{dz}{dt} + F_v v \tag{3}$$

$$\sigma_1(v) = \sigma_1 e^{-(v/v_d)^2} \tag{4}$$

where v is the relative velocity between the two surfaces and z is the average deflection of the bristles. σ_0 is the bristle stiffness and σ_1 is the bristle damping. The term v_s is used to introduce the velocity at which the Stribeck effect begins while the parameter v_d determines the velocity interval around zero for which the velocity damping is active. Figure 1(b) shows the friction force as a function of velocity. F_s is the average static friction while F_C is the average Coulomb friction. For very low velocities, the viscous friction F_v is negligible but is included for model completeness. F_s, F_C, and F_v are all estimated experimentally by subjecting a real mechanical system to a series of steady state torque responses. The parameters σ_0, σ_1, v_s and v_d are also determined by measuring the steady state friction force when the velocity is held constant (Canudas de Wit et al, 1995).

2.2 Experimental System

For these experiments, a Hirata ARi350 SCARA (Selective Compliance Assembly Robot Arm) robot was used. The Hirata robot has four axes named A, B, Z and W. The main rotational axes are A-axis (radius 350mm) and B-axis (radius 300mm) and they control the end-effector motion in the horizontal plane. The Z-axis moves the end-effector in the vertical plane with a linear motion, while the W-axis is a revolute joint and rotates the end effector about the Z-axis. A photograph of the robot is shown in Figure 2.

For these experiments, only the A and B axis of the Hirata robot are controlled. Both the A and B axes have a harmonic gearbox between the motor and robot arm. Their gear ratios are respectively 100:1 and 80:1. All of the servomotors on the Hirata robot are permanent magnet DC type and the A and B axis motors are driven with Baldor® TSD series DC servo drives. Each axis has characteristics of high non-linear friction whose parameters are obtained by direct measurement. For both axes, the static friction is approximately 1.4 times the Coulomb friction.

Matlab’s xPC target oriented server was used to provide control to each of the servomotor drives. For these experiments, each digital drive was used in current control mode which in effect means the output voltage from the 12-bit D/A converter gives a torque command to the actuator’s power electronics. The system controller was compiled and run using Matlab’s real time xPC Simulink® block code. A 12-bit A/D converter was used to read the actuator’s shaft encoder position signal.

2.3 PID + Impulse Hybrid Controller

Figure 3 shows the block diagram of a PID linear controller + impulsive controller. This hybrid controller has been suggested by Li (Li et al, 1998) whereby the PID driving torque and impulsive controller driving torque are summed together. It is unnecessary to stop at the end of each sampling period and so the controller can be used for both position and speed control.

The controller can be divided into two parts; the upper part is the continuous driving force for large scale movement and control of external force disturbances. The lower part is an additional proportional controller k_{pwm} with a pulse width modulated sampled-data hold (PWMH), and is the basis of the impulsive controller for the control of stick-slip.

The system controller is sampled at 2 kHz. The impulse itself is sampled and applied at one twentieth of the overall sampling period (i.e. 100 Hz) to match the mechanical system dynamics. Figure 4 shows a typical output of the hybrid controller for one impulse sampling period τ_s. The pulse with height f_p is added to the PID output. Because the PID controller is constantly active, the system has the ability to counteract random disturbances applied to the servomechanism. The continuous part of the controller is tuned to react to large errors and high velocity, while the impulse part is optimized for final positioning where stiction is most prevalent.

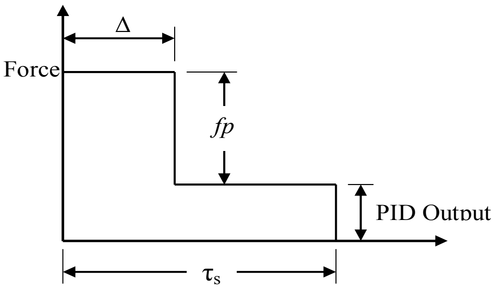


Figure 4: Friction controller output.

For large errors, the impulse width approaches the full sample period τ_s, and for very large errors, it transforms into a continuous driving torque. When this occurs, the combined control action of the PID controller and the impulsive controller will be continuous. Conversely, for small errors, the PID output is too small to have any substantial effect on the servomechanism dynamics.

The high impulse sampling rate, combined with a small error, ensures that the integral (I) part of the PID controller output has insufficient time to rise and produce limit cycling. To counteract this loss of driving torque, when the error is below a threshold, the impulsive controller begins to segment into individual pulses of varying width and becomes the primary driving force. One way of achieving this is to make the pulse width Δ determined by:

$$\Delta = \frac{k_{pwm} \cdot e(k)\tau_s}{f_p} \quad \text{if } \mathrm{k}_{\mathrm{pwm}} \cdot |e(k)| \leq |f_p|$$
$$\Delta = \tau_s \qquad \text{otherwise} \qquad (6)$$

In (6)

$$f_p = |f_p| \cdot sign(e(k)) \qquad (7)$$

where e(k) is the error input to the controller, $|f_p|$ is a fixed pulse height greater than the highest static friction and τ_s is the overall sampling period. For the experimental results of this paper, the impulsive sampling period τ_s was 10ms and the pulse width could be incrementally varied by 1ms intervals. The pulse width gain k_{pwm}, is experimentally determined by matching the mechanism's observed displacement d to the calculated pulse width t_p using the equation of motion:

$$d = \frac{f_p(f_p - f_C)}{2mf_C} t_p^2, \quad f_p > 0 \qquad (8)$$

The gain is iteratively adjusted until the net displacement for each incremental pulse width is as small as practical.

2.4 Minimum Pulse Width

The precision of the system is governed by the smallest incremental movement which will be produced from the smallest usable width pulse. Because the shape of the pulse is affected by the system's electrical circuit response, a practical limit is placed on the amplitude of the pulse over very short durations and this restricts the amount of energy that can be contained within a very thin pulse. Consequently, there exists a minimum pulse width that is necessary to overcome the static friction and guarantee plastic movement.

For the Hirata robot, the minimum pulse width guaranteeing plastic displacement was determined to be 2ms and therefore the pulse width is adjusted between 2 and 10ms. Any pulse smaller than 2ms results in elastic movement of the mating surfaces in the form of pre-sliding displacement. In this regime, short impulses can produce unpredictable displacement or even no displacement at all. In some cases, the mechanism will spring back greater than the forward displacement resulting in a larger error. Figure 5 shows the displacement of the experimental system of five consecutive positive impulses followed by five negative impulses. The experiment compares impulses of width 2ms and 1.5ms. For impulses of 2ms, the displacement is represented by the consistent staircase movement. For a width of 1.5ms, the displacement is unpredictable with

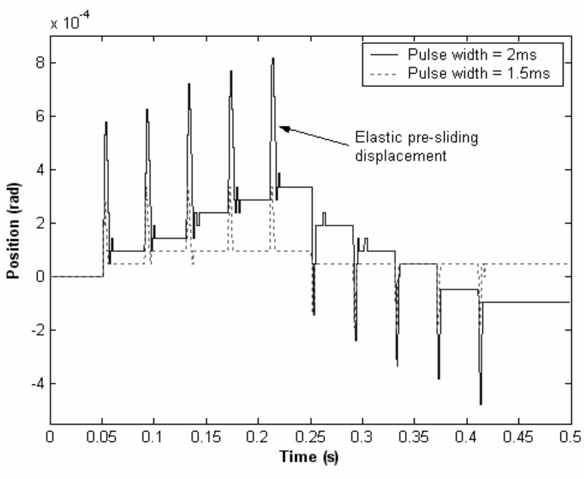


Figure 5: Experimentally measured displacement for both positive and negative impulses using successive pulse widths 1.5ms and 2ms.

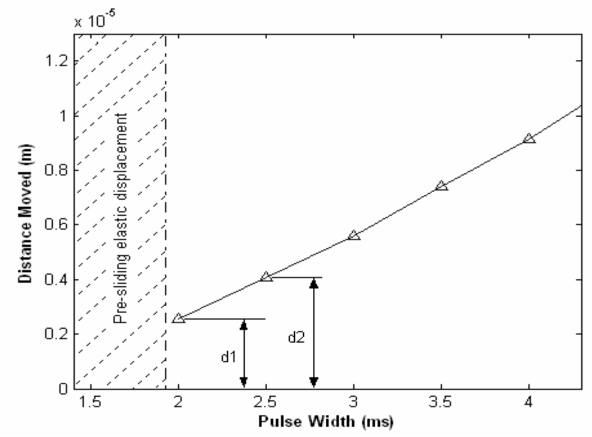


Figure 6: Simulated displacements as a function of pulse width.

mostly elastic pre-sliding movement which results in zero net displacement.

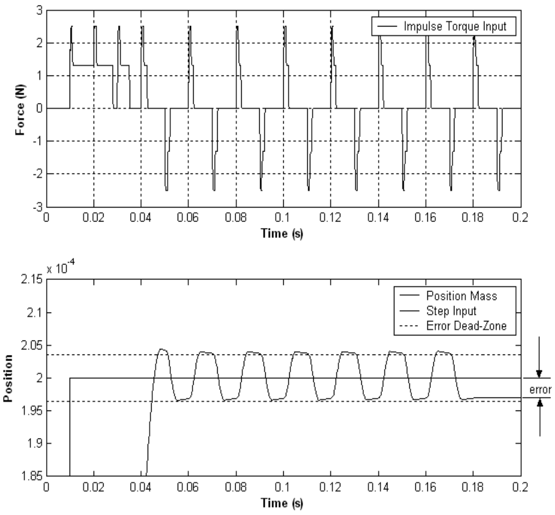
Wu et al (Wu et al, 2004) use the pre-sliding displacement as a means to increase the precision of the controller by switching the impulse controller off and using a continuous ramped driving torque to hold the system in the desired position. The torque is maintained even after the machine is at rest. This is difficult in practice as pre-sliding movement must be carefully controlled in the presence of varying static friction so that inadvertent breakaway followed by limit cycling is avoided.

3 LIMIT CYCLE OFFSET

3.1 Motivation

Figure 6 shows the simulated displacements of varying pulse widths which have been labelled *d1, d2, d3…dn* respectively, where *d1* is the minimum pulse width which will generate non elastic movement and defines the system's resolution.

Using the variable pulse width PID + impulse controller for a position pointing task, the torque will incrementally move the mechanism towards the reference set point in an attempt to reach steady state. Around the set point, the system will inevitably begin to limit cycle when the error *e(k)* is approximately the same magnitude as the system resolution (the displacement for the minimum pulse width *d1*).



Parameter	F_s	F_C	σ_0	σ_1	F_v	v_s	v_d
Value	2	1	$4.5*10^5$	12,000	0.4	0.001	0.0004

Figure 7: Simulation of the impulse controller limit cycling around a position reference set-point where the final torque output is a pulse with a minimum width and the mean peak to peak oscillation is *d1*. The friction parameters used for the simulation are also given in the accompanying table.

For the limit cycle to be extinguished, the controller must be disabled. As an example, the limit cycle in Figure 7 is extinguished by disabling the impulse controller at t=0.18s, and in this case, the resulting error is approximately half the displacement of the minimum pulse width *d1*.

Limit cycling will occur for all general servomechanisms using a torque pulse because every practical system inherently has a minimum pulse width that defines the system's resolution. Figure 7 simulates a typical limit cycle with a peak to peak oscillation equal to the displacement of the minimum pulse width *d1*.

One way to automatically extinguish the limit cycle is to include a dead-zone that disables the controller output when the error is between an upper and lower bound of the reference point (see Figure 7). The final error is then dependent on the amount of offset the limit cycle has in relation to the reference point. Figure 7 shows a unique case where the ± amplitude of the limit cycle is almost evenly distributed either side of the reference set point; i.e. the centre line of the oscillation lies along the reference set point. In this instance, disabling the controller would create an error *e(k)* equal to approximately $\left|\frac{\mathrm{d1}}{2}\right|$. This however, would vary in practice and the centreline is likely to be offset by some arbitrary amount. The maximum precision of the system will therefore be between *d1* and zero.

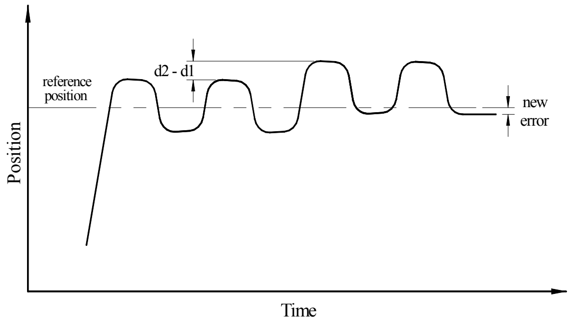


Figure 8: Conceptual example of reducing the steady state error using 'Limit Cycle Offset' with the limit cycle shifted up by *d2-d1* and the new error that is guaranteed to fall within the dead-zone.

3.2 Limit Cycle Offset Function

By controlling the offset of the limit cycle centreline, it is possible to guarantee that the final error lies within the dead-zone, and therefore to increase the precision of the system. As a conceptual example, Figure 8 shows a system limit cycling either side of the reference point by the minimum displacement *d1*. By applying the next smallest pulse *d2*, then followed by the smallest pulse *d1*, the limit cycle can be shifted by *d2 – d1*. The effect is that the peak to peak centreline of the oscillation has now been shifted away from the reference point.

However, at least one of the peaks of the oscillation has been shifted closer to the set point. If the controller is disabled when the mechanism is closest to the reference set point, a new reduced error is created. For this to be realised, the incremental difference in displacement between successively increasing pulses must be less than the displacement from the minimum pulse width; for example *d2 – d1 < d1*.

3.3 Modified Controller Design

For the limit cycle to be offset at the correct time, the impulse controller must have a set of additional control conditions which identify that a limit cycle has been initiated with the minimum width pulse. The controller then readjusts itself accordingly using a '*switching bound*' and finally disables itself when within a new specified error '*dead-zone'*. One way to achieve this is to adjust the pulse width so that it is increased by one increment when satisfying the following conditions:

$$\text{if} \quad \textit{switching bound} > |e(k)| \geq \textit{dead-zone}$$

$$\text{then} \quad \Delta = \frac{k_{pwm} \cdot e(k)\tau_s}{f_p} + 1$$

$$\text{otherwise} \quad \Delta = \frac{k_{pwm} \cdot e(k)\tau_s}{f_p} \tag{9}$$

where the switching bound is given by:

$$|\text{switching bound}| < \frac{\mathrm{d1}}{2} \tag{10}$$

and the dead-zone is given by:

$$|\text{dead-zone}| = \frac{(\mathrm{d2} - \mathrm{d1})}{2} \tag{11}$$

The steady state error *e(k)* becomes:

$$\left|\mathrm{e(k)}_{\text{steady state}}\right| \leq \frac{\text{deadzone}}{2} \tag{12}$$

3.4 Simulation of the Limit Cycle Offset Function

To demonstrate the limit cycle offset function, the modified controller is simulated using a simple unit mass with the ‘new’ friction model using Eqs. 1 to 4.

A simulated step response is shown in Figure 9 to demonstrate how the modified controller works. Here the mechanism moves towards the reference set point and begins limit cycling. Because at least one of the peaks of the limit cycle immediately lies within the switching bound, the controller shifts the peak to peak oscillation by *d2 - d1* by applying the next smallest pulse, and then followed by the smallest pulse. In this example, the first shift is insufficient to move either peak into the set dead-zone so the controller follows with a second shift. At time 0.1 seconds, the controller is disabled; however, the elastic nature of the friction model causes the mechanism’s position to move out of the dead-zone. As a result, the controller is reactivated (time 0.12s) and the controller follows with a third shift. In this instance, the mechanism reaches steady state at t=0.2s, and the final error is $|e(k)| \le \frac{1}{2}\cdot(\text{dead zone})$ which in this case is ± 1e-6 radians. A final analysis of the result shows that the new controller has reduced the error by an amount significantly more than a standard impulse controller. This reduction correlates directly to the improvement in the system’s accuracy by a factor of 4.

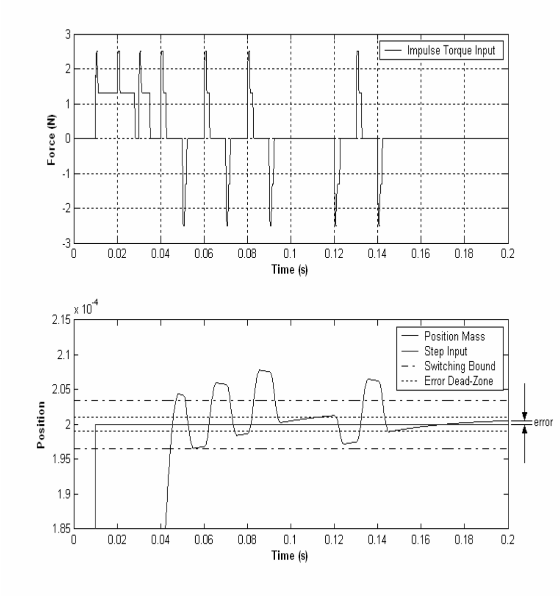


Figure 9: Simulation of the limit cycle offset function used with the PID + impulse controller.

4 EXPERIMENTAL

4.1 Position Pointing

This section evaluates the limit cycle offset function using the experimental Hirata robot having position dependent variables. Figure 10 shows a steady state limit cycle for a position pointing step response of 0.001 radians using a PID + impulse hybrid controller. The mean peak to peak displacement of the smallest non-elastic part of the limit cycle is μ_d.

The experiment was repeated using the limit cycle offset function with the same position step reference of 0.001 radians. Figure 11 shows a sample experiment and in this example, the limit cycle offset function is activated at t=0.9s. At this time, the amplitude of the non elastic part of the limit cycle is identified as lying between the switching bounds. The switching bounds and dead-zone are set according to the methodology given earlier. Once the offset function is activated, the controller adjusts itself by forcing the proceeding pulse to be one increment wider before returning to the smallest pulse width. This results in the limit cycle being shifted down into the dead-zone region where the impulse controller is automatically disabled at t=0.95s. At this time, the final error is guaranteed to fall within the error dead zone which can be seen from Fig 11 to be in the vicinity of ±1e-4 radians.

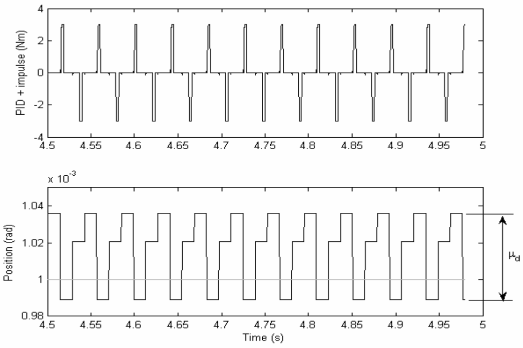


Figure 10: Steady state limit cycle for the PID + impulse hybrid controller when applying a unit step input to the Hirata robot. The mean peak to peak displacement μ_d is the non-elastic part of limit cycle.

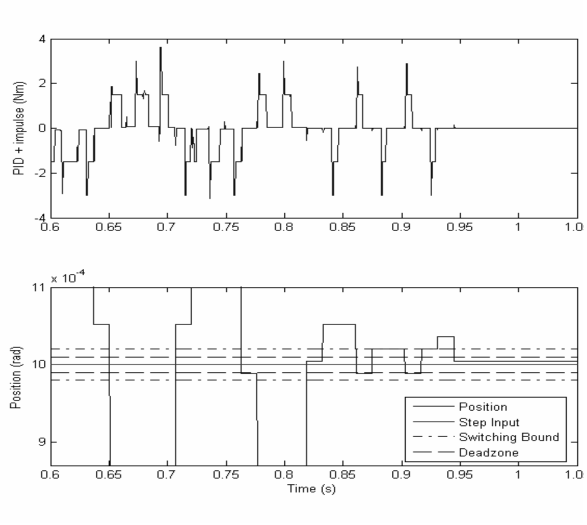


Figure 11: Using the 'Limit Cycle Offset' function to reduce the final steady state error of the Hirata robot.

4.2 Discussion of Results

This set of results demonstrates the Limit Cycle Offset function can be successfully applied to a commercial robot manipulator having characteristics of high non-linear friction. The results show that the unmodified controller will cause the robot to limit cycle near steady state position and that the peak to peak displacement is equal to the displacement of the smallest usable width pulse.

By using the Limit Cycle Offset function, the limit cycle can be detected and the pulse width adjusted so that at least one of the peaks of the limit cycle is moved towards the reference set point. Finally, the results show that the controller recognises the limit cycle as being shifted into a defined error dead-zone whereby the controller is disabled. The steady state error is therefore guaranteed to fall within a defined region so that the steady state error is reduced. For the SCARA robot, the improvement in accuracy demonstrated was 1.1e-4 radians in comparison to 4.5e-4 radians achieved without the limit cycle offset.

5 CONCLUSION

Advances in digital control have allowed the power electronics of servo amplifiers to be manipulated in a way that will improve a servomechanism precision without modification to the mechanical plant. This is particularly useful for systems having highly non-linear friction where conventional control schemes alone under perform. A previously developed hybrid PID + Impulse controller which does not require the mechanism to come to a complete stop between pulses has been modified to further improve accuracy. This modification shifts the limit cycling into a different position to provide substantial additional improvement in the mechanism's position accuracy. This improvement has been demonstrated both in simulations and in experimental results on a SCARA robot arm. The mechanism does not have to come to a complete stop between pulses, and no mechanical modification has to be made to the robot.

REFERENCES

Armstrong-Hélouvry, B., 1991, "*Control of Machines with Friction*" Kluwer Academic Publishers, 1991, Norwell MA.

Armstrong-Hélouvry, B., Dupont, P., and Canudas de Wit, C., 1994, "*A survey of models, analysis tools and compensation methods for the control of machines with friction*" *Automatica*, vol. 30(7), pp. 1083-1138.

Canudas de Wit, C., Olsson, H., Åström, K. J., 1995 "*A new model for control of systems with friction*" I*EEE Tansactions on Automatic Control*, vol. 40 (3), pp. 419-425.

Dahl, P., 1968, "*A solid friction model*" Aerospace Corp., El Segundo, CA, Tech. Rep. TOR-0158(3107-18)-1.

Dahl, P, 1977, "*Measurement of solid friction parameters of ball bearings*" Proc. of 6th annual Symp. on Incremental Motion, *Control Systems and Devices*, University of Illinois, ILO.

Hojjat, Y., and Higuchi, T., 1991 "*Application of electromagnetic impulsive force to precise positioning*" *Int J. Japan Soc. Precision Engineering*, vol. 25 (1), pp. 39-44.

Johannes, V. I.., Green, M.A., and Brockley,C.A., 1973, "*The role of the rate of application of the tangential force in determining the static friction coefficient*", *Wear*, vol. 24, pp. 381-385.

Johnson, K.L., 1987, "*Contact Mechanics*" Cambridge University Press, Cambridge.

Kato, S., Yamaguchi, K. and Matsubayashi, T., 1972, "*Some considerations of characteristics of static friction of machine tool slideway*" *J. o Lubrication Technology*, vol. 94 (3), pp. 234-247.

Li, Z, and Cook, C.D., 1998, "*A PID controller for Machines with Friction*" Proc. Pacific Conference on Manufacturing, Brisbane, Australia, 18-20 August, 1998, pp. 401-406.

Olsson, H., 1996, "*Control Systems with Friction*" Department of Automatic Control, Lund University, pp.46-48.

Popovic, M.R., Gorinevsky, D.M., Goldenberg, A.A., 2000, "*High precision positioning of a mechanism with non linear friction using a fuzzy logic pulse* *controlle*r" *IEEE Transactions on Control Systems Technology*, vol. 8 (1) pp. 151-158.

Rabinowicz, E., 1958, "*The intrinsic variables affecting the stick-slip process*," *Proc. Physical Society of London*, vol. 71 (4), pp.668-675.

Rathbun, D,. Berg, M. C., Buffinton, K. W., 2004, *"Piecewise-Linear-Gain Pulse Width Control for Precise Positioning of Structurally Flexible Systems Subject to Stiction and Coulomb Friction", ASME J .of Dynamic Systems, Measurement and Control, vol. 126, pp. 139-126.*

Richardson, R. S. H., and Nolle, H., 1976, "*Surface friction under time dependant loads*" *Wear*, vol. 37 (1), pp.87-101.

Wu, R,. Tung, P., 2004, *"Fast Positioning Control for Systems with Stick-Slip Friction"*, *ASME J .of Dynamic Systems, Measurement and Control*, vol. 126, pp. 614-627.

Yang, S., Tomizuka, M., 1988, "*Adaptive pulse width control for precise positioning under the influence of stiction and Coulomb friction*" *ASME J .of Dynamic Systems, Measurement and Control*, vol. 110 (3), pp. 221-227.

COLLABORATIVE CONTROL IN A HUMANOID DYNAMIC TASK

Diego Pardo and Cecilio Angulo
GREC - Knowledge Engineering Research Group
UPC - Technical University of Catalonia
Avda. Victor Balaguer s/n. Vilanova i la Geltrú, Spain
diego.pardo@upc.edu, cecilio.angulo@upc.edu

Keywords: Robot control architecture, Sensorimotor learning, Coordination policy, Reinforcement learning.

Abstract: This paper describes a collaborative control scheme that governs the dynamic behavior of an articulated mobile robot with several degrees of freedom (DOF) and redundancies. These types of robots need a high level of coordination between the motors performance to complete their motions. In the employed scheme, the actuators involved in a specific task share information, computing integrated control actions. The control functions are found using a stochastic reinforcement learning technique allowing the robot to automatically generate them based on experiences. This type of control is based on a modularization principle: complex overall behavior is the result of the interaction of individual simple components. Unlike the standard procedures, this approach is not meant to follow a trajectory generated by a planner, instead, the trajectory emerges as a consequence of the collaboration between joints movements while seeking the achievement of a goal. The learning of the sensorimotor coordination in a simulated humanoid is presented as a demonstration.

1 INTRODUCTION

Robots with several Degrees of Freedom (DOF) and redundant configurations are more frequently constructed; humanoids like Qrio (Kuroki et al., 2003), Asimo (Hirai et al., 1998) or HRP-2 (Kaneko et al., 2004), and entertainment robots like Aibo (Fujita and Kitano, 1998) are examples of it. Complex movements in complex robots are not easy to calculate, the participation of multiple joints and its synchronization requirements demands novel approaches that endow the robot with the ability of coordination. An attempt to drive the dynamics of its body optimally, measuring its performance in every possible configuration, would bring to a combinational explosion of its solution space.

The habitual use of simplified mathematical models to represent complex robotic systems, i.e., approximating non-linearities and uncertainties, would result in policies that execute approximately optimal control, thus, two major assumptions command the developed work. First, there is no pre-established mathematical model of the physics of the robot's body from which a control law could be computed; and second, the control design philosophy is focused on the action performance of the robot and not on the trajectory achievement by its joints; as an alternative, stochastic reinforcement learning techniques applied to numerical simulation models are studied. An optimal control problem, where a cost function is minimized to compute the policies, is stated.

The goal of this work is to solve a multi-joint robot motion problem where coordination is need. Planners and trajectory generators are usually in charge of complex motions where many joints are involved and the mechanical stabilization of the robot is in risk; the relationship between sensory signals and motor commands at the level of dynamics is viewed as a low level brick in the building of control hierarchy. Here we use a dynamic control scheme that also solves the trajectory problem.

It is important to mention that a previous work uses coordination at the level of dynamics to extend a manipulator robot capacity (Rosenstein and Barto, 2001), their objective was to use a biologically inspired approach to profit synergical movements between joints, like human muscles relationship. They use a hand-made pre-established PID controllers in every joint of the robot and the final applied torque in each motor is the linear combination of the PID's output, then collaboration is shown. The combination parameters are obtained by direct search methods (Rosenstein, 2003). By using linear controllers to solve a nonlinear problem, they implement a hierarchical motor program that runs various feedback controllers, i.e, they switch between PID's (parameters and goal) in the middle of the motion.

Here we extend that result by proposing a more general scheme, where sensor information is processed in independent and specific layers to produce coordinate control actions. Furthermore, at the architecture definition, the control functions are no restricted to PID's and its linear combination. Additionally, we use reinforcement learning to compute all the architecture controllers.

This paper is organized as follows Section 2 is devoted to the presentation of the architecture and a linear implementation of practical use is outlined in 3; while section 4 validates it through a simulation experiment of a robot equilibrium dynamical task. Section 6 gathers the conclusions and points to the future work.

2 LAYERED COOPERATIVE CONTROL

The Layered Cooperative Control (LCC) scheme is meant to generate motions while solving a *Dynamical Task* (DT). It is assumed that the final configuration of the DT is known, i.e. the set point of the joints angles is preestablished; but how to coordinate motions to reach this configuration, restricted to its body dynamics, is unknown. It is also assumed that the states of the system are observable and that each joint position is commanded by independent servos.

Figure 1 represents schematically the main idea of the LCC. The control actions that drive the motors are processed by two layers of controller functions. Each component of the first layer (D) manipulates the dynamics of a singular joint; the output of this layer intends to position its corresponding joint in a previously specified angle, but it is filtered by the second layer of controllers (f). Each motor is driven by a different function, which calculates the control action based on all the signals originated in the previous layer.

The controllers of the first layer are related with the dynamics of the link, they must be established once. The second layer of functions (f) must be designed facing the DT intended to be completed. Functionally and computationally speaking the architecture is layered; every time step two mappings take place, one for error position reasons and the other seeking coordination between joints.

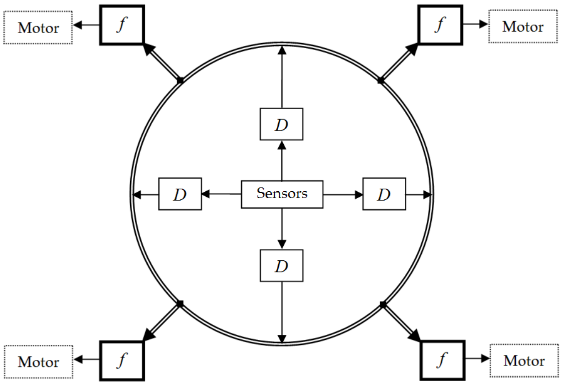


Figure 1: LCC Basic Idea : Two layers of control.

Every controller of the second layer (f) uses the information originated in all the first layer's controllers. This collaboration allows joint controllers to know the dynamical state of the others, and then to act according to it in order to complete the DT.

2.1 Architecture Formulation

Let the articulated mobile robot have n joints. Each joint has an error based controller D_i with $(i = 1, ..., n)$ in the first layer. This is a SISO function, whose input is the error of the position of its respective joint (e_i) and whose output is a control action (u_i). The error is calculated using the sensor information providing the actual position of the joint (θ_i) and the already known set point or goal configuration (θ_i^*).

$$\begin{aligned} e_i &= \theta_i - \theta_i^* \\ u_i &= D_i(e_i) \qquad i = 1, ..., n \end{aligned}$$

Let the dynamical task have associated n controllers f_i in the second layer with $(i = 1, ..., n)$, these are MISO functions whose inputs are the n outputs u_i of the first layer described above. The output of the second layer functions v_i is the velocity applied on the corresponding motor (M_i).

$$v_i = f_i(u_1, ..., u_n) \qquad i = 1, ..., n$$

The robot state is represented by a continuous time dynamical system controlled by a parameterized policy,

$$\begin{aligned} \dot{x} &= g(x, v) + h(x) \cdot v & (1) \\ v &= \pi_w(x, v) & (2) \end{aligned}$$

where vector x represents the states of the robot, g and h its dynamics, v the control action and π the control policy parameterized by the vector w.

2.2 LCC Synthesis

Once the control scheme has been stated, the next step is the definition of a methodology to compute the functions (f_i, D_i). Here a policy gradient reinforcement learning (PGRL) algorithm is employed. PGRL methods (see (Williams, 1992),(Sutton et al., 2000)) are based on the measurement of the performance of a parameterized policy π_w applied as control function during a delimited amount of time. In order to measure the performance of the controller, the following function is defined,

$$V(w) := J(x, \pi_w(x)) \tag{3}$$

where the measurement of the performance V of the parameters w is done by defining the *cost function* J.

By restricting the scope of the policy to certain class of parameterized functions $u = \pi_w(x)$, the performance measure (3) is a surface where the maximum value corresponds to the optimal set of parameters $w \in \mathbf{R}^d$. The search for the maximum can be performed by standard gradient ascent techniques,

$$w_{k+1} = w_k + \eta \nabla_w V(w) \tag{4}$$

where η is the step size and $\nabla_w V(w)$ is the gradient of $V(w)$ with respect to w. The analytical formulation of this gradient is not possible without the acquisition of a mathematical model of the robot. The numerical computation is also not evident, then, a stochastic approximation algorithm is employed : the 'weight perturbation' (Jabri and Flower, 1992), which estimates the unknown gradient using a Gaussian random vector to orientate the change in the vector parameters. This algorithm is selected due to its good performance, easy derivation and fast implementation; note that the focus of this research is not the choice of a specific algorithm, nor the development of one, but rather the cognitive architecture to provide robots with the coordination learning ability.

This algorithm uses the fact that, by adding to w a small Gaussian random term z with $E\{z_i\} = 0$ and $E\{z_i z_j\} = \sigma^2 \delta_{ij}$, the following expression is a sample of the desired gradient

$$\alpha(w) = [J(w+z) - J(w)] \cdot z \tag{5}$$

Then, both layers' controllers can be found using this PGRL algorithm.

$$\begin{aligned} u_i &= D_{k_i}(e_i) \\ v_i &= f_{w_i}(u_1, ..., u_n) \end{aligned} \tag{6}$$

In order to be consequent with the proposed definition of each layer, the training of the vector parameters must be processed independently; first the dynamical layer and then the coordination one. It is assumed that the movement has to take place between a limited amount of time T, where signals are collected to compute the cost function (3), then, the gradient is estimated using (5) and a new set of parameters obtained applying the update strategy of (4). Notice that in the case of the first layer, each function D_{k_i} is trained and updated separately from the others of the same layer, but when learning the coordinate layer, all the functions f_{w_i} must be updated at the same time, because in this case the performance being measured is that of the whole layer.

3 LINEAR IMPLEMENTATION

The previous description of the LCC is too general to be of practical use. Then, it is necessary to make some assumptions about the type of functions to be used as controllers. A well known structure to control the dynamic position of one link is the Proportional Integral Derivative (PID) error compensator. It has the following form,

$$u_i = K_{P_i} \cdot e_i + K_{D_i} \cdot \frac{de_i}{dt} + K_{I_i} \cdot \int e_i \, dt \tag{7}$$

The functionality of its three terms (K_P, K_D, K_I) offers management for both transient and steady-state responses, therefore, it is a generic and efficient solution to real world control problems. By the use of this structure, a link between optimal control and PID compensation is revealed to robotic applications. Other examples of optimization based techniques for tuning a PID are (Daley and Liu, 1999; Koszalka et al., 2006).

The purpose of the second layer is to compute the actual velocity to be applied on each motor by gathering information about the state of the robot while processing a DT. The PID output signals are collected and filtered by this function to coordinate them. Perhaps the simplest structure to manage this coordination is a gain row vector W_i, letting the functions in the second layer be the following,

$$f_{w_i}(\mathbf{u}) = \mathbf{W}_i \cdot \mathbf{u}$$

Then, a linear combination of $\mathbf{u}$ commands the coordination. The matrix W_{DT_m} encapsules all the information about this layer for the mth DT.

$$\mathbf{W}_{DT_m} = \begin{bmatrix} w_{11} & \dots & w_{in} \\ \vdots & \ddots & \vdots \\ w_{n1} & \dots & w_{nn} \end{bmatrix} \tag{8}$$

Where the term w_{ij} is the specific weight of the jth PID in the velocity computation of the joint i.

4 HUMANOID COORDINATION

Classical humanoid robot motions are typically very slow in order to maintain stability; e.g., the biped walking based on Zero Moment Point (ZMP) (Vukobratovic and Stepanenko, 1972) condition avoids those states far from the pre-determined trajectory in which the momentum is guaranteed. By doing this, thousands of trajectories and capabilities of the robot could be restricted. Highly dynamic capabilities need those avoided states to perform quick and realistic motions.

4.1 The Simulation Robot Environment

A simulated humanoid is employed as a test-bed to evolve the controllers. It was modeled using Webots (Michel, 2004), a 3D robot simulation platform . Details can be obtained in (Webots,).

This particular model of the full body humanoid robot has a total of 25 DOF, each link torque is limited to $[-10, 10]$ Nm, additionally it has a camera, a distance sensor, a gps, 2 touch sensors in the feet, and a led in the forehead. The model includes physical characteristics; the software processes kinematic and dynamic simulation of the interaction between the robot and its environment.

4.2 The One-Leg Equilibrium Task

The goal of the simulation experiment is to reach a final configuration called *'one-leg equilibrium point'* starting from the passive stand up position. Three joints are involved in this motion (see Figure 2(a).): The longitudinal axis of the Back, the transversal axis of the left hip and the left knee; their goal states, i.e. angles in radians, are $\theta_1^* = 0.4, \theta_2^* = 1.5$ and $\theta_3^* = 1.5$ respectively, with final velocity $\dot{\theta}^* = 0$ for all the joints. Both configurations, initial and final, are shown in Figure 2(b). The goal configuration has been designed to generate a slight inclination of the back to the right side, changing the center of mass of the body of the robot, thus, letting more weight supported on the right leg, and then allowing the left leg to lift, by means of the blending of the hip and knee articulations.

Figure 3 shows the simulation sequence of the motion of the robot using standard low level controllers. The robot falls down when attempts to reach the goal configuration due to the forces generated by the friction between the floor and its foot soles. The whole body of the robot suffers a destabilization at the start of the movement.

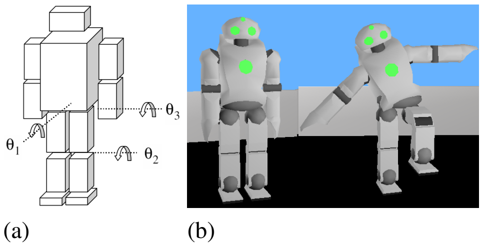


Figure 2: Simulated Humanoid (a). Joints involved in the motion. (b) Initial and goal configuration.

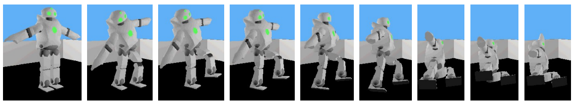


Figure 3: Direct Solution: Failure Demonstration.

4.3 The Learning Procedure

Table 1 shows the PGRL algorithm. A supervisor program is in charge of the training; its EVALUATE subroutine delivers the set of values to be evaluated in the robot controllers. It starts a controlled episode from the initial configuration and intends to achieve the goal states; every evaluation takes $T = 12s$, within this time the cost function is measured and returned. The supervisor repeats this operation until a convergence criterium is matched or the number of iterations is too large.

In this experiment the PID values are learned but not calculated. The following is the discrete implementa-

Table 1: Weight perturbation PGRL Algorithm.

```
input
  step size $\eta \in [0, 1]$
  search size $\sigma \geq 0$
  max step $\Delta_{max} \in [0, 1]$
initialize
  $W_0 \leftarrow \mathbb{I} \in \mathbf{R}^{3\times 3}$ (Identity Matrix)
  $\alpha \leftarrow 0$
  $J \leftarrow$ EVALUATE($\mathbf{W}$)
repeat
  1. $z \leftarrow N(0, \sigma)$
  3. $J_z \leftarrow$ EVALUATE($\mathbf{W+z}$)
  4. $\alpha \leftarrow [J_z - J] \cdot z$
  5. $\Delta W \leftarrow -\eta\alpha$
  6. $\Delta W \leftarrow \min(\Delta W, \Delta_{max})$
  7. $W \leftarrow W + \Delta W$
  8. $J \leftarrow$EVALUATE($\mathbf{W}$)
until convergence or number of iterations too large
return $W, J$
```

tion of the PID employed,

$$u_i^t = K_P \cdot e_i^t + K_D \cdot (e_i^t - e_i^{t-1}) + K_I \cdot \sum_{j=0}^{t} e_i^j \qquad (9)$$

Where u_i^t represents the control signal associated with joint i at step time t, it depends on its corresponding position error $e_i^t = \theta_i^* - \theta_i^t$. This controller must provide zero position error in a single joint motion, therefore its parameters are learned using as performance criteria the following expression,

$$J_i(V) = -\sum_{t=0}^{T} (\theta_i^* - \theta_i^t)^2 + \sum_{t=0}^{T} (\dot{\theta}_i^t)^2 \qquad (10)$$

The PID is tested using a time step of $\Delta T = 32ms$ during $T = 12s$. The algorithms constants used are $\sigma = 0.032$ and $\eta = 0.05$. For the learning of the particular PID in charge of controlling the back, special restrictions are implemented to the reward function, penalizing those episodes where the robot falls down. The penalization is proportional to the time the robot stays on the ground.

The linear parameterized function selected for the coordination layer is:

$$\mathbf{v_t} = \mathbf{W} \cdot \mathbf{u_t} \qquad (11)$$

where $v_t \in \Re^3$ are the actual final velocities; $u_t \in \Re^3$ the outputs of the PIDs; and $W \in \Re^{3\times3}$ is the transformation that contains the parameters to be found (W_{ij}).

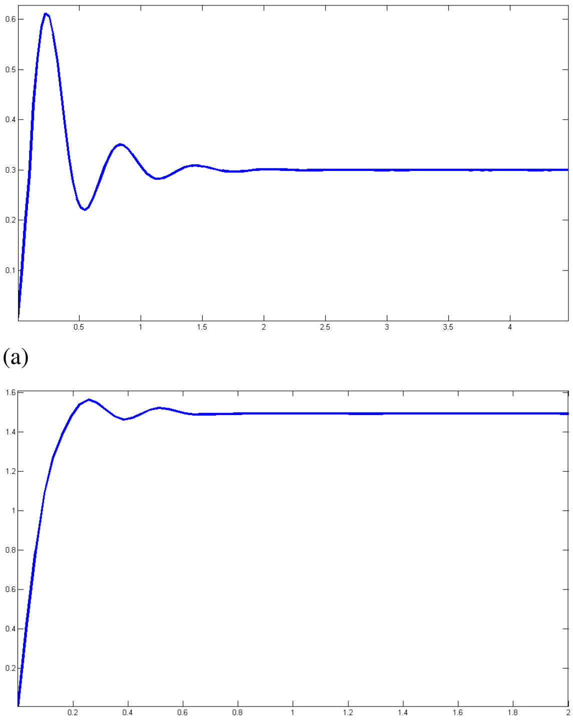
The algorithm starts with a diagonal matrix as coordination controller, i.e. $W_0 = I$, meaning that the first collaborative control function attempted is that in which just the original PID controllers act. The non-diagonal values of K deliver coordination. Same algorithms constants and time step are used in this stage of the LCC implementation.

5 RESULTS

5.1 PID Controller

Several local minima are found. Figure 4(a) shows the output of the back joint using one of the PID solutions found. It presents big overshoot, but no stationary error to a set point of $\theta^* = 0.3$, and mainly: The robot learns how 'not to fall'.

The behavior of the Back joint is equivalent with an inverted pendulum system with a big mass being manipulated (robot torso). For this joint, and for a better controlled output, a more complex parameterized policy would be needed. On the other hand, Figure 4(b) presents the output of the hip joint. It is evident that a better solution is achieved, this is because the training of this PID is done in a lay down position, with no risk of falling, and the mass of the link being controlled is widely insignificant compared with that in the back-joint case.



(a)

(b)

Figure 4: **(a)** Back-Joint position output (radians), using a learned PID (set point=0.3rad), **(b)** Hip-Joint position output (radians) using a learned PID (set point=1.5rad).

Final values for the independent joints controllers (Back, Hip and Knee) are :

$$\begin{aligned} PID_1 &= [0.746946 \;\; 0.109505 \;\; 0.0761541] \\ PID_2 &= [0.76541 \;\; 0.0300045 \;\; 0] \\ PID_3 &= [0.182792 \;\; 0.0114312 \;\; 0.00324704] \end{aligned}$$

5.2 Coordination Layer

After several iterations, the best solution is found; a sequence of the humanoid learned motion is depicted in Figure 5. The associate coordination controller is the following:

$$W = \begin{bmatrix} 0.6612900 & -0.3643360 & -0.1936260 \\ -0.0497653 & 1.0266600 & -0.0684569 \\ 0.0285586 & 0.1469820 & 0.9642390 \end{bmatrix}$$

Note how the non-diagonal parameters have grown, allowing the coordination. This characteristic has

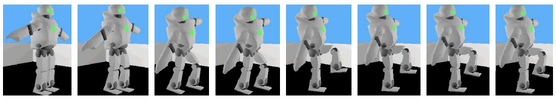
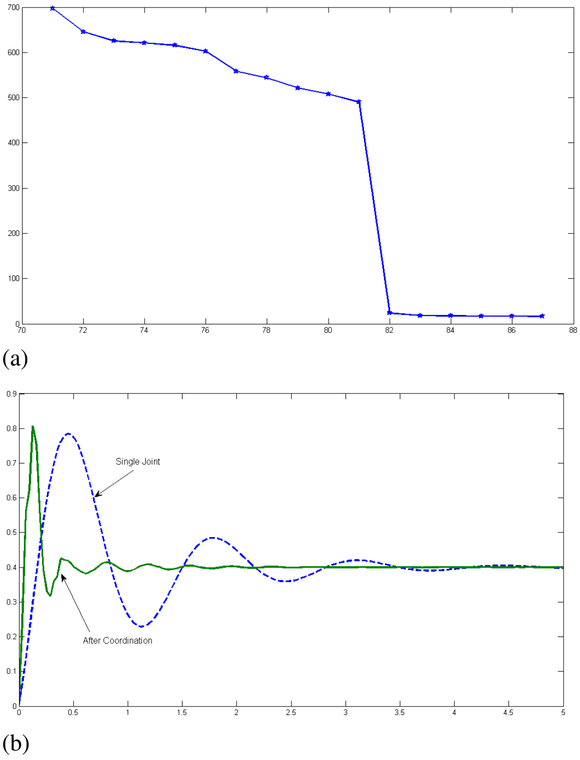


Figure 5: Simulated Humanoid One-Leg Equilibrium task.



(a)

(b)

Figure 6: **(a)** Reward Evolution $J(W)$ of the best learned solution (last 20 tirals); **(b)** Back-joint position, before (dotted line) and after the coordination learning.

been previously pointed out in (Rosenstein and Barto, 2001), but here in an unstable 3D system task.

For this particular solution, the evolution of the reward is presented in Figure 6(a). Once the robot learns how 'not to fall' the convergence rate increases and the final results get to lower values.

It is important to mention that the robot presents some absolute displacement, tending to go backwards. A better designed reward function is needed to avoid absolute displacements, but a good solution is harder to find in this scenario. Finally, it is important to emphasize the change in the behavior of the back joint after the second layer phase. It seems like the system learns that the overshoot must be shorter in time in order to be able to lift the leg with out falling down. Figure 6(b). shows the output of the back joint before and after the second layer of learning.

6 CONCLUSIONS

The LLC scheme proposes to use the dynamic interaction of the robots articulations as a trajectory generator, without the use of a trajectory planner. Exploiting dynamics gives the robot the ability to explore motion interactions that result in optimized behaviors: *Learning at the level of dynamics to succeed in coordination*.

The presented solution overcomes the model dependency pointed above; by the presentation of a systematic control scheme the ideas of (Rosenstein and Barto, 2001) are extended. Here, the low level controllers are found using learning techniques and the formulation of a control architecture allows the implementation of different parameterized policies.

The interaction between Machine Learning, Control Systems and Robotics creates an alternative for the generation of artificial systems that consistently demonstrate some level of cognitive performance.

Currently, highly dynamic motions in humanoids are widely unexplored. The capability of this robots will be extended if motions that compromise the whole-body dynamic are explored. The LCC is tested within a simulated humanoid and succeed in the performance of a very fast motion, one in which the whole-body equilibrium is at risk.

Coordination is possible thanks to the sharing of information.

ACKNOWLEDGEMENTS

This work has been partly supported by the Spanish MEC project ADA (DPI2006-15630-C02-01). Authors would like to thank Olivier Michel and Ricardo Tellez for their support and big help.

REFERENCES

Daley, S. and Liu, G. (1999). Optimal pid tuning using direct search algorithms. *Computing and Control Engineering Journal*, 10(2):251–56.

Fujita, M. and Kitano, H. (1998). Development of an autonomous quadruped robot for robot entertainment. *Autonomous Robots*, 5(1):7–18.

Hirai, K., Hirose, M., Haikawa, Y., and Takenaka, T. (1998). The development of honda humanoid robot. In *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA*.

Jabri, M. and Flower, B. (1992). Weight perturbation: An optimal architecture and learning technique for analog VLSI feedforward and recurrent multilayer networks.

IEEE Transactions on Neural Networks, 3(1):154–157.

Kaneko, K., Kanehiro, F., Kajita, S., Hirukawa, H., Kawasaki, T., Hirata, M., Akachi, K., and Isozumi, T. (2004). Humanoid robot hrp-2. In *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA*.

Koszalka, L., Rudek, R., and Pozniak-Koszalka, I. (2006). An idea for using reinforcement learning in adaptive control systems. In *Proceedings of the IEEE Int Conference on Networking, Systems and Mobile Communications and Learning Technologies*, Kerkrade, Netherlands.

Kuroki, Y., Blank, B., Mikami, T., Mayeux, P., Miyamoto, A., Playter, R., Nagasaya, K., Raibert, M., Nagano, M., and Yamaguchi, J. (2003). A motion creating system for a small biped entretainment robot. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems, IROS*.

Michel, O. (2004). Webots: Professional mobile robot simulation. *Journal of Advanced Robotics Systems*, 1(1):39–42.

Rosenstein, M. T. (2003). *Learning to exploit dynamics for robot motor coordination*. PhD thesis, University of Massachusetts, Amherst.

Rosenstein, M. T. and Barto, A. G. (2001). Robot weightlifting by direct policy search. In *Proceedings of the IEEE International Conference on Artificial Intelligence, IJCAI*, pages 839–846.

Sutton, R., McAllester, D., Singh, S., and Mansour, Y. (2000). Policy gradient methods for reinforcement learning with function approximation. *Advances in Neural Information Processing Systems*, 12:1057–1063.

Vukobratovic, M. and Stepanenko, J. (1972). On the stability of anthropomorphic systems. *Mathematical Biosciences*, 15:1–37.

Webots. http://www.cyberbotics.com. Commercial Mobile Robot Simulation Software.

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256.

A FUZZY SYSTEM FOR INTEREST VISUAL DETECTION BASED ON SUPPORT VECTOR MACHINE

Eugenio Aguirre, Miguel García-Silvente, Rui Paúl
Department of Computer Science and A.I., E.T.S. Ingeniería Informática
University of Granada, 18071 Granada, Spain
{eaguirre, M.Garcia-Silvente, ruipaul}@decsai.ugr.es

Rafael Muñoz-Salinas
Deparment of Computing and Numerical Analysis, E.P.S.
University of Cordoba, Cordoba, Spain
rmsalinas@uco.es

Keywords: Human-Robot Interaction, Interest Detection, Head Pose Estimation, Fuzzy Logic, Support Vector Machine.

Abstract: Despite of the advances achieved in the past years in order to design more natural interfaces between intelligent systems and humans, there is still a great effort to be done. Considering a robot as an intelligent system, determining the interest of the surrounding people in interacting with it is an interesting ability to achieve. That information can be used to establish a more natural communication with humans as well as to design more sophisticated policies for resource assignment. This paper proposes a fuzzy system that establishes a level of possibility about the degree of interest that people around the robot have in interacting with it. First, a method to detect and track persons using stereo vision is briefly explained. Once the visible people is spotted, their interest in interacting with the robot is computed by analyzing its position and its level of attention towards the robot. These pieces of information are combined using fuzzy logic. The level of attention of a person is calculated by analyzing the pose of his head that is estimated in real-time by a view based approach using Support Vector Machines (SVM). Although the proposed system is based only on visual information, its modularity and the use of fuzzy logic make it easier to incorporate in the future other sources of information to estimate with higher precision the interest of people. At the end of the paper, some experiments are shown that validate the proposal and future work is addressed.

1 INTRODUCTION

The interaction between Intelligent Systems and human beings is a topic that is focusing a great research effort nowadays. The development of natural and multimodal interfaces is needed to enhance the interaction abilities of current Intelligent Systems. In particular, within the area of Robotics, the development of successful robotic systems applied to service tasks in home and office environments implies the generation of natural human-robot interfaces. In that sense, important issues that must be taken into account are how robots can detect the presence of persons around them and how do they recognize when and how long a person is interested in establishing an interaction. In order to achieve this goal, it is necessary to solve several problems. First, a robot must be able to detect persons in its vicinity and track their movements over time. People tracking is not an easy task since several persons could be moving at the same time, crossing their trajectories and occluding each others. We can find many works in the literature on this topic (Fritsch et al., 2003; Snidaro et al., 2005). The techniques to perform the detection and tracking are frequently based on the integration of different information sources such as: skin color, face detectors, visual analysis of the motion or laser range finder.

Once a robot is able to recognize and track the persons in its vicinity, it should be able to detect their interest in establishing an interaction with it. In that task, several types of signals from the human can be taken into account (both verbal and non-verbal). Some authors (Bennewitz et al., 2005) use sound source localization or speech recognition besides visual perception to detect which persons are the most interested. In other cases, facial expressions (Song et al., 2001) or hand gestures (Ghidary et al., 2002) are analyzed. Finally, other authors (Kulic and Croft, 2003) propose the use of non-verbal signals present in physiological monitoring systems that include skin conductance, heart rate, pupil dilation and brain and muscle neural activity.

In regards to the role of the fuzzy logic in robotics, an extensive catalogue of the uses of fuzzy logic in autonomous robots can be found in (Saffiotti, 1997). Fuzzy logic has been successfully applied to a multitude of problems such as: design of controlling behaviors for navigation, behavior coordination, map building, integration of deliberative and reactive layers, etc (Aguirre and González, 2000). Lately, fuzzy logic has also been applied to the area of human-robot interaction. In (Bien and Song, 2003) several soft computing techniques are applied to service robotic systems for comfortable interaction and safe operation. Fuzzy logic is used for recognizing facial emotional expression and for coordinating bio-signals with robotic motions. In (Kulic and Croft, 2003) several sets of fuzzy rules are used for estimating intent based on physiological signals.

In this work we are interested in computing a value of possibility of the interest of a person to interact with the robot. This value is computed using only visual information, but the modularity of the system makes easy the posterior incorporation of other types of input data as sound or laser range finder. The interest is computed according to the position of the person and its degree of attention. In a first step, people detection and tracking problems are solved by a stereoscopic system. The use of stereo vision brings several advantages when developing human-robot applications. On the one hand, the information regarding disparities becomes more invariable to illumination changes than the images provided by a single camera, being a very advantageous factor for the background estimation (Darrell et al., 2001). Furthermore, the possibility to know the distance to the person could be of great assistance for the tracking as well as for a better analysis of their gestures.

Once the surrounding people is spotted, we propose a new method for estimating the interest of the detected people in interacting with the robot by means of computer vision and fuzzy logic. The person's attention is detected by the analysis of the pose of his head. To detect the head pose we have employed a view based approach using Support Vector Machines (SVM) (Cristianini and Shawe-Taylor, 2000) that let us classify the head pose in real time.

The approach presented in this work is not only valid for robotic applications. It can also be employed in intelligent systems that use stereoscopic devices. For example, it can be applied in intelligent spaces where one or several intelligent devices wish to interact with people according the interest shown by each person.

The remainder of this paper is structured as follows. Section 2 gives an general overview of the hardware and software system, describing the method employed for the detection and tracking of people in the surroundings of the robot. In Section 3 it is explained the SVM based approach to estimate the head pose and the fuzzy system for estimating the interest of people. In Section 4 it is shown the experimentation carried out, and finally, Section 5 outlines some conclusions and future works.

2 PEOPLE DETECTION AND TRACKING

The hardware system is comprised by a laptop to process the information, a stereoscopic system with a binocular camera (PtGrey, 2005) and a Nomad 200 mobile robot (see Fig. 1).

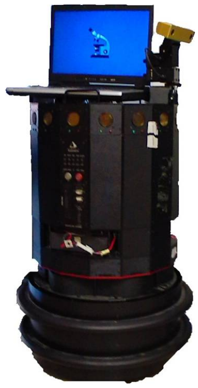


Figure 1: Robot with stereo vision system.

The ability of detecting and tracking people is fundamental in robotic systems when it is desirable to achieve a natural human-robot interaction. They are achieved in our architecture by combining stereo vision and color using plan-view maps. Following, the process for people detection and tracking is explained in a summarized way. The readers more interested in this process are referred to (Muñoz-Salinas et al., 2006).

Our robot has a stereo camera that is mounted on a pan-tilt unit (PTU). The stereo camera captures two images from slightly different positions (calibrated stereo pair) that are transferred to the computer to calculate a *disparity image* containing the points matched in both images. Knowing the intrinsic parameters of the stereo camera it is possible to reconstruct the three-dimensional position p_{cam} of a matched pixel (u,v). Then, the points captured are translated to a *"robot"* reference system, placed at the center of the robot at ground level in the direction of the heading of the robot. Generally, the number of points captured by our stereo system is very high. In order to perform a reduction of the amount of information, the points captured by the camera are orthogonally projected into a 2D plan-view map O named *occupancy map* (Harville, 2004; Haritaoglu et al., 2002; Hayashi et al., 2004).

The next step in our processing, is to identify the different objects present in O that could correspond to human beings (*human-like objects*). For that purpose, O is processed with a closing operator in order to link possible discontinuities in the objects caused by the errors in the stereo calculation. Then, objects are detected as groups of connected cells. Those objects whose area are similar to the area of a human being and whose sum of cells (occupancy level of the object) is above a threshold θ_{occ} are considered human-like objects. This test is performed in a flexible way so that it is possible to deal with the stereo errors and partial occlusions. However, the human-like objects detected might not belong to real people but to elements of the environment. The approach employed in this work to detect a person consists in detecting if any of the human-like objects found in O show a face in the camera image.

Face detection is a process that can be time consuming if applied on the entire image, thus, it is only applied on regions of the camera image where the head of each object should be (head region). As the human head has a typical average width and height, the system analyzes first if the upper part of a human-like object has similar dimensions. If the object does not pass this test, the face detector is not applied to it. This test is performed in a flexible manner so that it can handle stereo errors and people with different morphological characteristics can pass it. If the human-like object passes this test, the corresponding region in the image is analyzed to detect if it contains a face. The face detector employed is based on the face detector of Viola and Jones (Viola and Jones, 2001) which was later improved by Lienhart (Lienhart and Maydt, 2002). We have employed the OpenCv's Library (Intel, 2005) implementation that is trained to detect frontal human faces and works on gray level images.

Once a face has been detected on a human-like object, a color model of the person torso is created (Comaniciu et al., 2000). The idea is to assist the tracking process by capturing information about the color of the clothes of the user so that the robot can distinguish him/her from other people in the environment. Therefore, pixels around what it should be the chest of the person are used. The position of the chest in the camera image is estimated as 40 cm below the top of the head region. The size of the region used to create the color model depends on the distance of the person from the camera. When the object is far from the camera the region used is smaller to avoid including pixels from the background and it becomes bigger when the object is near to the camera.

Tracking consists in detecting in subsequent frames the human-like object that corresponds to the person being tracked. The Kuhn's well-known Hungarian Method for solving optimal assignment problems (Kuhn, 1955) is employed for that purpose. Two pieces of information are combined (position and color) to assign a value to each human-like object indicating its likelihood to be the person being tracked. On one hand, a prediction of the future position of the person being tracked is calculated using the Kalman filter. The nearer a human-like object is from the position estimated for the person being tracked, the higher likelihood it will have to be him/her. On the other hand, color information is employed to achieve a more robust tracking. The more similar the color of a human-like object is to the clothes' color of the person being tracked, the higher likelihood it will have to be him/her. Both likelihood are combined so that when the person being tracked is near others, color information can help to distinguish him/her. The human-like object with highest likelihood is considered to be the person being tracked if its likelihood value exceeds a certain threshold. In that case, the Kalman filter is updated with the new observations and also the color model of the person is updated so that it can adapt to the illumination changes that take place.

When the position of the person being tracked is located, the system determines the location of his head in the camera image. In this work, the head is modeled as an ellipse whose size in the camera image is determined according to the distance of the person to the camera. Firstly, the system calculates an initial estimation of the head position in the camera image based on stereo information. Then, the initial position is refined by a local search process. For that purpose, the gradient around the ellipse perimeter is examined in order to determine the likelihood of a position using the Birchfield's method (Birchfield, 1998). The position with higher likelihood is considered the person's head position.

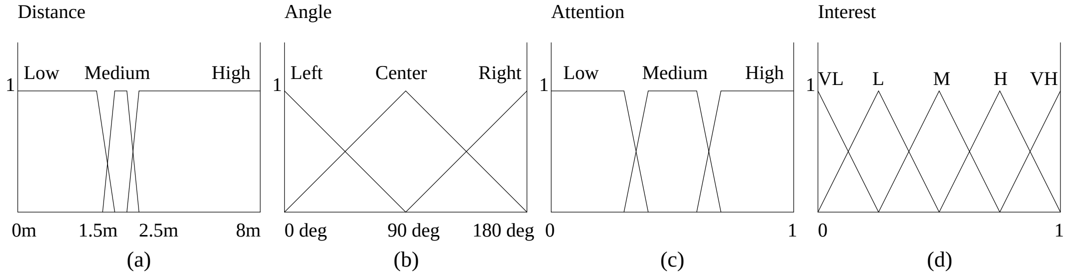


Figure 2: Fuzzy sets of the linguistic variables: (a) Distance (b) Angle (c) Attention (d) Interest.

3 INTEREST DETECTION

In the previous section, we have described as the robot is able to detect and track them the persons in its vicinity by using the stereo system. This section explains our approach for estimating the interest of the detected people in interacting with the robot by means of fuzzy logic. The approach presented in this work is based on stereo vision but the system can be easily expanded to merge other sources of information. The advantages of using fuzzy logic are mainly three. Firstly, the robot has to deal with information from the stereo system that is affected by uncertainty and vagueness. Fuzzy logic is a good tool to manage these factors using linguistic variables. Secondly, the human knowledge can be usually expressed as rules. Fuzzy logic allows to establish relationships among the variables of a problem through fuzzy rules providing an inference mechanism. Finally, there are methods in fuzzy logic to fuse the results from several fuzzy rules in order to achieve a final overall result. Therefore, the system designed in this work, based exclusively in stereo information, can be integrated with other fuzzy systems using other types of information as source sound localization, gesture analysis or speech recognition systems.

In this work, the determination of the degree of interest of a person is based on its position and its degree of attention. The position of a person is analyzed using both its distance to the center of the robot and its angle in respect to the heading direction of the robot. The first feature is measured by the linguistic variable $Distance$ and the second one by the linguistic variable $Angle$. These linguistic variables have three possible values each of them, that are shown in Fig. 2. The meaning of these two variables is the following: if the person is detected near to the robot and more or less centered with respect to it, then we consider that the person is more interested in establishing interaction with the robot than when the person is far or at the left or right side of the robot. Nevertheless, the position of the person is not enough to determine his interest in interacting with the robot. Thus, the third feature shown in this paper is the person's attention detected by the analysis of the pose of his head. To detect the head pose we have employed a view based approach using SVM that is explained in the next section.

3.1 Estimating Face Attention using SVM

One of the most prominent cues to detect if a person is paying attention to the system is the orientation of his face, i.e., a higher degree of attention can be assumed when a person is looking at the system than when it is backwards. This section describes our approach for face attention estimation.

We have divided head poses in three main categories: "A" that comprehends all the frontal faces (faces looking directly at the camera), "B" that comprehends all the slightly sided faces (faces looking to some point slightly above, below or aside from the camera) and "C" that comprehends all the other faces (side faces, faces looking at some point in the ceiling or ground, backward heads). Figure 3 shows examples of each one of the categories employed.

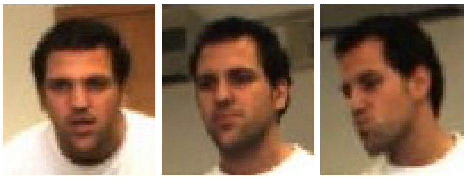


Figure 3: Head Pose Estimation: Classes A, B and C.

We have created a head pose database comprised by a total of 4000 samples equally distributed among the three classes. The database contain images of 21 different people (men and women), of different races, with different hair cuts and some of them wearing glasses. The database samples were manually classified into categories "A", "B" or "C" according to where people were looking at. All the images are gray-scale and 48x40 sized.

Since the information contained in the patterns is redundant, we have applied Principal Component Analysis (PCA) to reduce the data dimensionality. PCA (Henry and Dunteman, 1989) is a technique widely employed for dimensionality reduction able to retain those characteristics of the data set that contribute most to its variance, by keeping lower-order principal components and ignoring higher-order ones. Such low-order components often contain the "most important" aspects of the data. The more high-order characteristics we remove, the faster the process of training and estimation. However, it is important to use a number of characteristics that allow us to achieve good results without affecting our need to have real time results. We made some tests with different number of characteristics and we determined that 50 characteristics allowed a good trade-off between classification accuracy and computing time.

The training process has been carried out using SVM. For that purpose, we have employed the libsvm library (free software available in Internet (Chang and Lin, 2006)). For more information about SVM, the interest reader is referred to (Cristianini and Shawe-Taylor, 2000). To certificate that results were satisfactory before applying the model we trained the SVM with 85% of the data set and kept the remainder 15% to test the model generated. The result on the test set was of 93.14% of accuracy.

Our system estimates the attention of each detected person in real time despite of his/her movements. For that purpose, the head location in the camera image of each person is determined using the techniques described in Sect. 2. The head region of each person is resized to $48x40$ pixels and then, the first 50 principal components are extracted and passed to the SVM classifier.

SVM estimates the head pose in one of the three categories previously indicated. However, the classifier output is an instantaneous value that does not take into account past observations. In order to consider past observations, we define the variable $HP_{(t)}$ as:

$$HP_{(t)} = \alpha HP_{(t-1)} + (1-\alpha) SvmOut_t \qquad (1)$$

where $SvmOut_t$ is defined on the basis of the classifier output as:

$$SVMOut_t = \begin{cases} 1 & \text{if current output of SVM = "A";} \\ 0.5 & \text{if current output of SVM = "B";} \\ 0 & \text{if current output of SVM = "C".} \end{cases}$$

In Eq. 1, α is a weighting factor that ponders the influence of past observations. In this work we set $\alpha = 0.3$ that is sufficient to avoid abrupt variations and isolated pose estimation errors.

To deal with the uncertainty and vagueness in this process we use a linguistic variable called "Attention" and divide it into "High", "Medium" and "Low" values (see Fig. 2). This variable will take as input values the measures of face attention estimation considered by HP (Eq. 1). In figure 2 it is possible to see the labels for the variable "Attention".

3.2 Fuzzy System for Interest Estimation

Once the three linguistic variables have been defined, the rules base that integrates them are explained in this section. The idea that governs the definition of the rules base is dominated by the value of the variable *Attention*. If the attention has an high value the possibility of interest is also high depending on the distance and the angle of the person to the robot. If the attention is medium then the possibility of interest has to be decrease but like in the former case depending on the distance and angle. Finally if the attention is low, it means that the person is not looking at all to the area where the robot is located and the possibility of interest is defined as low or very low depending on the other variables. The rules for the case in which *Attention* is High are shown by Table 1. The other cases are expressed in a similar way using the appropriate rules. The output linguistic variable is *Interest* that has the five possible values shown by Figure 2(d).

Table 1: Rules in the case of high Attention.

IF			THEN
Attention	Distance	Angle	Interest
High	Low	Left	High
High	Low	Center	Very High
High	Low	Right	High
High	Medium	Left	Medium
High	Medium	Center	High
High	Medium	Right	Medium
High	High	Left	Low
High	High	Center	Medium
High	High	Right	Low

Finally to compute the value of possible interest, a fuzzy inference process is carried out using the operator minimum as implication operator. Then the output fuzzy sets are aggregated and the overall output is obtained. The overall output fuzzy set can be understood as a possibility distribution of the interest of the person in the $[0,1]$ interval. Therefore values near to 1 mean a high level of interest and vice versa.

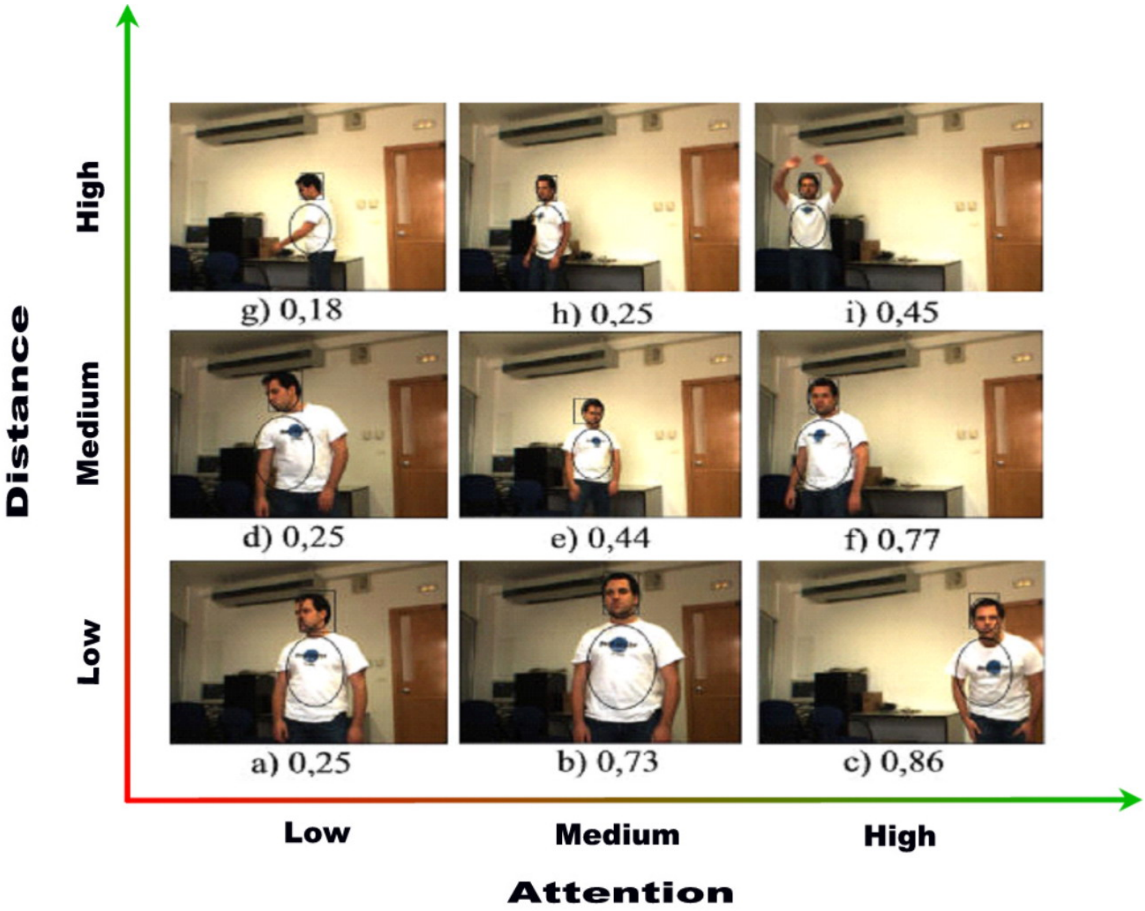


Figure 4: Interest according distance and attention. Frames in each row (*a*, *b* and *c*), (*d*, *e* and *f*) and (*g*, *h* and *i*) show the person at different distances while frames in each column (*a*, *d* and *g*), (*b*, *e* and *h*) and (*c*, *f* and *i*) show the person with different attention.

4 EXPERIMENTATION

A broader experimentation has been done to validate our system. All of these experimentations results were very satisfactory in respect to interest estimation using our system. Because of space reasons we opted to describe in detail only one of the experimentations. To perform the stereo process we have used images of size 320*x*240 and sub-pixel interpolation to enhance the precision in the stereo calculation. The operation frequency of our system is about 30 Hz without considering the time required for stereo computation.

Regarding the interest estimation, we have checked that the interest degree assigned to each tracked person increases and decreases dynamically accordingly to the behavior of the person in relation to the robot. To test it, one person has been recorded moving in front of the robot in a manner that it was possible to have frames of the person in all the situations regarding “Attention” and “Distance”. Frames from this video can be seen in Fig. 4. In this figure, frames *a*, *b*, and *c* show the person close to the robot (”Low Distance”) and looking in different ways. Interest is higher when when person looks towards the robot. In frames *d*, *e* and *f* it is possible to observe the same situation but at a higher distance (”Medium Distance”). Therefore the final interest value computed is slightly lower. In frames *g*, *h* and *i* we have the same person in the same situations (in respect to where he is looking at) but at even a higher distance from the robot. Therefore the final interest value computed is even lower. It is also possible to observe that when the person is looking approximately the same way (in frames *a*, *d*, *g* person is showing low attention, in frames *b*, *e*, *h* person is showing medium attention and in frames *c*, *f*, *i* he is showing high attention) interest varies usually according distance. The closer to the robot the higher the interest.

As it was expected, the higher value of interest

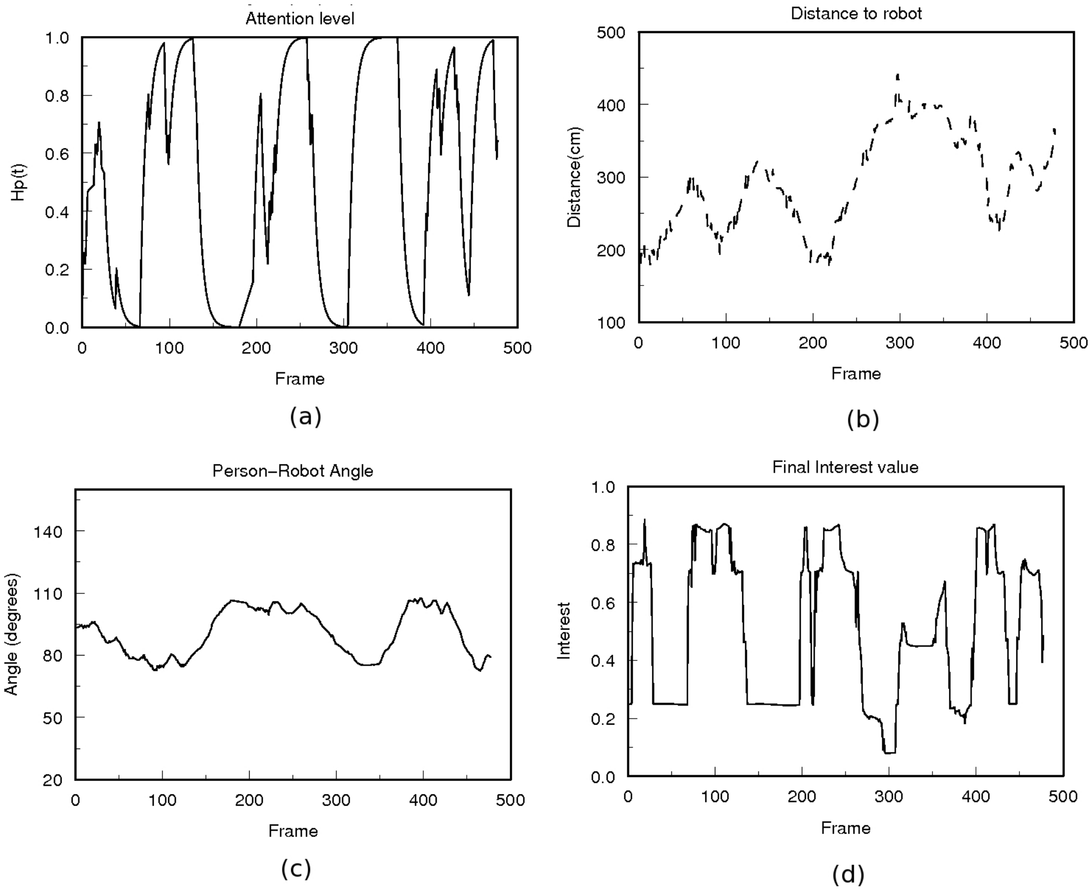


Figure 5: Graphs that show the variation of variables "Attention" (a), "Distance" (b), "Angle" (c) and "Interest" (d) during the video referred in Section 4.

is achieved in frame *c*, because the person is looking directly at the robot and it is very close to it. On the other hand, in frame *g* the person is not paying any attention to the robot, moving at a distance faraway from it. Therefore the lowest interest value is achieved in this frame.

It is also possible to observe in the graphs showed in Fig. 5 the variation of variables "Attention", "Distance" and "Angle" in Fig. 5.a, Fig. 5.b and Fig. 5.c respectively during the whole video. The forth graph Fig. 5.d is the interest computed also for the whole video. It is also possible to observe in these graphs the relationship among the interest computed and the other variables.

In order to better understand the performance of the system, several videos are available in the following web site http://decsai.ugr.es/∼ruipaul/interest.htm.

5 CONCLUSIONS AND FUTURE WORK

In this paper we have shown a system for detecting, tracking and estimating the interest of the people in the surroundings of a mobile robot, using stereo vision, head pose estimation by SVM and fuzzy logic. As a first step, the method for people detection and tracking has been briefly shown. While a person is being tracked, the fuzzy system computes a level of possibility about the interest that this person has in interacting with the robot. This possibility value is based on the position of the person in relation with the robot, as well as on an estimation of the attention that the person pays to the robot. To examine the attention that a person pays to the robot we analyze in real time the head pose of the person. This analysis is solved by a view based approach using Support Vector Machines. Thanks to SVM head pose can be detected achieving a great percentage of success that is no dependent of the morphological features of the heads. The experimentation shows that the system is able to detect the persons present in its vicinity, track their motions and give a value of possible interest on the interaction of the persons with the robot.

The proposed method can be easily updated in future works to analyze other types of input data as sounds or laser range finder. Also, the degree of interest will be useful to plan the actions of the robot towards the persons in order to allow a more natural human-robot interaction.

ACKNOWLEDGEMENTS

This work has been partially supported by the Spanish MEC project TIN2006-05565 and Andalusian Regional Government project TIC1670.

REFERENCES

Aguirre, E. and González, A. (2000). Fuzzy behaviors for mobile robot navigation: Design, coordination and fusion. *International Journal of Approximate Reasoning*, 25:255–289.

Bennewitz, M., Faber, F., Joho, D., Schreiber, M., and Behnke, S. (2005). Integrating vision and speech for conversations with multiple persons. In *IROS'05: Proceedings of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 2523 – 2528.

Bien, Z. and Song, W. (2003). Blend of soft computing techniques for effective human-machine interaction in service robotic systems. *Fuzzy Sets and Systems*, 134(1):5–25.

Birchfield, S. (1998). Elliptical Head Tracking Using Intensity Gradients and Color Histograms. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 232–237.

Chang, C. and Lin, C. (2006). Libsvm. a library for support vector machines. *http://www.csie.ntu.edu.tw/ cjlin/libsvm/*.

Comaniciu, D., Ramesh, V., and Meer, P. (2000). Real-Time Tracking of Non-Rigid Objects using Mean Shift. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 142–149.

Cristianini, N. and Shawe-Taylor, J. (2000). *An Introduction To Support Vector Machines (and other Kernel Based Methods)*. Cambridge University Press.

Darrell, T., Demirdjian, D., Checka, N., and Felzenszwalb, P. (2001). Plan-view trajectory estimation with dense stereo background models. In *Eighth IEEE International Conference on Computer Vision (ICCV 2001)*, volume 2, pages 628 – 635.

Fritsch, J., Kleinehagenbrock, M., Lang, S., Plötz, T., Fink, G. A., and Sagerer, G. (2003). Multi-modal anchoring for human-robot interaction. *Robotics and Autonomous Systems*, 43(2-3):133–147.

Ghidary, S. S., Nakata, Y., Saito, H., Hattori, M., and Takamori, T. (2002). Multi-modal interaction of human and home robot in the context of room map generation. *Autonomous Robots*, 13(2):169–184.

Haritaoglu, I., Beymer, D., and Flickner, M. (2002). Ghost 3d: detecting body posture and parts using stereo. In *Workshop on Motion and Video Computing*, pages 175 – 180.

Harville, M. (2004). Stereo person tracking with adaptive plan-view templates of height and occupancy statistics. *Image and Vision Computing*, 2:127–142.

Hayashi, K., Hashimoto, M., Sumi, K., and Sasakawa, K. (2004). Multiple-person tracker with a fixed slanting stereo camera. In *6th IEEE International Conference on Automatic Face and Gesture Recognition*, pages 681–686.

Henry, G. and Dunteman (1989). *Principal Components Analysis*. SAGE Publications.

Intel (2005). *OpenCV: Open source Computer Vision library*.

Kuhn, H. W. (1955). The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97.

Kulic, D. and Croft, E. (2003). Estimating intent for human robot interaction. In *International Conference on Advanced Robotics*, pages 810–815.

Lienhart, R. and Maydt, J. (2002). An Extended Set of Haar-Like Features for rapid Object detection. In *IEEE Conf. on Image Processing*, pages 900–903.

Muñoz-Salinas, R., Aguirre, E., and García-Silvente, M. (2006). People detection and tracking using stereo vision and color. *To appear in Image and Vision Computing*. Available online at www.sciencedirect.com.

PtGrey (2005). Bumblebee. Binocular stereo vision camera system. *http://www.ptgrey.com/products/bumblebee/index.html*.

Saffiotti, A. (1997). The uses of fuzzy logic in autonomous robot navigation. *Soft Computing*, 1:180–197.

Snidaro, L., Micheloni, C., and Chiavedale, C. (2005). Video security for ambient intelligence. *IEEE Transactions on Systems, Man and Cybernetics, Part A*, 35:133 – 144.

Song, W., Kim, D., Kim, J., and Bien, Z. (2001). Visual servoing for a user's mouth with effective intention reading in a wheelchair-based robotic arm. In *ICRA*, pages 3662–3667.

Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 511–518.

POSTERS

DATA INTEGRATION SOLUTION FOR PAPER INDUSTRY
A Semantic Storing Browsing and Annotation Mechanism for Online Fault Data

Sergiy Nikitin, Vagan Terziyan
Agora Center, University of Jyväskylä, Mattilanniemi 1, Jyväskylä, Finland
senikiti@cc.jyu.fi, vagan@it.jyu.fi

Jouni Pyötsiä
Metso Automation
Jouni.Pyotsia@metso.com

Keywords: Semantic Web, Semantic Storage, Information Integration, Paper Industry.

Abstract: A lot of IT solutions exist for simplification and time saving of industrial experts' activities. However, due to large diversity of tools and case-by-case software development strategy, big industrial companies are looking for an efficient and viable information integration solution. The companies have realized the need for an integrated environment, where information is ready for extraction and sophisticated querying. We present here a semantic web-based solution for logging and annotating online fault data, which is designed, and implemented for a particular business case of a leading paper machinery maintenance and automation company.

1 INTRODUCTION

Rapid changes and discontinuities in the 21st century business environment will challenge companies with the growing demand for competitive advantages within their business solutions. To ensure high flexibility, sustainable growth and profitability, companies have to search for new innovative approaches to products and services development. New innovative business solutions call for strong integration of automation technology, information and communication technology (ICT), and business processes. At the same time, embedded intelligence in different machines and systems gives new possibilities for automated business process operation over the network throughout the machines and systems life cycles.

The new emerging remote service solutions imply that products transform into life cycle services and these services, in turn, transform into customers' service processes. Business messages coming from intelligent machines and systems drive these processes, utilizing embedded intelligence and ICT solutions.

In the future, a variety of collaborative resources, like intelligent machines, systems and experts, will create a huge amount of new information during the life cycles of machines and systems. Message flow management and compression to on-line knowledge are already a demanding issue for the logging of product-related activities. On the other hand, optimization requirements demand more effective knowledge utilization and the speeding up of network-based learning in the process of collaboration between different resources.

Industry challenges the IT-sector with the new requirements that are dictated by the need to offer essentially new services to customers in order to be competitive in the market. These requirements may become hard to meet using conventional tools and approaches. The growth in the information volumes we want to store and process by integrating data from different sources leads to an unprecedented level of complexity. Modern Enterprise Resource Planning (ERP) systems are trying to provide integrated solutions for large companies. However the installation and adjustment of such systems may take a half a year, involving hundreds of consultants and subcontractors.

The current trend towards more open service-based computing environments is a new approach to componentization and components distribution. Service-oriented architecture aims to achieve a new level of reusability and business process flexibility; however, to ensure the interoperability between the components we need a common "glue" that would adjust the semantics of the data being exchanged. The Semantic Web technology (Berners-Lee, T., et al., 2001) introduces a set of standards and languages for representation of a domain model with the explicit semantics. The main instrument of domain model construction is ontology, which allows for domain data representation in a formalized and unified way.

In this paper we present a solution that utilizes Semantic Web technology to provide a tool for online maintenance data browsing, analysis and annotation. We utilize experience obtained in the SmartResource project (SmartResource, 2006) and apply the General Adaptation Framework (Kaykova et al., 2005) to align data with the domain ontology. The paper is organized as follows: In the next section we describe the paper machinery ICT infrastructure, and Section 3 presents the solution we have developed using Semantic Web tools and standards. We end with conclusions and future work.

2 IT INFRASTRUCTURE IN PAPER INDUSTRY

Metso Corporation is a global supplier of process industry machinery and systems as well as know-how and aftermarket services. The corporation's core businesses are fiber and paper technology, rock and minerals processing, and automation and control technology. Metso's strategy is based on an in-depth knowledge of its customers' core processes, close integration of automation and ICT, and a large installed base of machines and equipment. Metso's goal is to transform into a long-term partner for its customers. Based on the remote service infrastructure, it develops solutions and services to improve efficiency, usability and quality of customers' production processes throughout their entire life cycles.

2.1 Remote Service Infrastructure

Metso's remote service infrastructure consists of a service provider's Central Hub and several customers' Site Hubs, which are integrated over the network (see Figure 1).

The key issues in a Site Hub solution are: open standards, information security, reliability, connectivity and manageability.

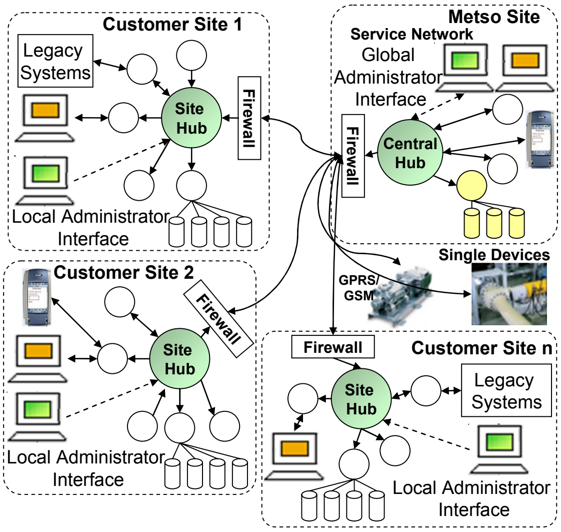


Figure 1: Site Hub network architecture.

Message Center is the main component of the Site Hub. It checks the validity of messages and routes them to the correct receivers. Messaging in Site Hub is based on Web Services technology.

Hub-based integrated infrastructure combined with secure connectivity allows easy incorporation of new business logic on both customer and Metso sites. A messaging mechanism between customers and Metso provides a very flexible medium for information exchange and new service provisioning.

3 LOGGING AND ANNOTATION OF MAINTENANCE DATA

The main purpose of the system we present here is to store alarm data, generated by paper machine monitoring systems. When an alarm happens, a SOAP/XML message (SOAP, 2003) is generated and sent to the Site Hub, which then forwards it to the Central Hub. We have established a message flow from the Central Hub to the computer at the university network, where messages are processed by our system.

3.1 Architecture of the System

The system can be divided into two main subcomponents – *Message Handler* and *Message Browser* (see Figure 2).

Message Handler receives and processes SOAP/XML messages from customers. It invokes the *Adapter* to transform the XML content into an

RDF-graph (RDF, 2004) object and store it in the Sesame RDF storage (Sesame).

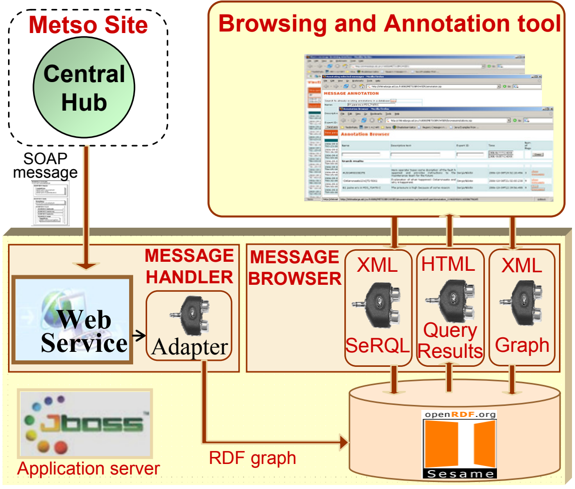


Figure 2: Architecture of the system.

The RDF storage contains an *Ontology* that plays the role of a schema for all data within the storage. Based on the analysis of SOAP/XML messages, we have defined main concepts (classes) with the corresponding properties (see Figure 3).

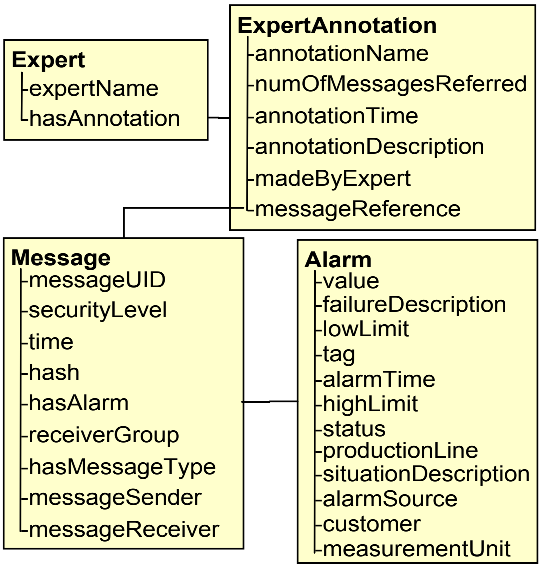


Figure 3: Ontology classes.

The *Message* class describes such message properties as message sender and receiver, message reception time, etc. The *Message* class also refers to an *Alarm* class, which contains information about the reason for message generation, such as measurements of sensors, status data and exact module of the production line where the alarm happened. The *ExpertAnnotation* class defines the structure for labelling groups of messages with an expert's decision and has references to instances of the *Message* class.

The *Message Browser* component provides a web-based interface for browsing and filtering messages stored in the RDF-storage, according to user-defined filtering criteria.

The purpose of message filtering is to distinguish the groups of messages leading to exceptional situations. The expert provides annotations for message groups which are stored to the RDF-storage and that can be used as samples for machine learning algorithms. The client-server interaction is implemented using AJAX technology (Garrett, 2005), which provides a more dynamic script-based interaction with the server (see Figure 4). When a user performs any action that requires invocation of server functionality, the script on a client side wraps the required parameters into XML format and sends it to the server. For example, in order to filter the messages, a user selects the needed parameters and specifies parameter values within the corresponding textboxes (see Figure 4).

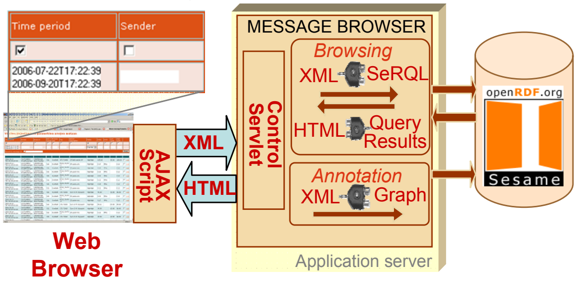


Figure 4: Client-server interaction.

On the server side, the *Control Servlet* handles the XML document. For filtering, it generates a SeRQL query (Broekstra, 2004) and executes it. On the client side, a dedicated callback script function processes the response and shows the result in a web browser.

3.2 Integration with the Agent Platform

We realize that the extension of the system will challenge the complexity of development and maintenance. That is why, following the autonomic computing paradigm (Kephart, 2003), we have tested agent-based scenario (see Figure 5) implemented on a JADE agent platform (Bellifemine; 2001). We have assigned an agent to manage RDF-storage activities (*Metso Storage Agent*) and provided a *Metso Expert Agent* to interact with a maintenance expert.

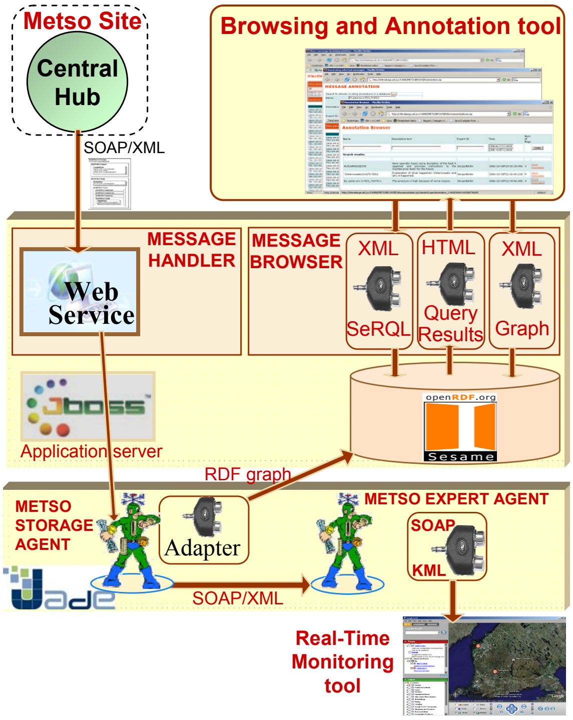


Figure 5: Agent-enabled system.

The messages coming from customers are handled by the *Metso Storage Agent*, which incorporates *Adapter* to perform transformation and storage. Then, the *Metso Storage Agent* sends the message to the *Metso Expert Agent*, which updates the situation on a Real-time Monitoring Tool and provides an expert with the message content and a link to the browsing and annotation tool.

4 CONCLUSIONS

Although we have succeeded with the implementation of the solution presented here, there are still many issues to cope with in order to meet key industrial requirements, such as scalability, maintainability and robustness. RDF-storages can handle billions of triples, but there are no mature semantic storage-oriented development patterns or guidelines. Nevertheless, the simplicity and efficiency of querying, as well as model extending, provide incontestable arguments in favour of semantic data storages. The ontological domain model brings more benefits to customers when there are more sources integrated. However, the complexity of such a system, if developed using conventional approaches, will be too burdensome to maintain and extend. In order to distribute the complexity, we introduce self-manageable entities in the agent-based communication scenario.

ACKNOWLEDGEMENTS

This research has been supported by the SmartResource project, funded by TEKES, and the industrial consortium of Metso Automation, TeliaSonera and ABB. The preparation of this paper was partially funded by the COMAS graduate school.

REFERENCES

Bellifemine, F., Poggi, A., and Rimassa, G. 2001. JADE: a FIPA2000 compliant agent development environment. In Proceedings of the Fifth international Conference on Autonomous Agents (Montreal, Quebec, Canada). AGENTS '01. ACM Press, New York, NY, 216-217. DOI= http://doi.acm.org/10.1145/375735.376120

Berners-Lee, T., Hendler, J., and Lassila, O. (2001) The Semantic Web, Scientific American, Vol. 284, No. 5, pp. 34-43.

Broekstra, J., Kampman A., and F. van Harmelen. Sesame: An Architecture for Storing and Querying RDF Data and Schema Information. In D. Fensel, J. Hendler, H. Lieberman, and W.Wahlster, editors, Semantics for the WWW. MIT Press, 2001.

Broekstra, J., Kampman, A. SeRQL: An RDF query and transformation language. In Proceedings of the International Semantic Web Conference, ISWC 2004, Hiroshima, Japan.

Garrett, J., Ajax: A New Approach to Web Applications (white paper).,http://www.adaptivepath.com/publications/essays/archives/000385.php, February 2005.

Kaykova O., Khriyenko O., Kovtun D., Naumenko A., Terziyan V., Zharko A., General Adaption Framework: Enabling Interoperability for Industrial Web Resources, In: International Journal on Semantic Web and Information Systems, Idea Group, ISSN: 1552-6283, Vol. 1, No. 3, July-September 2005, pp.31-63.

Kephart J.O., Chess D.M., 2003. The vision of autonomic computing, IEEE Computer, Vol. 36, No. 1, pp. 41-50

RDF – Resource Description Framework, A W3C Recommendation, Feb 2004, http://www.w3.org/RDF/

SmartResource – a TEKES funded project, http://www.cs.jyu.fi/ai/OntoGroup/SmartResource_details.htm

SOAP – Simple Object Access Protocol, A W3C Recommendation, 2003, http://www.w3.org/TR/soap/

FAST COMPUTATION OF ENTROPIES AND MUTUAL INFORMATION FOR MULTISPECTRAL IMAGES

Sié Ouattara, Alain Clément and François Chapeau-Blondeau
Laboratoire d'Ingénierie des Systèmes Automatisés (LISA), Université d'Angers
62 avenue Notre Dame du Lac, 49000 Angers, France
Corresponding author: *francois.chapeau-blondeau@univ-angers.fr*

Keywords: Multispectral images, Entropy, Mutual information, Multidimensional histogram.

Abstract: This paper describes the fast computation, and some applications, of entropies and mutual information for color and multispectral images. It is based on the compact coding and fast processing of multidimensional histograms for digital images.

1 INTRODUCTION

Entropies and mutual information are important tools for statistical analysis of data in many areas. For image processing, so far, these tools have essentially been applied to scalar or one-component images. The reason is that these tools are usually derived from multidimensional histograms, whose direct handling is feasible only in low dimension due to their memory occupation and related processing time which become prohibitively large as the number of image components increases. Here, we use an appraoch for multidimensional histograms allowing compact coding and fast computation, and show that this approach easily authorizes the computation of entropies and mutual information for multicomponent or multispectral images.

2 A FAST AND COMPACT MULTIDIMENSIONAL HISTOGRAM

We consider multispectral images with D components $X_i(x_1,x_2)$, for $i = 1$ to D, each X_i varying among Q possible values, at each pixel of spatial coordinate (x_1,x_2). A D-dimensional histogram of such an image would comprise Q^D cells. For an image with $N_1 \times N_2 = N$ pixels, only at most N of these Q^D cells can be occupied, meaning that, as D grows, most of the cells of the D-dimensional histogram are in fact empty. For example, for a common 512×512 RGB color image with $D = 3$ and $Q = 256 = 2^8$, there are $Q^D = 2^{24} \approx 16 \times 10^6$ colorimetric cells with at most only $N = 512^2 = 262\,144$ of them which can be occupied. We developed the idea of a compact representation of the D-dimensional histogram (Clément and Vigouroux, 2001; Clément, 2002), where only those cells that are occupied are coded. The D-dimensional histogram is coded as a linear array where the entries are the D-tuples (the colors) present in the image and arranged in lexicographic order of their components $(X_1,X_2,\dots X_D)$. To each entry (in number $\leq N$) is associated the number of pixels in the image having this D-value (this color). An example of this compact representation of the D-dimensional histogram is shown in Table 1.

The practical calculation of such a compact histogram starts with the lexicographic ordering of the N D-tuples corresponding to the N pixels of the image. The result is a linear array of the N ordered D-tuples. This array is then linearly scanned so as to merge the neighboring identical D-tuples while accumulating their numbers to quantify the corresponding population of pixels. With a dichotomic quick sort algorithm to realize the lexicographic ordering, the whole process of calculating the compact multidimensional histogram can be achieved with an average complexity of $O(N \log N)$, independent of the dimension D. Therefore, both compact representation and its fast calculation are afforded by the process for the

Table 1: An example of compact coding of the 3-dimensional histogram of an RGB color image with $Q = 256$. The entries of the linear array are the components $(X_1, X_2, X_3) = $ (R, G, B) arranged in lexicographic order, for each color present in the image, and associated to the poupulation of pixels having this color.

R	G	B	population
0	0	4	13
0	0	7	18
0	0	23	7
⋮	⋮	⋮	⋮
255	251	250	21
255	251	254	9

multidimensional histogram.

For example, for a 9-component 838×762 satellite image with $Q = 2^8$, the compact histogram was calculated in about 5 s on a standard 1 GHz-clock desktop computer, with a coding volume of 1.89 Moctets, while the classic histogram would take 3.60×10^{16} Moctets completely unmanageable by today's computers.

3 ENTROPIES AND MUTUAL INFORMATION FOR IMAGES

3.1 Fast Computation from Compact Histogram

The multidimensional histogram of an image, after normalization by the number of pixels, can be used for an empirical definition of the probabilities $p(\vec{X})$ associated to the D-values present in image $\vec{X}$. In the compact histogram, coded as in Table 1, only those D-values of $\vec{X}$ with nonzero probability are represented. This is all that is needed to compute any property of the image that is defined as a statistical average over the probabilities $p(\vec{X})$. This will be the case for the statistical moments of the distribution of D-values in the image (Romantan et al., 2002), for the principal axes based on the cross-covariances of the components X_i (Plataniotis and Venetsanopoulos, 2000), and for the entropies, joint entropies and mutual information that we consider in the sequel.

An entropy $H(\vec{X})$ for image $\vec{X}$ can be defined as (Russ, 1995)

$$H(\vec{X}) = -\sum_{\vec{X}} p(\vec{X}) \log p(\vec{X}) \ . \tag{1}$$

The computation of $H(\vec{X})$ of Eq. (1) from the normalized compact histogram from Table 1, is realized simply by a linear scan of the array while summing the terms $-p(\vec{X}) \log p(\vec{X})$ with $p(\vec{X})$ read from the last column. This preserves the overall complexity of $O(N \log N)$ for the whole process leading to $H(\vec{X})$.

The compact histogram also allows one to envisage the joint entropy of two multicomponent images $\vec{X}$ and $\vec{Y}$, with dimensions D_X and D_Y respectively. The joint histogram of $(\vec{X}, \vec{Y})$ can be calculated as a compact histogram with dimension $D_X + D_Y$, which after normalization yields the joint probabilities $p(\vec{X}, \vec{Y})$ leading to the joint entropy

$$H(\vec{X}, \vec{Y}) = -\sum_{\vec{X}} \sum_{\vec{Y}} p(\vec{X}, \vec{Y}) \log p(\vec{X}, \vec{Y}) \ . \tag{2}$$

A mutual information between two multicomponent images follows as

$$I(\vec{X}, \vec{Y}) = H(\vec{X}) + H(\vec{Y}) - H(\vec{X}, \vec{Y}) \ . \tag{3}$$

And again, the structure of the compact histogram preserves the overall complexity of $O(N \log N)$ for the whole process leading to $H(\vec{X}, \vec{Y})$ or $I(\vec{X}, \vec{Y})$.

So far in image processing, joint entropies and mutual information have essentially been used for scalar or one-component images (Likar and Pernus, 2001; Pluim et al., 2003), because the direct handling of joint histograms is feasible only in low dimension, due to their memory occupation and associated processing time which get prohibitively large as dimension increases. By contrast, the approach of the compact histogram of Section 2 makes it quite tractable to handle histograms with dimensions of 10 or more. By this approach, many applications of entropies and mutual information become readily accessible to color and multispectral images. We sketch a few of them in the sequel.

3.2 Applications of Entropies

The entropy $H(\vec{X})$ can be used as a measure of complexity of the multicomponent image $\vec{X}$, with application for instance to the following purposes:

- • An index for characterization / classification of textures, for instance for image segmentation or classification purposes.
- • Relation to performance in image compression, especially lossless compression.

For illustration, we use the entropy of Eq. (1) as a scalar parameter to characterize RGB color images carrying textures as shown in Fig. 1. The entropies $H(\vec{X})$ given in Table 2 were calculated from 512×512 three-component RGB images $\vec{X}$ with $Q = 256$. The whole process of computing a 3-dimensional histogram and the entropy took typically less than one second on our standard desktop computer. For comparison, another scalar parameter $\sigma(\vec{X})$ is also given in Table 2, as the square root of the trace of the variance-covariance matrix of the components of image $\vec{X}$. This parameter $\sigma(\vec{X})$ measures the overall average dispersion of the values of multicomponent image $\vec{X}$. For a one-component image $\vec{X}$, this $\sigma(\vec{X})$ would simply be the standard deviation of the gray levels. The results of Table 2 show a specific significance for the entropy $H(\vec{X})$ of Eq. (1), which does not simply mimic the evolution of a common measure like the dispersion $\sigma(\vec{X})$. As a complexity measure, $H(\vec{X})$ of Eq. (1) is low for synthetic images as `Chessboard` and `Wallpaper`, and is higher for natural images in Table 2.

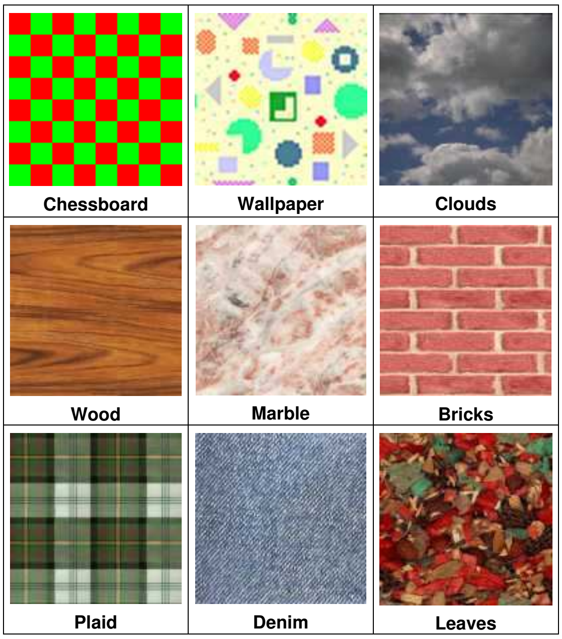


Figure 1: Nine three-component RGB images $\vec{X}$ with $Q = 256$ carrying distinct textures.

3.3 Applications of Mutual Information

The mutual information $I(\vec{X},\vec{Y})$, or other measures derived from the joint entropy $H(\vec{X},\vec{Y})$, can be used as an index of similarity or of relationship, between two multicomponent images $\vec{X}$ and $\vec{Y}$, with application for instance to the following purposes:

- Image matching, alignment or registration, especially in multimodality imaging.
- Reference matching, pattern matching, for pattern recognition.
- Image indexing from databases.

Table 2: For the nine distinct texture images of Fig. 1: entropy $H(\vec{X})$ of Eq. (1) in bit/pixel, and overall average dispersion $\sigma(\vec{X})$ of the components.

texture	$H(\vec{X})$	$\sigma(\vec{X})$
Chessboard	1.000	180.313
Wallpaper	7.370	87.765
Clouds	11.243	53.347
Wood	12.515	31.656
Marble	12.964	44.438
Bricks	14.208	57.100
Plaid	14.654	92.284
Denim	15.620	88.076
Leaves	17.307	74.966

- Homogeneity / contrast assessment for segmentation or classification purposes.
- Performance evaluation of image reconstruction, especially in lossy compression.
- Analysis via principal, or independent, component analysis.

For illustration, we consider a lossy compression on an RGB color $N_1 \times N_2$ image $\vec{X}$ via a JPEG-like operation consisting, on the $N_1 \times N_2$ discrete cosine transform of $\vec{X}$, in setting to zero a given fraction $(1 - \mathrm{CR}^{-1})$ of the high frequency coefficients, or equivalently in retaining only the $N_1/\sqrt{\mathrm{CR}} \times N_2/\sqrt{\mathrm{CR}}$ low-frequency coefficients. From this lossy compression of initial image $\vec{X}$, the decompression reconstructs a degraded image $\vec{Y}$. While varying the compression ratio CR, the similarity between images $\vec{X}$ and $\vec{Y}$ is measured here by the mutual information $I(\vec{X},\vec{Y})$ of Eq. (3) based on the 6-dimensional joint histogram for estimating the joint probabilities $p(\vec{X},\vec{Y})$. In addition, for comparison, we also used a more common measure of similarity formed by the cross-correlation coefficient $C(\vec{X},\vec{Y})$ between $\vec{X}$ and $\vec{Y}$, computed as one third of the sum of the cross-correlation coefficient between each marginal scalar component, R, G or B, of $\vec{X}$ and $\vec{Y}$. This $C(\vec{X},\vec{Y}) = 1$ if $\vec{X}$ and $\vec{Y}$ are two identical images, and it goes to zero if $\vec{X}$ and $\vec{Y}$ are two independent unrelated images. For the results presented in Fig. 2, the choice for initial image $\vec{X}$ is a 512×512 RGB `lena.bmp` with $Q = 256$. The whole process of the computation of an instance of the 6-dimensional joint histogram and the mutual information took around 3 s on our standard desktop computer.

Figure 2 shows that, as the compression ratio CR increases, the mutual information $I(\vec{X},\vec{Y})$ and the cross-correlation coefficient $C(\vec{X},\vec{Y})$ do not decrease in the same way. Compared to the mutual information $I(\vec{X},\vec{Y})$, it is known that the cross-correlation

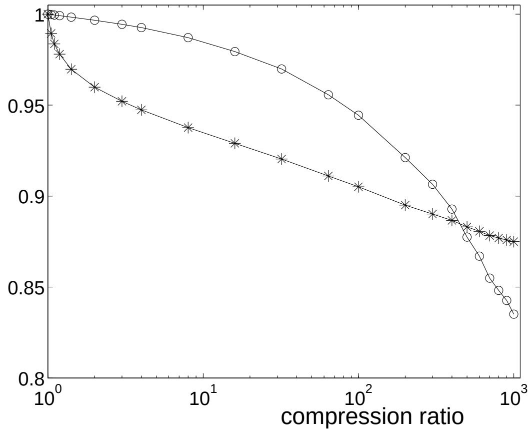


Figure 2: As a function of the compression ratio CR: (○) cross-correlation coefficient $C(\vec{X},\vec{Y})$ between initial RGB color image $\vec{X}$ and its compressed version $\vec{Y}$; ($*$) normalized mutual information $I(\vec{X},\vec{Y})/H(\vec{X})$ with the entropy $H(\vec{X}) = 16.842$ bits/pixel.

$C(\vec{X},\vec{Y})$ measures only part of the dependence between $\vec{X}$ and $\vec{Y}$. Figure 2 indicates that when the compression ratio CR starts to rise above unity, $I(\vec{X},\vec{Y})$ decreases faster than $C(\vec{X},\vec{Y})$, meaning that information is first lost at a faster rate than what is captured by the cross-correlation. Meanwhile, for large CR in Fig. 2, $I(\vec{X},\vec{Y})$ comes to decrease slower than $C(\vec{X},\vec{Y})$. This illustrates a specific contribution of the mutual information computed for multicomponent images.

Another application of mutual information between images can be found to assess a principal component analysis. On a D-component image $\vec{X} = (X_1,\dots X_D)$, principal component analysis applies a linear transformation of the X_i's to compute D principal components $(P_1,\dots P_D)$ with vanishing cross-correlation among the P_i's, in such a way that some P_i's can be selected for a condensed parsimonious representation of initial image $\vec{X}$. An interesting quantification is to consider the mutual information $I(\vec{X},P_i)$. From its theoretical properties, the joint entropy $H(\vec{X},P_i)$ reduces to $H(\vec{X})$ because P_i is deterministically deduced from $\vec{X}$, henceforth $I(\vec{X},P_i) = H(P_i)$. This relationship has been checked (on several 512 × 512 RGB color images with $D = 3$) to be precisely verified by our empirical entropy estimators for $I(\vec{X},P_i)$ based on the computation of $(D+1)$-dimensional histograms for $(\vec{X},P_i)$. This offers a quantification of the relation between $\vec{X}$ and its principal components P_i; a subset of the whole P_i's could be handled in a similar way. Another useful quantification shows that principal component analysis, although it cancels cross-correlation between the components, does not cancel dependence between them, and sometimes it may even increase it in some sense, as illustrated by the behavior of the mutual information in Table 3, with $I(P_1,P_2)$ larger than $I(X_1,X_2)$ for image (2). The mutual information can serve as a measure to base other separation or selection schemes of the components from an initial multispectral image $\vec{X}$.

Table 3: For a 512 × 512 RGB color image $\vec{X}$ with $D = 3$ and $Q = 256$: cross-correlation coefficient $C(\cdot,\cdot)$ and mutual information $I(\cdot,\cdot)$ of Eq. (3), between the two initial components X_1 and X_2 with largest variance, and between the two first principal components P_1 and P_2 after principal component analysis of $\vec{X}$. (1) image $\vec{X}$ is `lena.bmp`. (2) image $\vec{X}$ is `mandrill.bmp`.

	$C(X_1,X_2)$	$I(X_1,X_2)$	$C(P_1,P_2)$	$I(P_1,P_2)$
(1)	0.879	1.698	0.000	0.806
(2)	0.124	0.621	0.000	0.628

4 CONCLUSION

We have reported the fast computation and compact coding of multidimensional histograms and showed that this approach authorizes the estimation of entropies and mutual information for color and multispectral images. Histogram-based estimators of these quantities as used here, become directly accessible with no need of any prior assumption on the images. The performance of such estimators clearly depends on the dimension D and size $N_1 \times N_2$ of the images; we did not go here into performance analysis, especially because this would require to specify statistical models of reference for the measured images. Instead here, more pragmatically, on real multicomponent images, we showed that, for entropies and mutual information, direct histogram-based estimation is feasible and exhibits natural properties expected for such quantities (complexity measure, similarity index, ...). The present approach opens up the way for further application of information-theoretic quantities to multispectral images.

REFERENCES

Clément, A. (2002). *Algorithmes et outils informatiques pour l'analyse d'images couleur. Application à l'étude de coupes histologiques de baies de raisin en microscopie optique.* Ph. D. thesis, University of Angers, France.

Clément, A. and Vigouroux, B. (2001). Un histogramme compact pour l'analyse d'images multi-composantes.

In *Proceedings 18è Colloque GRETSI sur le Traitement du Signal et des Images*, pages 305–307, Toulouse, France, 10–13 Sept. 2001.

Likar, B. and Pernus, F. (2001). A hierarchical approach to elastic registration based on mutual information. *Image and Vision Computing*, 19:33–44.

Plataniotis, K. N. and Venetsanopoulos, A. N. (2000). *Color Image Processing and Applications*. Springer, Berlin.

Pluim, J. P. W., Maintz, J. B. A., and Viergever, M. A. (2003). Mutual-information-based registration of medical images: A survey. *IEEE Transactions on Medical Imaging*, 22:986–1004.

Romantan, M., Vigouroux, B., Orza, B., and Vlaicu, A. (2002). Image indexing using the general theory of moments. In *Proceedings 3rd COST 276 Workshop on Information and Knowledge Management for Integrated Media Communications*, pages 108–113, Budapest, Hungary, 11–12 Oct. 2002.

Russ, J. C. (1995). *The Image Processing Handbook*. CRC Press, Boca Raton.

PROBABILISTIC MAP BUILDING CONSIDERING SENSOR VISIBILITY

Kazuma Haraguchi, Jun Miura
Department of Computer-Controlled Mechanical Systems, Graduate School of Engineering, Osaka University
2-1 Yamadaoka, Suita, Osaka, Japan
haraguti@cv.mech.eng.osaka-u.ac.jp, jun@mech.eng.osaka-u.ac.jp

Nobutaka Shimada, Yoshiaki Shirai
Department of Computer Science Faculty of Science Engineering Ritsumeikan University
1-1-1 Nojihigashi, kusatsu, shiga, Japan
shimada@ci.ritsumei.ac.jp, shirai@i.ci.ritsumei.ac.jp

Keywords: Obstacle Map Generation, Bayes Theorem, Occlusion, Spatial Dependency, Visibility.

Abstract: This paper describes a method of probabilistic obstacle map building based on Bayesian estimation. Most active or passive obstacle sensors observe only the most frontal objects and any objects behind them are occluded. Since the observation of distant places includes large depth errors, conventional methods which does not consider the sensor occlusion often generate erroneous maps. We introduce a probabilistic observation model which determines the visible objects. We first estimate probabilistic visibility from the current viewpoint by a Markov chain model based on the knowledge of the average sizes of obstacles and free areas. Then the likelihood of the observations based on the probabilistic visibility are estimated and then the posterior probability of each map grid are updated by Bayesian update rule. Experimental results show that more precise map building can be bult by this method.

1 INTRODUCTION

The well known SLAM(Simultaneous Localization And Mapping)(Montemerlo and Thrun, 2002; Thrun et al., 2004; Grisetti et al., 2005) process could be divided in two parts: position localization using current observations and the current map, and the map update using the estimated position. This paper discusses the problem of the conventional map update methods, *sensor occlusion*, and proposes a novel updating method solving it.

The positions of landmarks and obstacles should be represented on the map for navigation(Thrun, 2002b). There are two representations available for them:

1. Feature point on the obstacle
2. Obstacle existence on small map grid.

Former representation requires feature identification and matching and the feature position is update when new observation is available(Suppes et al., 2001). Latter representation identifies the map grid from which the observation comes. Since laser range sensors, ultrasonic sensors and stereo image sensors can precisely control the direction of transmitting and receiving light or sound, the place from which observation comes is easy to identify. Thus we use the latter map representation here.

The obstacle sensor output always includes observation errors. For example flight time measurement of light or sound have errors caused by wave refraction, diffraction and multi-path reflection. Stereo image sensor also has correspondence failures of image features. These errors lead to large errors or ghost observations. Therefore the map building should be established in a probabilistic way based on a certain error distribution model.

Here we use a probabilistic occupancy grid map for the map representation, which stores the obstacle existence probability in each map grid. In this representation, the obstacle existence probability of each grid is updated by evaluating the likelihood of obtained observation for the grid and integrate it with its prior probability in the way of Bayesian estimation. While a generic solution of this map building formulation is solved by Markov Random Field (MRF)

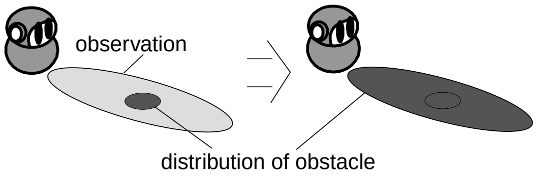


Figure 1: Result of update when the robot gets the observation of the obstacle of substantial margin of error.

(Rachlin et al., 2005), its computational cost is huge, especially in case of high grid resolution. Previous methods (Miura et al., 2002; Thrun, 2002a; Moravec, 1988) use the following assumptions for reduction of computation:

1. Observation obtained for each map grid is probabilistically independent (it depends on only the state (obstacle existence) of that grid).
2. Obstacle existence of each map grid is independent to each other.

Most obstacle sensors observe the most frontal objects and any objects behind them are occluded. Since such sensors having *sensor occlusion* characteristic does not satisfy the above assumption 1, the following serious problems are caused by forcedly using the assumption.

As shown in the left side picture of Figure 1, suppose that dark grey ellipsoidal region has higher probability of obstacle existence than the outside. Then suppose that the obstacle observation with large error (ex. distant observation with stereo vision) is obtained shown as the light grey region in the figure. In that situation, that observation probably comes from the most frontal part of the dark grey region. The assumption 1, however, the obstacle existence probability of each map grid is independently updated by integrating the observation and it is obviously overestimated. As a result everywhere distant from the current robot position tends to be estimated as obstacle in the map. Therefore the assumption 1 should be rejected.

In addition, the assumption 2 also leads to the other problems. If the obstacle existence in each map grid is independent, the probability of that a certain area is open as free-space is estimated as the product of the probability of each map grid. This leads to an obviously irrational result that more precise grid resolution is adopted, abruptly smaller becomes the free-space probability of the same area (in other words, the viewing field is more invisible due to occluding obstacles).

In real scenes, obstacles and free-spaces has a certain size. This points out the existence of co-occurrence between the adjacent grids, called *spatial continuity*. Since the co-occurrence becomes larger when the grid resolution is more precise, the free-space probability can be correctly estimated regardless of the grid resolution. Thus the assumption 2 also should be rejected.

Our map building method correctly considers sensor occlusion and spatial continuity. A certain obstacle is visible if and only if the space between the sensor and the obstacle is entirely open as free-space. Therefore we introduce a novel method of estimating *visibility* of the obstacle on each map grid by considering spatial continuity, and updating the obstacle existence probability with proper consideration of sensor occlusion.

2 MAP BUILDING CONSIDERING SENSOR VISIBILITY

2.1 Joint Probability of Adjacent Grids on Each Viewing Ray

We first divide the map grids into multiple viewing rays. The sensor observes the existence of obstacle on each ray. In this paper, we represent the viewing ray as the 4-connected digital line as shown in Figure 2. On each ray we consider the probabilistic grid state (whether obstacle exists on the grid or not) as a simple Markov chain. Thus each grid state can be represented as the joint probability of two grid states adjacent on each ray.

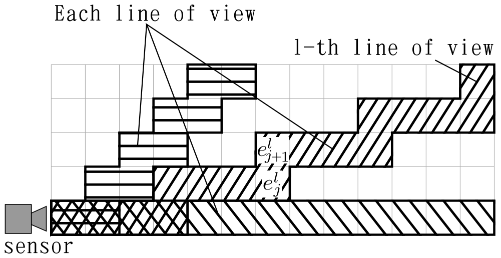


Figure 2: Approximated Lines of view.

We first estimate the joint probability of the jth and $j+1$th grid (the jth grid is nearer to the sensor) on a viewing ray (see Figure 2). Let $e^l_j = \left\{E^l_j, \bar{E}^l_j\right\}$ be the state of the jth grid on the lth ray (E:occupied by an obstacle, $\bar{E}$: not occupied), and $P(e^l_j, e^l_{j+1}|\mathbf{O})$ be the joint probability of the jth and $j+1$th grid under $\mathbf{O}$, the series of the previous observations. Then the joint probability after the latest observation $\mathbf{o}$ obtained

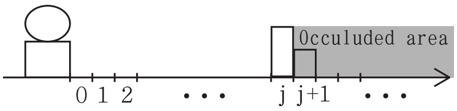


Figure 3: Occluded area behind the j-th grid.

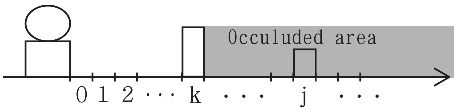


Figure 4: Occluded area behind the k-th grid.

is calculated as :

$$P(e^l_j,e^l_{j+1}|\mathbf{o},\mathbf{O}) = \alpha^l P(o^l|e^l_j,e^l_{j+1},\mathbf{O})P(e^l_j,e^l_{j+1}|\mathbf{O}). \qquad (1)$$

o^l, α^l, $P(e^l_j,e^l_{j+1}|\mathbf{O})$ and $P(o^l|e^l_j,e^l_{j+1},\mathbf{O})$ respectively denote the observation on the ray l, a normalizing constant, the prior probability and the likelihood. The likelihood $P(o^l|e^l_j,e^l_{j+1},\mathbf{O})$ should be calculated by considering the sensor visibility as the following section.

2.2 Likelihood Considering Sensor Visibility

Since an observation on the ray l, o^l, depends on grid states just on the the ray, the likelihood $P(o^l|e^l_j,e^l_{j+1},\mathbf{O})$ is calculated as:

$$P(o^l|e^l_j,e^l_{j+1},\mathbf{O}) = \sum_{\mathbf{m}^l\in\Omega^l} P(o^l|\mathbf{m}^l,e^l_j,e^l_{j+1},\mathbf{O})P(\mathbf{m}^l|e^l_j,e^l_{j+1},\mathbf{O}) \qquad (2)$$

where $\mathbf{m}^l$ denotes grid states on the ray l except e^l_j,e^l_{j+1}. The direct calculation of Eq. 2 requires huge summation of 2^{N_l} order, where N_l denotes the number of grids on the ray l.

Actually this calculation is drastically reduced by considering the sensor visibility. There exist four cases of the adjacent grid states: (E_j,E_{j+1}), $(E_j,\overline{E}_{j+1})$, $(\overline{E}_j,E_{j+1})$, $(\overline{E}_j,\overline{E}_{j+1})$. In (E_j,E_{j+1}) case (this means both the grid j and $j+1$ are occupied), since grid j occludes $j+1$ as shown in Figure 3, the likelihood $P(o^l|E^l_j,E^l_{j+1},\mathbf{O})$ is no longer dependent on $j+1$, and then it is represented as

$$P(o^l|E^l_j,E^l_{j+1},\mathbf{O})= P(o^l|E^l_j,\mathbf{O}). \qquad (3)$$

The above likelihood is acceptable only when grid j is visible from the sensor, namely whole grids between the sensor and grid j are empty. If not so, the most frontal occupied grid $k(< j)$ is observed (see Figure 4). Here, define an stochastic event F_k as follows:

$$F_k = \begin{cases} E_0 & (k=0) \\ \overline{E}_0 \cap \overline{E}_1 \cap \cdots \cap \overline{E}_{k-1} \cap E_k & (k>0). \end{cases}$$

Since $F_0,\cdots,F_k$ are mutually exclusive events, the right-hand of Eq.3 is expanded as follows:

$$\begin{aligned} P(o^l|E^l_j,\mathbf{O}) &= \sum_{k=0}^{j} P(o^l|F^l_k,E^l_j,\mathbf{O})P(F^l_k|E^l_j,\mathbf{O}) \\ &= \sum_{k=0}^{j} P(o^l|F^l_k)P(F^l_k|E^l_j,\mathbf{O}) \qquad (4) \end{aligned}$$

because o^l is no longer dependent on any grid behind grid k nor the previous observations $\mathbf{O}$.

The likelihoods for the rest three cases $(E_j,\overline{E}_{j+1})$, $(\overline{E}_j,E_{j+1})$, $(\overline{E}_j,\overline{E}_{j+1})$, is represented as follows in the same considerations:

$$P(o^l|E^l_j,\overline{E}^l_{j+1},\mathbf{O}) = \textstyle\sum_{k=0}^{j} P(o^l|F^l_k)P(F^l_k|E^l_j,\mathbf{O}) \qquad (5)$$

$$\begin{aligned} P(o^l|\overline{E}^l_j,E^l_{j+1},\mathbf{O}) &= \textstyle\sum_{k=0}^{j-1} P(o^l|F^l_k)P(F^l_k|\overline{E}^l_j,\mathbf{O}) \\ &\quad +P(o^l|F^l_{j+1})P(F^l_{j+1}|\mathbf{O}) \qquad (6) \end{aligned}$$

$$\begin{aligned} P(o^l|\overline{E}^l_j,\overline{E}^l_{j+1},\mathbf{O}) &= \textstyle\sum_{k=0}^{j-1} P(o^l|F^l_k)P(F^l_k|\overline{E}^l_j,\mathbf{O}) \\ &\quad +\textstyle\sum_{k=j+1}^{\infty} P(o^l|F^l_k)P(F^l_k|\mathbf{O}). \qquad (7) \end{aligned}$$

$P(o^l|F^l_k)$ in Eqs.(4)-(7) is a sensor kernel model which defines the measurement error distribution of the sensor observation. This is provided for each kind of sensor. These equations say that the likelihood of grid j and $j+1$ is contributed by grid k ahead grid j according to $P(F^l_k|E^l_j,\mathbf{O})$, $P(F^l_k|\overline{E}^l_j,\mathbf{O})$, $P(F^l_k|\mathbf{O})$, which mean conditional *visibilities* of grid k.

Since these visibilities are required to be estimated before the above likelihood calculations, they should be estimated by calculating the Markov chain because the actual grid states are not independent due to spatial continuity. The Markov chain is calculated based on the grid state probabilities (joint probability of e_j, and e_{j+1}) obtained in the previous time slice:

$$\begin{aligned} P(F^l_k|E^l_j,\mathbf{O}) =& P(\overline{E}^l_0|\overline{E}^l_1,\mathbf{O})P(\overline{E}^l_1|\overline{E}^l_2,\mathbf{O})P(\overline{E}^l_2|\overline{E}^l_3,\mathbf{O}) \\ & \cdots P(\overline{E}^l_{k-1}|E^l_k,\mathbf{O})P(E^l_k|E^l_j,\mathbf{O}) \qquad (8) \end{aligned}$$

$$\begin{aligned} P(F^l_k|\overline{E}^l_j,\mathbf{O}) =& P(\overline{E}^l_0|\overline{E}^l_1,\mathbf{O})P(\overline{E}^l_1|\overline{E}^l_2,\mathbf{O})P(\overline{E}^l_2|\overline{E}^l_3,\mathbf{O}) \\ & \cdots P(\overline{E}^l_{k-1}|E^l_k,\mathbf{O})P(E^l_k|\overline{E}^l_j,\mathbf{O}) \qquad (9) \end{aligned}$$

$$\begin{aligned} P(F^l_k|\mathbf{O}) =& P(\overline{E}^l_0|\overline{E}^l_1,\mathbf{O})P(\overline{E}^l_1|\overline{E}^l_2,\mathbf{O})P(\overline{E}^l_2|\overline{E}^l_3,\mathbf{O}) \\ & \cdots P(\overline{E}^l_{k-1}|E^l_k,\mathbf{O})P(E^l_k|\mathbf{O}) \qquad (10) \end{aligned}$$

where $P(E^l_k|\mathbf{O})$, $P(\overline{E}^l_{k-1}|E^l_k,\mathbf{O})$ and $P(\overline{E}^l_q|\overline{E}^l_{q+1},\mathbf{O})$ $(0 \le q < k-1)$ are calculated as follows:

$$P(E^l_k|\mathbf{O})= P(E^l_k,E^l_{k+1}|\mathbf{O}) + P(E^l_k,\overline{E}^l_{k+1}|\mathbf{O}) \qquad (11)$$

$$P(\overline{E}^l_{k-1}|E^l_k,\mathbf{O}) = \frac{P(\overline{E}^l_{k-1},E^l_k|\mathbf{O})}{P(E^l_{k-1},E^l_k|\mathbf{O})+P(\overline{E}^l_{k-1},E^l_k|\mathbf{O})} \qquad (12)$$

$$P(\overline{E}^l_q|\overline{E}^l_{q+1},\mathbf{O}) = \frac{P(\overline{E}^l_q,\overline{E}^l_{q+1}|\mathbf{O})}{P(\overline{E}^l_q,\overline{E}^l_{q+1}|\mathbf{O})+P(E^l_q,\overline{E}^l_{q+1}|\mathbf{O})}. \qquad (13)$$

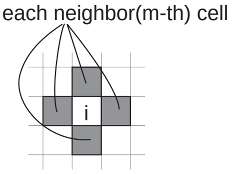


Figure 5: Adjacent grids.

$P(E^l_k|E^l_j,\mathbf{O})$ and $P(E^l_k|\bar{E}^l_j,\mathbf{O})$ are also obtained by calculating the Markov chain:

$$P(E^l_k,E^l_j|\mathbf{O}) = P(E^l_k|\mathbf{O})P(E^l_j|\mathbf{O}) + \alpha_{k,j}\prod_{q=k}^{j-1} c^l_{q,q+1} \quad (14)$$

$$P(E^l_k,\bar{E}^l_j|\mathbf{O}) = P(E^l_k|\mathbf{O})P(\bar{E}^l_j|\mathbf{O}) - \alpha_{k,j}\prod_{q=k}^{j-1} c^l_{q,q+1} \quad (15)$$

where

$$c^l_{q,q+1} = \frac{P(E^l_q,E^l_{q+1}|\mathbf{O}) - P(E^l_q|\mathbf{O})P(E^l_{q+1}|\mathbf{O})}{\sqrt{P(E^l_q|\mathbf{O})P(\bar{E}^l_q|\mathbf{O})P(E^l_{q+1}|\mathbf{O})P(\bar{E}^l_{q+1}|\mathbf{O})}} \quad (16)$$

$$\alpha_{k,j} = \sqrt{P(E^l_k|\mathbf{O})P(\bar{E}^l_k|\mathbf{O})P(E^l_j|\mathbf{O})P(\bar{E}^l_j|\mathbf{O})} \quad (17)$$

$c^l_{q,q+1}$ means correlation between the adjacent grid e^l_q and e^l_{q+1}. The grid visibilities are estimated from Eqs.(8)-(17), then the likelihoods considering sensor visibility, Eqs.(4)- (7), are calculated. Finally the posteriors, Eq.(1) is updated for each grid.

2.3 Posterior Update Across Viewing Rays

The grid posteriors for each viewing ray, estimated in the way of the previous section, conflict with the posteriors of the adjacent rays, because they actually have interactions across the rays and the independent update of each ray is just an approximation. True posteriors should satisfy the following two constraints: the first one is

$$\sum_{e_i,e_j \in \{E,\bar{E}\}} P(e_i,e_j) \equiv 1 \quad (18)$$

where grid i and j is adjacent, and the second one is

$$P(e_i) \equiv P(e_i,E_m) + P(e_i,\bar{E}_m) \quad (19)$$

for every grid m adjacent to grid j (see Figure 5). Therefore we estimate the true posteriors by least squares method under the constraints Eqs.(18),(19). The minimised error Δ^2 is written as

$$\Delta^2 = \sum_{i,j}\sum_{e_i,e_j}\left\{P^*(e_i,e_j) - P(e_i,e_j)\right\}^2 \quad (20)$$

where $P^*(e_i,e_j)$ is the conflicting posteriors, $P(e_i,e_j)$ is the estimated posteriors satisfying the constraints. This minimization is easily solved by Lagrange's method in low cost.

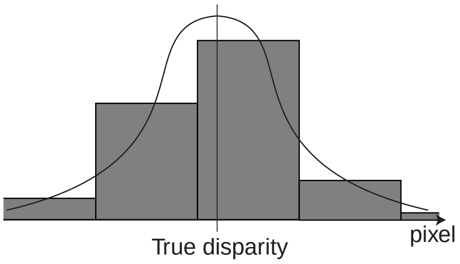


Figure 6: Probability density of the observation of the disparity.

3 EXPERIMENTS

3.1 Sensor Kernel Model of Stereo Vision

We compared a map building result for the simulation on a viewing ray and a real indoor scene using our method and a conventional method considering no sensor visibility nor spatial continuity.

In these experiments, we used edge-based stereo vision for observation of obstacles. Stereo vision provides depth information for each edge feature and it is well-known that the measurement error is inversely proportional to square of the depth. In addition erroneous feature matching can be found in a certain probability. Thus its sensor kernel model $P(o^l|F^l_k)$, required in Eqs.(4)- (7), are defined here as follows:

$$P(o^l|F^l_k) = P(T)P(o^l|F^l_k,T) + P(\bar{T})P(o^l|F^l_k,\bar{T}) \quad (21)$$

where o^l is the measured disparity for the viewing ray l, $P(T)$_(fixed to 0.8 in the following experiment, $P(T)+P(T \equiv 1)$) is the probability of obtaining a correct matching, $P(o^l|F^l_k,T)$ is a gaussian (see Figure 6), $P(o^l|F^l_k,\bar{T})$ is a uniform distribution over disparity range [0,60]. The obtained sensor kernel model $P(o^l|F^l_k)$ is shown in Figure 7.

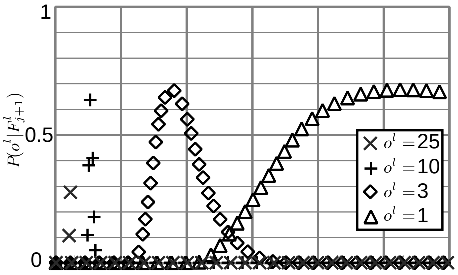


Figure 7: Observation model.

Our method updates the posteriors based on the likelihoods considering sensor visibility, Eqs.(4)-(7).

In the conventional method, we assume that state of each grid is independent from state of other grid as:

$$P(e_i^l|\mathbf{o},\mathbf{O}) = \frac{P(o_i^l|e_i^l,\mathbf{O})P(e_i^l|\mathbf{O})}{P(o_i^l|E_i^l,\mathbf{O})P(E_i^l|\mathbf{O})+P(o_i^l|\bar{E}_i^l,\mathbf{O})P(\bar{E}_i^l|\mathbf{O})} \qquad (22)$$

where o_i^l denotes the observation on ith grid and on lth ray. On this assumption we can calculate each grid indipendently. The following conventional likelihoods (Miura et al., 2002; Thrun, 2002a; Moravec, 1988) are adopted as a benchmark:

$$\begin{aligned} P(o_k^l = o_E|E_k^l) &= P(o^l|F_k^l) && (23)\\ P(o_k^l = o_E|\bar{E}_k^l) &= P(\bar{T}) && (24)\\ P(o_k^l = o_{\bar{E}}|E_k^l) &= P(\bar{T}) && (25)\\ P(o_k^l = o_{\bar{E}}|\bar{E}_k^l) &= P(T)\{1-P(o^l|F_k^l,T)\}. && (26) \end{aligned}$$

3.2 Posterior Update on a Viewing Ray

Our method of posterior update, of course, requires an initial prior distribution. In addition it requires an initial correlation parameter using for estimating the spatial continuity. In the following experiment, the initial prior $P(E_i)$ is uniformly set to 0.1 and the spatial correlation $c_{q,q+1}$ to 0.871 where the grid size is 5cm × 5cm These initial parameters are estimated based on average obstacle size (40cm × 40cm) in actual room scene samples.

Since the aim of this paper is to show the effectiveness of our posterior updating considering sensor visibility, we suppose that the exact robot position and orientation is known (NOT SLAM problem).

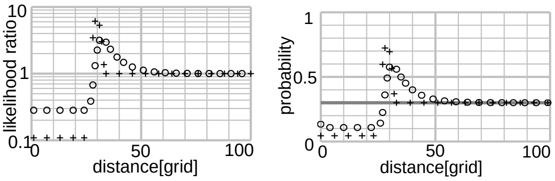
We compare our method with the convensional one without considering sensor visibility. We use three situations for the map update.

3.2.1 Case of Prior Probability of Uniform Distribution

Figure 8 shows the likelihood ratio and the posteriors when we take the observation (equivalent to the disparity of 10 pixels) in the uniform disribution. With considering sensor visibility, posteriors in the grid just behind the 30th grid is higher than that without considering visibility because we consider the average size of obstacles(equivalent to 40cm) in this experiment (see section 3.1).

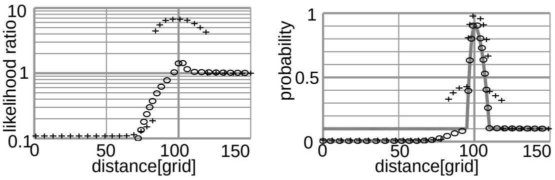
3.2.2 Case of Uncertain Observation from Distant Place

In Figure 9 we establish that posteriors are estimated high from the 95th to the 110th grids after we obtain the observations. Figure 9 shows the likelihood and the posteriors when we get the observation (equivalent to the disparity of 3 pixels) with large error. Without considering visibility, posteriors in the range from the 80th to the 120th grid (equivalent to the disparity uncertainty of 3 pixels) becomes high. But we can not obtain the information in the range from the 90th to the 120th grid because we obtain the observation with the error which is larger than distribution of obstacles. For this reason with considering visibility, posteriors in the range from the 90th to the 120th grid does not change as Figure 9(b). On the other hand, in front of the range we can obtain the information that the obstacles does not exist. For this reason both methods show that posterior probability of obstacles existence becomes low in front of the range.



(a) Likelihood (b) Result of posterior update

Figure 8: Result of posterior update when the robot get the observation of the obstacle in the uniformly-distributed probablity area. ”+” represents results of the conventional method. ”◯” represents results of our method. The gray curve in (b) represents the prior probablity.

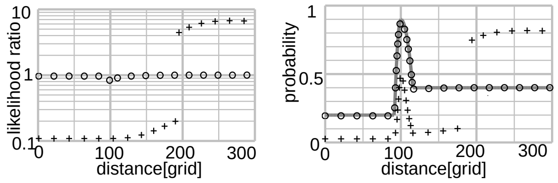


(a) Likelihood (b) Result of posterior update

Figure 9: Result of posterior update when the robot get the observation of the obstacle in the uniformly-distributed probablity area. ”+” represents results of the conventional method. ”◯” represents results of our method. The gray curve in (b) represents the prior probablity.

3.2.3 Case of Failure Observation for Occluded Area

In Figure 10 we establish that we obtain the failure observation for an occluded area when existence of obstacle is obvious. The error of this observation is about from the 200th to the 560th grid (equivalent to the disparity of 1 pixel). But it is highly possible that the observation is failure because obstacle existence about the 100th grid is already know. With considering sensor visibility, the failure observation is automatically detected and the posteriors kept unchanged.



(a) Likelihood (b) Result of posterior update

Figure 10: Result of posterior update when the robot get the observation of the obstacle in the uniformly-distributed probablity area. ”+” represents results of the conventional method. ”◯” represents results of our method. The gray curve in (b) represents the prior probablity.

3.3 Result of Map Building

Next we show the result of map building for an actual room as shown in Figure 11. The mobile robot moved in the room observing obstacles with stereo cameras, starting from *A* point in Figure 11 via *B*, *C*, *B*, *D* through the grey line in the figure, and finally arrived at *B* point.

Figure 12 shows a view captured by the left camera. Figure 12(a) is a view from *A* point, and Figure 12(b) is another view from *B* point. We compared the built map of our method (Figure 14) to that of the conventional one (Figure 13). Sub-figure (a) and (b) of each figure shows the temporary map when the robot reached to *B* point via *A*, *B*, *C* and *D* point via *A*, *B*, *C*, *B* respectively, and sub-figure (c) shows the final map after the robot arrived at *B* point. Our method and the conventional method show the significantly difference at the circular region labelled *X*. The conventional method once estimate *X* as ’free-space’ clearly when the robot was close to *X* (Figure 13(a)). When the robot re-observed *X* region from farther point, however, the detail information of *X* was missing and estimated as ’obstacle’ rather than free-space due to the erroneous stereo observation from far point. (Figure 13(b),(c)). In contrast, our method correctly estimated the free-space in *X* without compromise due to the erroneous observation from far point (Figure 14(a),(b),(c)).

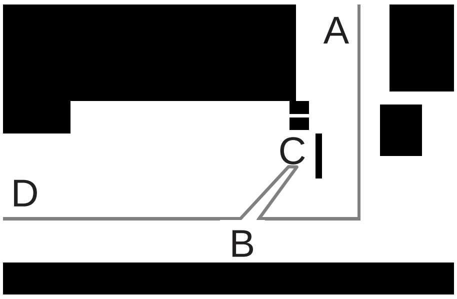
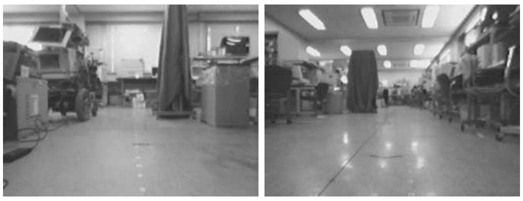


Figure 11: Rough map of obstacles.



(a) From A in Figure 11 (b) From D in Figure 11

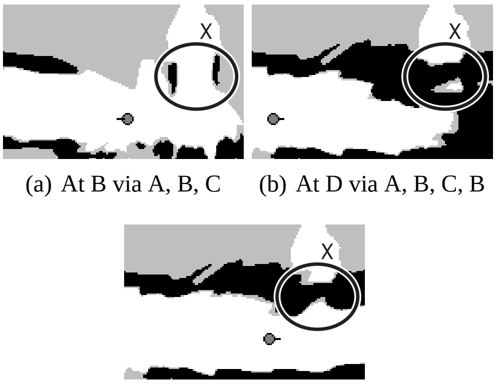
Figure 12: Robot’s view.

The process time for one update was about 1800ms on a PC with Athlon X2 4400+ CPU and 2GB memories.

4 CONCLUSIONS

We introduced a probabilistic observation model properly considering the sensor visibility. Based on estimating probabilistic visibility of each grid from the current viewpoint, likelihoods considering sensor visibility are estimated and then the posteriors of each map grid are updated by Bayesian update rule. For estimating grid visibility, Markov chain calculation based on spatial continuity is employed based on the knowledge of the average sizes of obstacles and free areas.

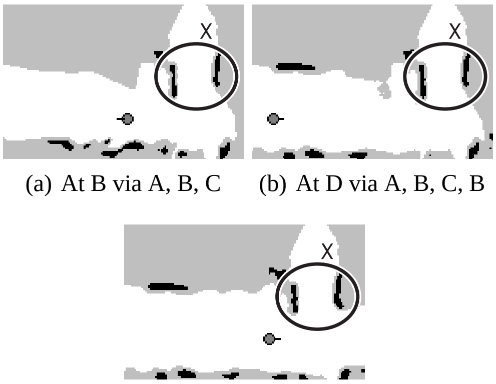
This paper concentrates the aim on showing the proof of concept of the probabilistic update considering sensor visibility and spatial continuity. For application to the real robot navigation, SLAM framework is necessary. In addition there are moving objects like human and semi-static objects like chairs or doors in the real indoor scene. The expansion to SLAM and environments with movable objects is the future works.



(a) At B via A, B, C (b) At D via A, B, C, B

(c) At B via A, B, C, B, D

Figure 13: Result of map update without considering visibility and spatial dependencies.



(a) At B via A, B, C (b) At D via A, B, C, B

(c) At B via A, B, C, B, D

Figure 14: Result of map update considering visibility and spatial dependencies.

REFERENCES

Grisetti, G., Stachniss, C., and Burgard, W. (2005). Improving Grid-based SLAM with Rao-Blackwellized Particle Filters by Adaptive Proposals and Selective Resampling. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2443–2448.

Miura, J., Negishi, Y., and Shirai, Y. (2002). Mobile Robot Map Generation by Integrationg Omnidirectional Stereo and Laser Range Finder. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 250–255.

Montemerlo, M. and Thrun, S. (2002). FastSLAM 2.0: An Improved Particle Filtering Algorithm for Simultaneous Localization and Mapping that Provably Converges.

Moravec, H. P. (1988). Sensor fusion in certainty grids for mobile robots. In *AI Magazine, Summer*, pages 61–74.

Rachlin, Y., Dolan, J., and Khosla, P. (2005). Efficient mapping through exploitation of spatial dependencies. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 3117–3122.

Suppes, A., Suhling, F., and Hötter, M. (2001). Robust Obstacle Detection from Stereoscopic Image Sequences Using Kalman Filtering. In *DAGM-Symposium*, pages 385–391.

Thrun, S. (2002a). Learning occupancy grids with forward sensor models. In *Autonomous Robots*.

Thrun, S. (2002b). Robotic Mapping: A Survey. In Lakemeyer, G. and Nebel, B., editors, *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann.

Thrun, S., Montemerlo, M., Koller, D., Wegbreit, B., Nieto, J., and Nebot, E. (2004). FastSLAM: An Efficient Solution to the Simultaneous Localization And Mapping Problem with Unknown Data Association. *Journal of Machine Learning Research.*

A NEWMARK FMD SUB-CYCING ALGORITHM

J. C. Miao [1,2], P. Zhu [1], G. L. Shi [1] and G. L. Chen [1]

School of Mechanical Engineering, Shanghai JiaoTong University, Shanghai, China[1]

Department of Mechanical Engineering, Shazhou institute of technology, Zhangjiagang, China[2]

miaojiancheng@sjtu.edu.edu, pzhu@sjtu.edu.cn, glshi@263.net, glchen@sjtu.edu.cn

Keywords: Flexible multi-body dynamics, Condensed FMD equation, Sub-cycling algorithm, Energy balance.

Abstract: This paper presents a Newmark method based sub-cycling algorithm, which is suitable for solving the condensed flexible multi-body dynamic (FMD) problems. Common-step update formulations and sub-step update formulations for quickly changing variables and slowly changing variables of the FMD are established. Stability of the procedure is checked by checking energy balance status. Examples indicate that the sub-cycling is able to enhance the computational efficiency without dropping results accuracy greatly.

1 INTRODUCTION

Flexible multi-body system (FMS) can be applied in various domains such as space flight, automobiles and robots. In these domains, accurate and efficient computation of the flexible bodies undergoing large overall motion is important for design and control of the system (Huang and Shao, 1996).

Conventional integration methods, such as the Newmark algorithm, the Runge-Kutta algorithm and the Gear algorithm and *etc* (Dan Negrut, et al, 2005), were widely applied to solve FMD equations.

Sub-cycling was proposed by Belytschko T. *et al* (Belytschko T.,1978). Mark *et al* (Mark, 1989) applied the method to simulate an impact problem and computational cost of the sub-cycling was only 15% of that without sub-cycling. Gao H *et al.* (Gao, 2005) used sub-cycling to simulate auto-body crashworthiness and declared that the cost is only 39.3% of no sub-cycling. Tamer *et al* (Tamer, 2003) pointed out that the FMD sub-cycling methods have not yet been presented in literatures.

2 SUB-CYCLING FOR FMD

A sub-cycling is constituted by two types of cycles, main-cycle and sub-cycle. The key for sub-cycling is to appropriately treat with interactions of nodes on the interface (Daniel, 1997).

2.1 Condensed Model of FMD

A flexible body is displayed in figure 1. Definition of variables can be referenced in the literature (Lu, 1996).

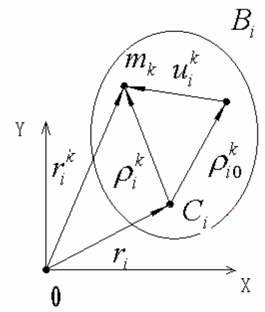


Figure 1: A spatial arbitrary flexible body.

The FMD equation can be established by means of the Lagrange formation.

$$\mathbf{M}\ddot{\mathbf{q}}+\mathbf{K}\mathbf{q}+\mathbf{C}_{\mathbf{q}}^{T}\boldsymbol{\lambda}=\mathbf{Q}_{\mathbf{F}}+\mathbf{Q}_{\mathbf{v}}\cdots(2-1)$$

$$\mathbf{C}(\mathbf{q},t)=0\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots(2-2)$$

Thereinto, **M** is a general mass matrix, **K** is a general stiffness matrix, **C**(**q**,t) is the constrains, $\mathbf{Q}_\mathbf{F}$ and $\mathbf{Q}_\mathbf{V}$ are general external load and general centrifugal load. Λ is the Lagrange multiplier. By means of two times of differentials of the constrain equation, an augmented FMD equation can be obtained as below.

$$\begin{bmatrix}\mathbf{M}_{\mathbf{r}} & \mathbf{C}_{\mathbf{q}}^{T}\\ \mathbf{C}_{\mathbf{q}} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\ddot{\mathbf{q}}\\ \boldsymbol{\lambda}\end{bmatrix}=\begin{bmatrix}\mathbf{Q}_{\mathbf{F}}^{*}\\ \mathbf{Q}_{\mathbf{C}}\end{bmatrix}\cdots\cdots\cdots(2-3)$$

Due to the constraints, variables in (2-3) are independent. By decomposition, a condensed FMD equation can be established as following (Haug, 1989).

$$\hat{\mathbf{M}}_i(\mathbf{q}_i,\mathbf{q}_d,t)\ddot{\mathbf{q}}_i=\hat{\mathbf{Q}}_i(\dot{\mathbf{q}}_i,\mathbf{q}_i,\dot{\mathbf{q}}_d,\mathbf{q}_d,t)\cdots(2-4)$$

Thereinto:

$$\hat{\mathbf{M}}_i=\mathbf{M}_{ii}-\mathbf{M}_{id}\mathbf{C}_{\mathbf{q}_d}^{-1}\mathbf{C}_{\mathbf{q}_i}-\mathbf{C}_{\mathbf{q}_d}^{T}\left(\mathbf{C}_{\mathbf{q}_d}^{-1}\right)^{T}\left[\mathbf{M}_{di}-\mathbf{M}_{dd}\mathbf{C}_{\mathbf{q}_d}^{-1}\mathbf{C}_{\mathbf{q}_i}\right]$$
$$\hat{\mathbf{Q}}_i=\mathbf{Q}_{\mathbf{F}i}^{*}-\mathbf{M}_{id}\mathbf{C}_{\mathbf{q}_d}^{-1}\boldsymbol{\gamma}-\mathbf{C}_{\mathbf{q}_i}^{T}\left(\mathbf{C}_{\mathbf{q}_d}^{-1}\right)^{T}\left[\mathbf{Q}_{\mathbf{F}d}^{*}-\mathbf{M}_{dd}\mathbf{C}_{\mathbf{q}_d}^{-1}\boldsymbol{\gamma}\right]$$

Equation（2-4）is a pure differential formation. It is suitable for sub-cycling (Lu, 1996).

2.2 Newmark Integration for FMD

The Newmark integration is as following. There, $\mathbf{q}_t$ denotes value of the general variable at time t. Δ t is the step size. β and γ are adjustable parameters.

$$\mathbf{q}_{t+\Delta t}=\mathbf{q}_t+\dot{\mathbf{q}}_t\Delta t+\frac{\Delta t^2}{2}\left[(1-2\beta)\ddot{\mathbf{q}}_t+2\beta\ddot{\mathbf{q}}_{t+\Delta t}\right]\cdots(2-5)$$
$$\dot{\mathbf{q}}_{t+\Delta t}=\dot{\mathbf{q}}_t+\Delta t\left[(1-\gamma)\ddot{\mathbf{q}}_t+\gamma\ddot{\mathbf{q}}_{t+\Delta t}\right]\cdots\cdots\cdots\cdots\cdots(2-6)$$

Define symbols below.

$$a_0=\frac{1}{\beta\Delta t^2}\quad a_1=\frac{\gamma}{\beta\Delta t}\quad a_2=\frac{1}{\beta\Delta t}\quad a_3=\frac{1}{2\beta}-1$$
$$a_4=\frac{\gamma}{\beta}-1\quad a_5=\frac{\Delta t}{2}\left(\frac{\gamma}{\beta}-2\right)\quad a_6=\Delta t(1-\gamma)\quad a_7=\gamma\Delta t$$

The dynamic equation at time t+Δ t can be established below.

$$\hat{\mathbf{M}}_i\left(\mathbf{q}_i^{t+\Delta t},\mathbf{q}_d^{t+\Delta t}\right)\left[a_0\left(\mathbf{q}_i^{t+\Delta t}-\mathbf{q}_i^{t}\right)-a_2\dot{\mathbf{q}}_i^{t}-a_3\ddot{\mathbf{q}}_i^{t}\right]$$
$$=\hat{\mathbf{Q}}_i\left({}^{(1)}\mathbf{q}_i^{t+\Delta t},{}^{(1)}\dot{\mathbf{q}}_i^{t+\Delta t},{}^{(1)}\mathbf{q}_d^{t+\Delta t},{}^{(1)}\dot{\mathbf{q}}_d^{t+\Delta t}\right)\cdots(2-7)$$

Finally, we can get the results.

$$\mathbf{q}_i^{t+\Delta t}=\left[a_0\hat{\mathbf{M}}_i^{t+\Delta t}\right]^{-1}\hat{\mathbf{Q}}_i^{t+\Delta t}+\mathbf{q}_i^{t}+\frac{a_2}{a_0}\dot{\mathbf{q}}_i^{t}+\frac{a_3}{a_0}\ddot{\mathbf{q}}_i^{t}\cdots(2-8)$$

Equation (2-8) need be solved iteratively. The iteration process is as following.

$${}^{(k)}\mathbf{q}_i^{t+\Delta t}=\left[a_0\,{}^{(k-1)}\hat{\mathbf{M}}_i^{t+\Delta t}\right]^{-1}{}^{(k-1)}\hat{\mathbf{Q}}_i^{t+\Delta t}$$
$$+\mathbf{q}_i^{t}+\frac{a_2}{a_0}\dot{\mathbf{q}}_i^{t}+\frac{a_3}{a_0}\ddot{\mathbf{q}}_i^{t}\cdots\qquad(2-9)$$

Thereinto, the top left corner marks represent the number of the iteration step. Start-up initialisation of iteration can be set-up below.

$${}^{(1)}\mathbf{q}_{t+\Delta t}=\mathbf{q}_t+\dot{\mathbf{q}}_t\Delta t+\frac{\Delta t^2}{2}\left[(1-2\beta)\ddot{\mathbf{q}}_t+2\beta^{(1)}\ddot{\mathbf{q}}_{t+\Delta t}\right]\cdots(2-10)$$

2.3 Newmark Sub-cycling for FMD

For simplification, all variables are separated into two categories. The smaller step size is set to be Δ t. The larger step size is set to be nΔ t. n is a positive integer. Thus, $\ddot{\mathbf{q}}$ is expressed as a decomposed format. And the condensed FMD formula can be decomposed as a block matrix format as following.

$$\begin{bmatrix}\hat{\mathbf{M}}_{LL}^{t} & \hat{\mathbf{M}}_{LS}^{t}\\ \hat{\mathbf{M}}_{SL}^{t} & \hat{\mathbf{M}}_{SS}^{t}\end{bmatrix}\begin{bmatrix}\ddot{\mathbf{q}}_L\\ \ddot{\mathbf{q}}_S\end{bmatrix}=\begin{bmatrix}\hat{\mathbf{Q}}_L^{t}\\ \hat{\mathbf{Q}}_S^{t}\end{bmatrix}\cdots(2-12)$$

The subscript symbol, L and S, represent the larger step size and the smaller step size. **M** and **Q** are the general mass matrix and the general external force. According to balance status of the FMS at t+Δ t and t+nΔ t, two groups of equations can be obtained as following.

$$\hat{\mathbf{M}}_{LLn}\ddot{\mathbf{q}}_{Ln}+\hat{\mathbf{M}}_{LSn}\ddot{\mathbf{q}}_{Sn}=\hat{\mathbf{Q}}_{Ln}\cdots(2-13)$$
$$\hat{\mathbf{M}}_{SL1}\ddot{\mathbf{q}}_{L1}+\hat{\mathbf{M}}_{SS1}\ddot{\mathbf{q}}_{S1}=\hat{\mathbf{Q}}_{S1}\cdots(2-14)$$

Thereinto, $\ddot{\mathbf{q}}_{Ln}$ and $\ddot{\mathbf{q}}_{Sn}$ are accelerations at $t+n\Delta t$. $\ddot{q}_{L1}$ and $\ddot{q}_{S1}$ are accelerations at $t+\Delta t$. The general mass matrix and the general external force are defined similarly. Define symbols below.

$$a_0^1=a_0,a_0^n=\frac{a_0^1}{n^2},a_2^1=a_2,a_2^n=\frac{a_2^1}{n},a_3^1=a_3,a_3^n=a_3^1$$

In order to compute interaction between the coupling variables at the common update, $\mathbf{q}_{Sn}$ and $\mathbf{q}_{L1}$ need be estimated simultaneously. Similar to the method (Daniel, 1997), $\mathbf{q}_{L1}$ can be linearly interpolated and $\mathbf{q}_{Sn}$ can be linearly extrapolated by means of the trapezoid rule. The formats of the interpolations are expressed below.

$$\mathbf{q}_{L1}=\frac{n-1}{n}\mathbf{q}_{L0}+\frac{1}{n}\mathbf{q}_{Ln}\cdots\cdots\cdots\cdots(2-15)$$
$$\mathbf{q}_{Sn}=n\left(\mathbf{q}_{Sn}-\mathbf{q}_{S0}\right)+\mathbf{q}_{S0}\cdots\cdots\cdots(2-16)$$

Hence action of the slowly changing variables to the rapidly changing variables is calculated below.

$$-\mathbf{F}_{SL1}=\hat{\mathbf{M}}_{SL1}\left[\frac{a_0^1}{n}\left(\mathbf{q}_{Ln}-\mathbf{q}_{L0}\right)-a_2^1\dot{\mathbf{q}}_{L0}-a_3^1\ddot{\mathbf{q}}_{L0}\right]\cdots(2-17)$$

The action of the rapidly changing variables to the slowly changing variables is calculated below.

$$-\mathbf{F}_{LSn}=\hat{\mathbf{M}}_{LSn}\left[na_0^n\left(q_{S_1}-\mathbf{q}_{S_0}\right)-a_2^n\dot{\mathbf{q}}_{S_0}-a_3^n\ddot{\mathbf{q}}_{S_0}\right]\cdots(2-18)$$

Imposing equations (2-17) and (2-18) into equations (2-13) and (2-14), and co-multiplying a number, *n*, to the left hand and the right hand of equation (2-13), the common update formula of the slowly changing variables and the rapidly changing variables can be obtained below.

$$a_0\begin{bmatrix}\hat{\mathbf{M}}_{SS1} & \hat{\mathbf{M}}_{Sl1}\\ \hat{\mathbf{M}}_{LSn} & \hat{\mathbf{M}}_{LLn}\end{bmatrix}\begin{bmatrix}\mathbf{q}_{S1}\\ \dfrac{\mathbf{q}_{Ln}}{n}\end{bmatrix}=\begin{bmatrix}\hat{\mathbf{Q}}_{S1}\\ n\hat{\mathbf{Q}}_{Ln}\end{bmatrix}+a_0\begin{bmatrix}\hat{\mathbf{M}}_{SS1} & \hat{\mathbf{M}}_{Sl1}\\ \hat{\mathbf{M}}_{LSn} & \hat{\mathbf{M}}_{LLn}\end{bmatrix}\begin{bmatrix}\mathbf{q}_{S0}\\ \dfrac{\mathbf{q}_{L0}}{n}\end{bmatrix}$$
$$+a_2\begin{bmatrix}\hat{\mathbf{M}}_{SS1} & \hat{\mathbf{M}}_{Sl1}\\ \hat{\mathbf{M}}_{LSn} & \hat{\mathbf{M}}_{LLn}\end{bmatrix}\begin{bmatrix}\dot{\mathbf{q}}_{S0}\\ \dot{\mathbf{q}}_{L0}\end{bmatrix}+a_3\begin{bmatrix}\hat{\mathbf{M}}_{SS1} & \hat{\mathbf{M}}_{Sl1}\\ n\hat{\mathbf{M}}_{LSn} & n\hat{\mathbf{M}}_{LLn}\end{bmatrix}\begin{bmatrix}\ddot{\mathbf{q}}_{S0}\\ \ddot{\mathbf{q}}_{L0}\end{bmatrix}\cdots(2-19)$$

In order to get the sub-step update formula, the following equation need be calculated at *t*+（*i*+1）Δ *t* , *i*=1,2*...(n-2*).

$$\hat{\mathbf{M}}_{SL(i+1)}\ddot{\mathbf{q}}_{L(i+1)}+\hat{\mathbf{M}}_{SS(i+1)}\ddot{\mathbf{q}}_{S(i+1)}=\hat{\mathbf{Q}}_{S(i+1)}\cdots(2-20)$$

Also, according to equation (2-7), the following equation can be obtained.

$$\hat{\mathbf{M}}_{SL(i+1)}\left[a_0^1\left(\mathbf{q}_{L(i+1)}-\mathbf{q}_{Li}\right)-a_2^1\dot{\mathbf{q}}_{Li}-a_3^1\ddot{\mathbf{q}}_{Li}\right]+$$
$$\hat{\mathbf{M}}_{SS(i+1)}\left[a_0^1\left(\mathbf{q}_{S(i+1)}-\mathbf{q}_{Si}\right)-a_2^1\dot{\mathbf{q}}_{Si}-a_3^1\ddot{\mathbf{q}}_{Si}\right]=\hat{\mathbf{Q}}_{S(i+1)}\cdots(2-21)$$

In equation (2-21), the slowly changing variables, which are used to compute the interaction of the coupling variables, can be linearly interpolated in terms of the trapezoid rule.

$$\mathbf{q}_{Li}=\frac{n-i}{n}\mathbf{q}_{L0}+\frac{i}{n}\mathbf{q}_{Ln}\cdots\cdots\cdots\cdots\cdots(2-22)$$

$$\ddot{\mathbf{q}}_{L_i}=\frac{n-i}{n}\ddot{\mathbf{q}}_{L0}+\frac{i}{n}\ddot{\mathbf{q}}_{Ln}\cdots\cdots\cdots\cdots\cdots(2-23)$$

In terms of equations (2-22) and (2-23), the action of the slowly changing variables to the rapidly changing variables can be approximately assessed.

$$-\mathbf{F}_{SL(i+1)}=\frac{1}{n}\hat{\mathbf{M}}_{SL(i+1)}\left[\left(a_0\left(\mathbf{q}_{Ln}-\mathbf{q}_{L0}\right)-a_2\left(\left(n-i\right)\dot{\mathbf{q}}_{L0}+i\dot{\mathbf{q}}_{Ln}\right)\right.\right.$$
$$\left.-a_3^1\left(\left(n-i\right)\ddot{\mathbf{q}}_{L0}+i\ddot{\mathbf{q}}_{Ln}\right)\right]\cdots\cdots\cdots\quad\cdots\cdots\cdots(2-24)$$

Imposing equation (2-24) into equation (2-21), the sub-step update format can be got as following.

$$a_0\hat{\mathbf{M}}_{SS(i+1)}\mathbf{q}_{S(i+1)}=\hat{\mathbf{Q}}_{S(i+1)}+\hat{\mathbf{M}}_{SS(i+1)}\left(a_0\mathbf{q}_{Si}+a_2\dot{\mathbf{q}}_{Si}+a_3\ddot{\mathbf{q}}_{Si}\right)$$
$$-\frac{a_0}{n}\hat{\mathbf{M}}_{SL(i+1)}\left(\mathbf{q}_{Ln}-\mathbf{q}_{L0}\right)+\frac{a_2}{n}\hat{\mathbf{M}}_{SL(i+1)}\left(\left(n-i\right)\dot{\mathbf{q}}_{L0}+i\dot{\mathbf{q}}_{Ln}\right)$$
$$+\frac{a_3^1}{n}\hat{\mathbf{M}}_{SL(i+1)}\left(\left(n-i\right)\ddot{\mathbf{q}}_{L0}+i\ddot{\mathbf{q}}_{Ln}\right)\cdots\cdots\cdots\cdots\quad\cdots(2-25)$$

The energy balance status can be calculated by means of the equation below (Mark and Belytschko, 1989).

$$\left|W_n^{kin}+W_n^{\text{int}}-W_n^{ext}\right|\le\delta\left\|W\right\|\cdots\cdots(2-26)$$

Thereinto, W_n^{ext} is the work of the external forces at *n*Δ *t*. W_n^{int} is the internal energy at *n*Δ *t*. W_n^{kin} is the kinetic energy of the system at *n*Δ *t*.δ is the available error coefficient.

3 NUMERICAL EXAMPLES

In this section, two numerical examples will be performed to validate availability and efficiency of the present sub-cycling algorithm.

3.1 A Bar-slider System

A bar-slider system is shown in figure 2. Mass of the rod is 2.0 kilograms and mass of the slider is 5.0 kilograms. The driving torque is 100 Nm/s. Stiffness of the spring is 1000 N/m. length of the rigid rod is 2 meters.

Results of rotational angle of the rod, the vibration amplification of the sliding block and the energy balance status computed by means of the sub-cycling and without sub-cycling are shown in figure 3 to figure 6. The scale values in brackets of the figure captions, such as (5:1), represent a sub-cycling with 5 sub-steps in one major step.

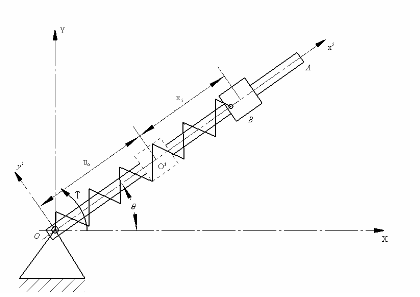


Figure 2: Rotational rod-spring-slider system.

Comparing the results in figure 3 to figure 6, we can see that no matter which scale of the sub-cycling is adopted, the results are very similar. The error of sub-cycling with scale 10:1 is a little larger than that of sub-cycling with scale 5:1. Yet all these two errors are very small if compare with the original results. The time cost of sub-cycling with scale 5:1 is 118 seconds and the time cost of the original algorithm without sub-cycling is 218 seconds. The proportion of time cost of these two algorithms is 54%. The time cost of sub cycling with scale 10:1 is 44 seconds and the proportion of time cost is only 20.2%. The results of balance checking illustrate that the sub-cycling is stable during the integral process.

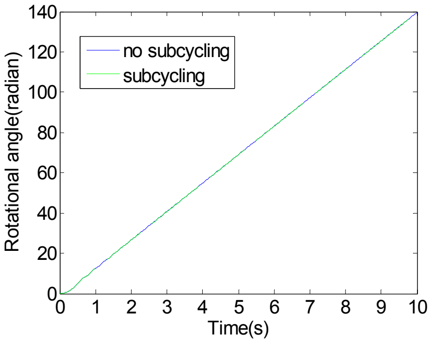


Figure 3: Rotational angle of the bar (5:1).

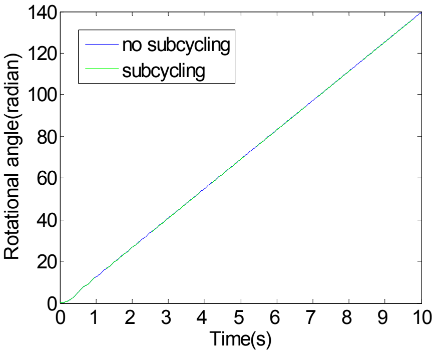


Figure 4: Rotational angle of the bar (10:1).

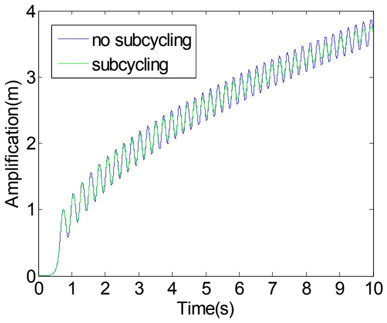


Figure 5: Vibration amplification of the slider (5:1).

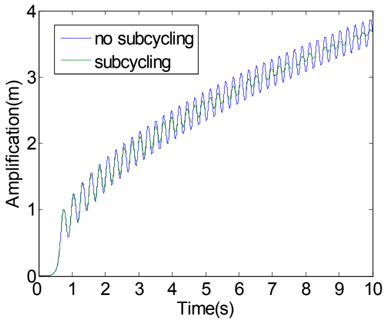


Figure 6: Vibration amplification of the slider (10:1).

3.2 Airscrew of a Jet Engine

FEM model of the airscrew of a jet engine are displayed in figure 7. Parameters of the model are as following (Units: *kg/N/mm/s*). The material of airscrew is aluminium alloy. EX=7e6, PR=3, DEN=2.78e-6. Diameter of the airscrew is D=900 *mm*, rotate speed is 8000 *rpm.*

We simulate the large range overall motion of the airscrew by means of the Newmark sub-cycling and the original Newmark respectively. The dynamic stress at the blade root is described in figure 8. The time cost of the Newmark sub-cycling is 1660 seconds and that of the Newmark is 2810 seconds. The computational efficiency is enhanced about 70%. The compared results show the good precision and stability of the Newmark sub-cycling.

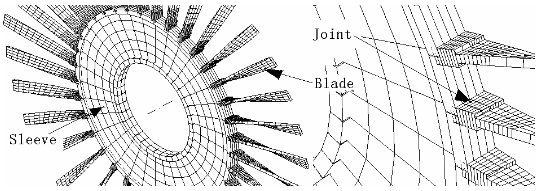


Figure 7: The FEM model and the local mesh.

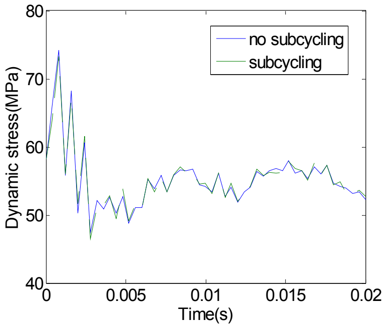


Figure 8: Dynamic stress of the blade root during rotation.

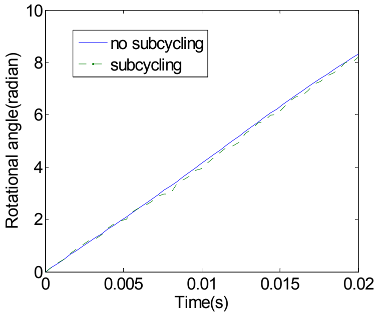


Figure 9: Rotational angle of the blade root.

4 CONCLUSIONS

This paper firstly presents a Newmark-based FMD sub-cycling algorithm. By modifying the Newmark integral formula to be fitted for sub-cycling of the FMD problems, not only the integral efficiency can be greatly improved, but also be more easy for convergence of the integral process.

Because of that different integral step sizes are adopted during the sub-cycling, the integral process can be more efficient and easier for convergence. At the same time, Unconditional stability of the original Newmark are still kept in the Newmark sub-cycling.

The number of the sub-step in one major cycle can be a little infection of the integral precision of a sub-cycling process. However, it is unobvious as the number is within a range. Generally speaking, the enhancement of the integral efficiency is more significant when the number is under a limitation.

By checking the energy balance status of the integral process real time and adjusting the step size when necessary, the sub-cycling procedure can keep a well convergence property and obtain the reasonable numerical computation results.

REFERENCES

Huang Wenhu, Shao Chengxun, 1996. *Flexible multi-body dynamics*, The science Press.

Dan Negrut, Jose L, Ortiz., 2005. On an approach for the linearization of the differential algebraic equations of multi-body dynamics. *Proceedings of IDETC/MESA. DETC2005-85109. 24-28.*Long Beach, USA.

Dan Negrut, Gisli Ottarsson, 2005. On an Implementation of the Hilber-Hughes-Taylor Method in the Context of Index 3 Differential-Algebraic Equations of Multibody Dynamics. *DETC2005-85096.*

Belytschko T., Mullen, Robert, 1978. Stability of explicit-implicit mesh partitions in time integration. *International Journal for Numerical Methods in Engineering*, 12(10): 1575-1586.

Neal, Mark O., Belytschko T., 1989. Explicit-explicit sub-cycling with non-integer time step ratios for structural dynamic systems. *Computers and Structures*, 31(6): 871-880.

Gao H.,Li G. Y.,Zhong Z. H., 2005. Analysis of sub-cycling algorithms for computer simulation of crashworthiness. *Chinese journal of mechanical engineering*, 41(11): 98-101.

Tamer M Wasfy, Ahmed K Noor, 2003. Computational strategies for flexible multi-body systems. *Applied Mechanics Reviews*. 56(6): 553-613.

W. J. T. Daniel, 1997. Analysis and implementation of a new constant acceleration sub-cycling algorithm. *international journal for numerical methods in engineering*. 40: 2841-2855.

Haug, E. J., 1989. *Computer-Aided Kinematics and Dynamics of Mechanical Systems*. Prentice-Hall. Englewood Cliffs, NJ.

Lu Youfang, 1996. *Flexible multi-body dynamics*. The higher education Press.

ROBOT TCP POSITIONING WITH VISION

Accuracy Estimation of a Robot Visual Control System

Drago Torkar and Gregor Papa
Computer Systems Department, Jožef Stefan Institute, Jamova c. 39, SI-1000 Ljubljana, Slovenia
drago.torkar@ijs.si, gregor.papa@ijs.si

Keywords: Robot control, Calibrated visual servoing.

Abstract: Calibrated 3D visual servoing has not fully matured as a industrial technology yet, and in order to widen its use in industrial applications its technological capability must be precisely known. Accuracy and repeatability are two of the crucial parameters in planning of any robotic task. In this paper we describe a procedure to evaluate the 2D and 3D accuracy of a robot stereo vision system consisting of two identical 1 Megapixel cameras, and present the results of the evaluation.

1 INTRODUCTION

In the last decades, more and more robots applications were used in industrial manufacturing which was accompanied by an increased demand for versatility, robustness and precision. The demand was mostly satisfied by increasing the mechanical capabilities of robot parts. For instance, to meet the micrometric positioning requirements, stiffness of the robot's arms was increased, high precision gears and low backlash joints introduced, which often led to difficult design compromises such as the request to reduce inertia and increase stiffness. This results in approaching the mechanical limits and increased cost of robots decreasing the competitiveness of the robot systems on the market (Arflex, 2005).

Lately, the robot producers have put much effort into incorporating visual and other sensors to the actual industrial robots thus providing a significant improvement in accuracy, flexibility and adaptability. Vision is still one of the most promising sensors (Ruf and Horaud, 1999) in real robotic 3D servoing issues (Hutchinson et al., 1995). It has been vastly investigated for the last two decades in laboratories but it's only now that it finds its way to industrial implementation (Robson, 2006) in contrast to machine vision which became a well established industry in the last years (Zuech, 2000). There are many reasons for this. The vision systems used with the robots must satisfy a few constraints that differ them from a machine vision measuring systems. First of all, the camera working distances are much larger, especially with larger robots that can reach several meters. Measuring at such distances with high precision requires much higher resolution which very soon reaches its technological and price limits. For example, nowadays 4 Megapixel cameras are the state of the art in vision technology but are not affordable in many robotic applications since their price almost reaches the robot price. The dynamics of the industrial processes requires high frame rates which in connection with real time processing puts another difficult constraint on system integrators. The unstructured industrial environment with changing environmental lighting is another challenge for the robot vision specialists.

When designing the vision system within robot applications it is very important to choose the optimal equipment for the task and to get maximal performance out of each component. In the paper we represent a procedure for the precision estimation of a calibrated robot stereo vision system in 2D and 3D environment. Such a system can be used in visual servoing applications for precise tool center point (TCP) positioning.

2 METHODOLOGY

Four types of accuracy tests were performed: a static 2D test, a dynamic 2D test, a static 3D test, and a dynamic 3D test. Throughout all the tests, an array of 10 infrared light emitting diodes (IR-LED) was used to establish its suitability for being used as a marker and a calibration pattern in the robot visual servoing applications.

Within the static 2D test, we were moving the IR-LED array with the linear drive perpendicular to the camera optical axes and measured the increments in the image. The purpose was to detect the smallest linear response in the image. The IR-LED centroids were determined in two ways: on binary images and on grey-level images as centers of mass. During image grabbing the array did not move thus eliminating any dynamic effects. We averaged the movement of centroids of 10 IR-LEDs in a sequence of 16 images and calculated the standard deviation to obtain accuracy confidence intervals. With the dynamic 2D test shape distorsions in the images due to fast 2D movements of linear drive were investigated. We compared a few images of IR-LED array taken during movement to statically obtained ones which provided information of photocenter displacements and an estimation of dynamic error.

We performed the 3D accuracy evaluation with 2 fully calibrated cameras in a stereo setup. Using again the linear drive, the array of IR-LEDs was moved along the line in 3D space with different increments and the smallest movement producing a linear response in reconstructed 3D space was sought. In the 3D dynamic test, we attached the IR-LED array to the wrist of an industrial robot, and dynamically guided it through some predefined points in space and simultaneously recorded the trajectory with fully calibrated stereo cameras. We compared the reconstructed 3D points from images to the predefined points fed to robot controller.

3 TESTING SETUP

The test environment consisted of:

- PhotonFocus MV-D1024-80-CL-8 camera with CMOS sensor and framerate of 75 fps at full resolution (1024x1024 pixels),
- Active Silicon Phoenix-DIG48 PCI frame grabber,
- Moving object (IR-LED array) at approximate distance of 2m. The IR-LED array (standard deviation of IR-LED accuracy is below 0.007 pixel, as stated in (Papa and Torkar, 2006)) fixed to Festo linear guide (DGE-25-550-SP) with repetition accuracy of +/-0.02mm.

For then static 2D test the distance from camera to a moving object (in the middle position) that moves perpendicularly to optical axis was 195cm; camera field-of-view was 220cm, which gives pixel size of 2.148mm; Schneider-Kreuznach lens CINEGON 10mm/1,9F with IR filter; exposure time was 10.73ms, while frame time was 24.04ms, both obtained experimentally.

For the dynamic 2D test conditions were the same as in static test, except the linear guide was moving the IR-LED array with a speed of 460mm/s and the exposure time was 1ms.

In the 3D reconstruction test the left camera distance to IR-LED array and right camera distance to IR-LED array were about 205cm; baseline distance was 123cm; Schneider-Kreuznach lens CINEGON 10mm/1,9F with IR filter; Calibration region-of-interest (ROI): 342 x 333 pixels; Calibration pattern: 6 x 8 black/white squares; Calibration method (Zhang, 1998); Reconstruction method (Faugeras, 1992). The reconstruction was done off-line and the stereo correspondence problem was considered solved due to a simple geometry of the IR-LED array and is thus not addressed here.

For the 3D dynamic test, an ABB industrial robot IRB 140 was used with the standalone fully calibrated stereo vision setup placed about 2m away from its base and calibrated the same way as before. The robot wrist was moving through the corners of an imaginary triangle with side length of approximately 12cm. The images were taken dynamically when the TCP was passing the corner points and reconstructed in 3D with an approximate speed of 500mm/s. The relative length of such triangle sides were compared to the sides of a statically-obtained and reconstructed triangle. The robot native repeatability is 0.02 mm and its accuracy is 0.01mm.

4 RESULTS

4.1 2D Accuracy Tests

The results of the evaluation tests are given below. Tests include the binary and grey-level centroids. For each movement increment the two figures are presented, as described below.

Pixel difference between the starting image and the consecutive images (at consecutive positions) – for each position the value is calculated as the average displacement of all 10 markers, while their position is calculated as the average position in the sequence of the 16 images grabbed at each position in static conditions. The lines in these figures should be as straight as possible.

The 0.01mm, 0.1mm, and 1mm increments for 2D tests are presented in Figure 1.

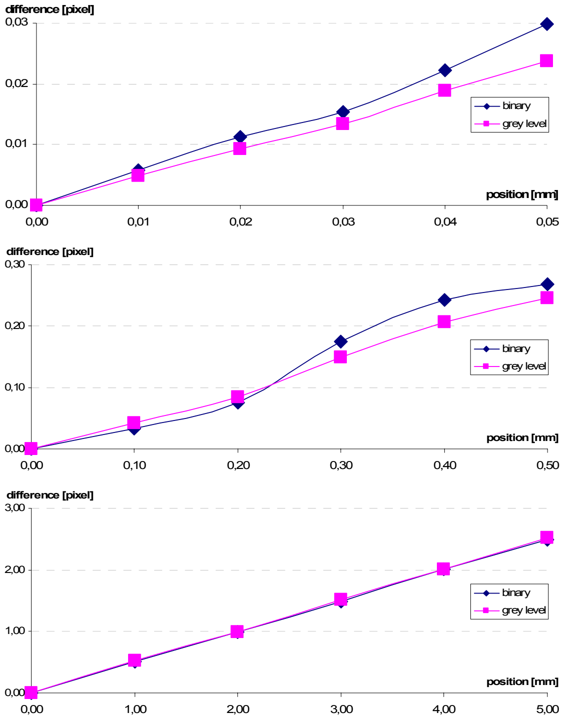


Figure 1: Pixel difference for 0.01mm (top), 0.1mm (middle), and 1mm (bottom) increments.

Figure 2 compares normalized pixel differences in grey-level images of a single marker.

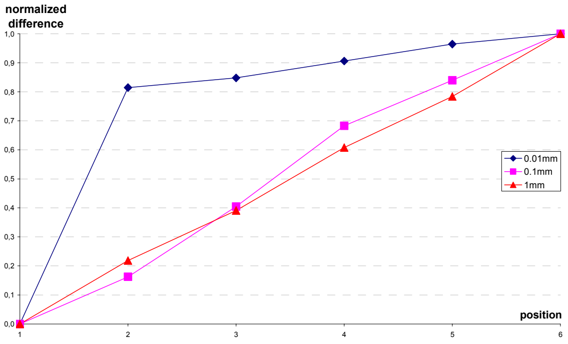


Figure 2: Normalized differences of grey-level images for each position comparing different increments.

A linear regression model was applied to measured data, and the R^2 values calculated to asses the quality of fit. The results are presented in Table 1 for 2D tests and in Table 2 for 3D tests. The R^2 value can be interpreted as the proportion of the variance in y attributable to the variance in x (see Eqn. 1), where 1 stands for perfect matching (fit) and a lower value denotes some deviations.

$$R^2 = \left(\frac{\sum (x-\bar{x})(y-\bar{y})}{\sqrt{\sum (x-\bar{x})^2 (y-\bar{y})^2}} \right)^2 \qquad (1)$$

Considering the R^2 threshold of 0.994 we were able to detect increments of the moving object in the range of 1/5 of a pixel. The value of the threshold is set to the value that gives a good enough approximation of the linear regression model, to ensure the applicable results of the measurements.

Table 1: Comparison of standard deviations and R^2 values for different moving increments in 2D.

increments [mm]	standard deviation[mm]		R^2	
	binary	grey-level	binary	grey-level
0.01	0.045	0.027	0.4286	0.6114
0.1	0.090	0.042	0.8727	0.9907
1	0.152	0.069	0.9971	0.9991

The dynamic 2D test showed that when comparing the centers of the markers of the IR-LED array and the pixel areas of each marker in statically and dynamically (linear guide moving at full speed) grabbed images there is a difference in center positions and also the areas of markers in dynamically grabbed images are slightly larger than those of statically grabbed images.

Table 2 presents the differences of the centers of the markers, and difference in sizes of the markers of the statically and dynamically grabbed images.

Table 2: Comparison of the images grabbed in static and dynamic mode.

	X	Y	width	height	area
static	484.445	437.992	6	6	27
dynamic	484.724	437.640	7	6	32

Regarding the results presented in Table 2, the accuracy of the position in direction *x* of dynamically grabbed images comparing to statically grabbed is in the range of 1/3 of a pixel, due to the gravity centre shift of pixel area of marker during the movement of the linear guide.

4.2 3D Reconstruction Tests

We tested the static relative accuracy of the 3D reconstruction of the IR-LED array movements by linear drive. The test setup consisted of the two calibrated Photonfocus cameras focused on the IR-

LED array attached to the linear drive which exhibited precise movements of 0.01mm, 0.1mm and 1mm. The mass centre points of 10 LEDs were extracted in 3D after each movement and relative 3D paths were calculated and compared to the linear drive paths. Only grey-level images were considered, due to the better results obtained in 2D tests, as stated in Figure 2 and in Table 1. The 0.01mm, 0.1mm, and 1mm increments for the 3D tests are presented in Figure 3.

The accuracy in 3D is lower than in the 2D case, due to calibration and reconstruction errors, and according to the tests performed it is approximately 1/2 of a pixel.

Table 4 presents the results of the 3D dynamic tests where the triangle area and side lengths *a, b* and *c*, reconstructed from dynamically-obtained images were compared to static reconstruction of the same triangles. 10 triangles were compared, each formed by a diode in IR-LED array. The average lengths and the standard deviations are presented.

Table 3: Comparison of standard deviations and R^2 values for different moving increments in 3D.

increments [mm]	standard deviation [mm]	R^2
0.01	0.058	0.7806
0.1	0.111	0.9315
1	0.140	0.9974

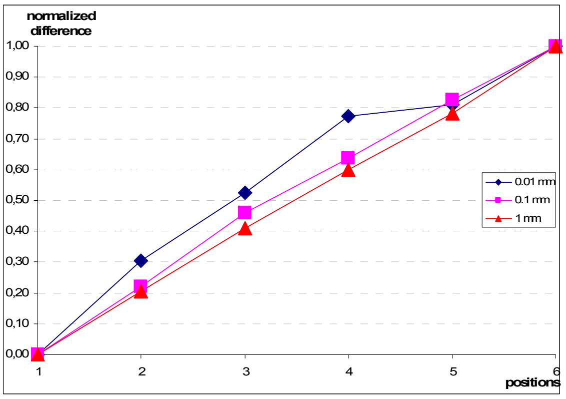


Figure 3: Pixel difference in the 3D reconstruction.

Table 4: comparison of static and dynamic triangles. All measurements are in mm.

	$\overline{a}$	σ	$\overline{b}$	σ	$\overline{c}$	σ
static	193.04	12.46	89.23	2.77	167.84	12.18
dynamic	193.51	12.43	89.03	2.77	167.52	12.03

We observe a significant standard deviation (up to 7%) of triangle side lengths which we ascribe to lens distortions since it is almost the same in the dynamic and in the static case. The images and the reconstruction in dynamic conditions vary only a little in comparison to static ones.

5 CONCLUSIONS

We performed the 2D and 3D accuracy evaluation of the 3D robot vision system consisting of 2 identical 1 Megapixel cameras. The measurements showed that the raw static 2D accuracy (without any subpixel processing approaches and lens distortion compensation) is confidently as good as 1/5 of a pixel. However, this is reduced to 1/2 of a pixel when image positions are reconstructed in 3D due to reconstruction errors.

In the dynamic case, the comparison to static conditions showed that no significant error is introduced with moving markers in both, 2D and 3D environment. For the speed level of an industrial robot the accuracy is though not reduced significantly.

ACKNOWLEDGEMENTS

This work was supported by the European 6th FP project Adaptive Robots for Flexible Manufacturing Systems (ARFLEX, 2005-2008) and the Slovenian Research Agency programme Computing structures and systems (2004-2008).

REFERENCES

Arflex European FP6 project official home page: http://www.arflexproject.eu

Faugeras, O., 1992. *Three-Dimensional Computer Vision: A geometric Viewpoint*, The MIT Press.

Hutchinson, S., Hager, G., Corke, P.I., 1995. A tutorial on visual servo control, *Yale University Technical Report*, RR-1068.

Papa, G., Torkar, D., 2006. Investigation of LEDs with good repeatability for robotic active marker systems, *Jožef Stefan Institute technical report*, No. 9368.

Robson, D., 2006. Robots with eyes, *Imaging and machine vision-Europe*, Vol. 17, pp. 30-31.

Ruf, A., Horaud, R., 1999. Visual servoing of robot manipulators, Part I: Projective kinematics, *INRIA technical report*, No. 3670.

Zhang, Z., 1998. A flexible new Technique for Camera Calibration, *Technical report*, MSRTR-98-71.

Zuech, N., 2000. *Understanding and applying machine vision*, Marcel Dekker Inc.

MULTI AGENT-BASED ON-LINE DIAGNOSTIC SCHEME OF SUBSTATION IED

Seong-Jeong Rim and Seung-Jae Lee
Next-generation Power Technology Center(NPTC), Myongji University, Yongin, Korea
rimsj@dreamwiz.com, sjlee@mju.ac.kr

Keywords: Multi-agents, On-Line Diagnosis, SA (substation automation), IED (intelligent electronic device).

Abstract: This paper presents a novel multi agent-based diagnostic scheme of substation IED. The proposed method is the on-line diagnostic scheme for the IEDs in IEC61850-based SA system using multi-agents. Multi-agents detect the fault of IED installed in substation and apply to improvement of reliability in protection and control system of substation. These agents in the SA system embedded in the station unit, the engineering unit, the trouble manager, backup IED and protection and control IED as the thread form. Through the implementation and the case studies of the proposed scheme, the availability is proven.

1 INTRODUCTION

IED (intelligent electronic device) mis-operations, such as failing to isolate a faulted line or incorrectly tripping a healthy line, are involved in major disturbances within power systems and can lead to catastrophic cascading events or even blackouts (Bertsch, J. *et al.* 2003). Giving IEDs the ability to diagnose would be a significant addition at critical locations in the power system by helping to reduce the number of mis-operations.

The existing diagnostic method of IED mainly uses off-line TBM (time-based maintenance) or CBM (condition-based maintenance) (Yu X.Z. et al, 2000). Especially IED doesn't operate in normal condition and operate it when the disturbance occurs in power system. It is hard to determine the fault condition before IED is tested.

On the other hand, the conventional scheme to improve the IED's reliability, have the redundant structure as the backup device (Ding M. et al., 2004). Redundant structure employs the dual device with the same function so that improves the reliability of the system. However, this system has the drawback which not only increases the probability of mis-operation but cost. Thus, the improvement scheme of the reliability is required to overcome these problems.

In this paper, a novel multi agent-based diagnostic scheme of substation IED is proposed. This method is the ODS (on-line diagnostic scheme) for the IEDs in IEC61850-based SA system using multi-agents (IEC 61850, 2002-2005). Multi-agents detect the fault of IED installed in substation and apply to improvement of reliability in protection and control system of substation. These agents in the SA system embedded in the station unit, the engineering unit, the trouble manager, backup IED and protection and control IED as the thread form. Through the implementation and the case studies of the proposed scheme, the availability is proven.

2 IEC61850-BASED SUBSTATION AUTOMATION

The scheme presented in this paper used many features of IEC61850 to diagnose the substation IED. The typical features of IEC61850, which make our scheme possible, consists of the network based data transfer (process bus), the standardized interface and SCL (Substation Configuration Language)-based engineering.

2.1 Process Bus

In the existing substation protection and control system, analogue data such as voltage and current are fed into protection relays from CT and VT which are installed on the primary protection equipments such as transformers and circuit-breakers. In the

IEC61850 based SA system, analogue signal will be converted into digital data by the process IED installed on the primary protection equipment and be sent to the bay IED through the process bus (Fig.1). The largest difference between the two schemes above is whether the data transfer path can be changed during operation. In the existing system, analogue data are transferred through hardware, so data transfer path cannot be changed unless the physical connection is changed. In this scheme, data are transferred on process bus, so the data transfer path can be changed by changing the destination of data packet.

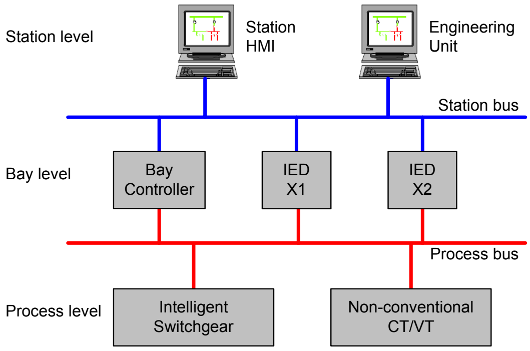


Figure 1: Structure of IEC61850-based SA System.

2.2 SCL-Based Engineering

For integration of various IEDs, IEC61850 suggests a standard engineering technique. The key is in two softwares called system configurator and IED configurator. Figure 2 shows the course of IEC61850 engineering. First, system configurator gets all the information about the substation from the SSD (System Specification Description) file that contains system related information and the ICD (IED Capability Description) file that contains IED related information, and creates the SCD (Substation Configuration Description) file that configures the function and data flow for each IED. All the files mentioned above are made by XML-based SCL. An IED configurator receives the SCD file and creates CID (Configured IED Description) file that contains the format suitable for IED. Finally to download CID file, each IED is able to communicate with among them.

In this scheme, when an IED error is detected, it reconfigures its data path so that the data that were transferred to bay IED is now transferred to the backup IED. The reconfiguration is feasible because of the SCL based engineering.

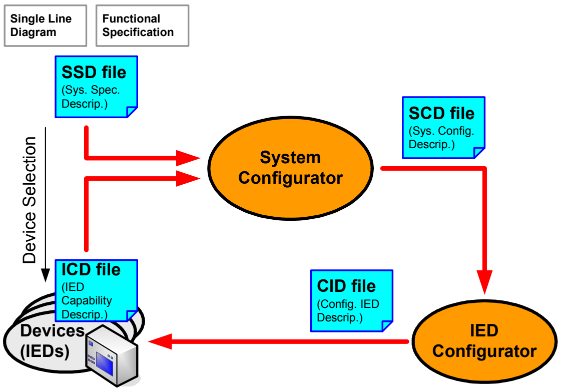


Figure 2: SCL based engineering.

2.3 Standardized Interface

In IEC61850, all the information about a substation is modelled into a standard interface called LN (Logical Node). Data exchange between these IEDs is possible as long as all these IEDs follow the standard LN interface regardless of IED hardware structure or algorithm. For example, if there are two distance relays made by different companies, whose signal filtering, calculation of RMS, error detection algorithm, and hardware are different, the two relays can be exchanged when they use the same services and the same LN interface.

3 MULTI AGENT-BASED ON-LINE DIAGNOSTIC SCHEME

3.1 On-Line Diagnostic Scheme

The structure of the ODS (on-line diagnostic scheme) is shown in Fig. 3. Note that the station unit, bay IED, process IED, and engineering unit constitute the substation automation system, which follows IEC61850 standard. Backup IED and trouble manager are added to the system to implement the ODS. Trouble manager is in charge of detecting the IED error by sending a test pattern to bay IED and comparing the results with expected ones. Backup IED is a redundant IED that downloads software from engineering unit and then starts to operate when error is detected on bay IED.

The TM (trouble manager) periodically checks the functions of each bay IED to determine whether the IED has an error. The fault detection procedure in trouble manager is as follows:

1) EU (engineering unit) assigns the DUT (device under test) and inform to TM.
2) TM transfers the information such as LN, preco nfigured control block and settings for the DUT to backup IED.
3) TM starts the sample value path from process IE D #1 to backup IED.
4) TM stops the sample value path from process IE D to DUT.
5) TM sends the test input pattern to DUT via proc ess bus.
6) DUT sends the response for the test input patter n to TM.
7) TM notifies the fault status of the DUT to statio n HMI if TM detects the abnormal status of DU T.

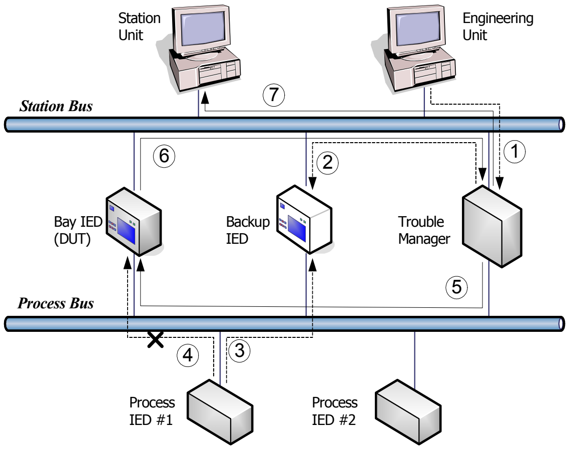


Figure 3: Structure of ODS.

If no fault is detected on DUT, TM requires EU to reconfigure the system to recover the original operation of DUT and then continue to check next IED. Also if fault is detected, system waits for repairing. Backup IED will replace the failed DUT until it is repaired.

3.2 Structure of Multi Agent-Based ODS

The proposed method uses several agents which distributed in a protection/control IED. This method detects the IED fault and replaces IED to be tested. In order to operate efficiently, these distributed systems, agents have the functionality which autonomously interacts between them. Figure 4 shows the proposed multi agent-based ODS structure.

As shown in figure 4, the multi agent-based ODS structure consists of data source, data storage, communication mechanism and data consumer. When agents detect the fault of IED in protection system, it replaces faulted IED and operates with backup IED properly.

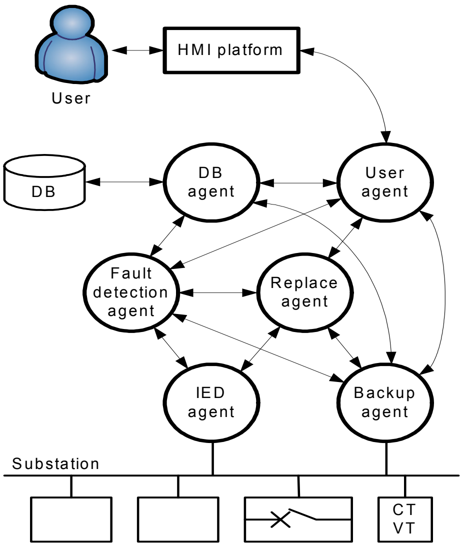


Figure 4: The proposed multi agent-based ODS structure.

Table 1 list a types and functionality of multi-agents required for performing the multi agent-based ODS.

Table 1: Types and functionality of agents.

Types	Duty	Ability
Fault detection	Detect the fault of DUT	Send the test pattern, Create the expected result, Compare with actual result
Replace	Replace DUT with backup IED	Control the communication service
DB	Response the required information	Add the information to DB, Response for query
User	Assign the IED to be tested and display the fault	Display the information, Transfer the query
Backup	Perform function of the DUT	Perform the function of DUT
IED	Perform function of protection/control IED and communicate with other agents	Response for the request

The agents listed in above table communicate each other with MMS (Manufacturer Message Specification) instead of ACL (Agent Communication Language), since the agents embedded in IEC61850-based SA system.

4 CASE STUDY

4.1 Test System

Figure 5 shows the test system to verify the performance the multi agent-based ODS. The part inside dotted line in this figure means the additional equipments to conventional SA system.

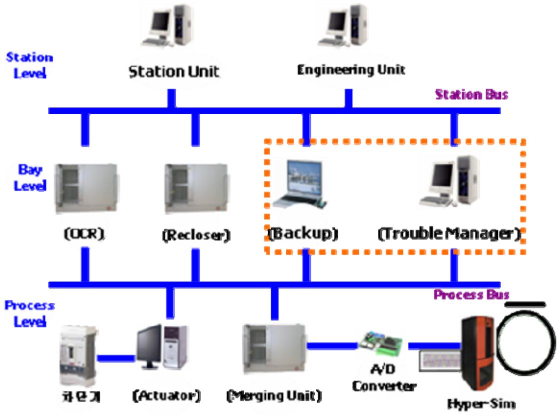


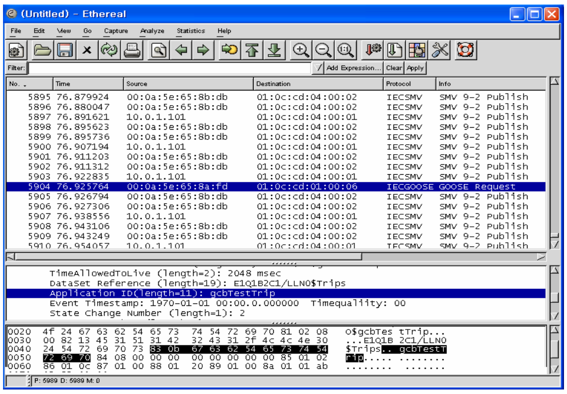
Figure 5: Test system.

In the test system, HyperSim (real-time simulator) uses to generate the fault in power system (HyperSim). MU (merging unit) transfers the digital signal to protection IED (OCR/Recloser) via network after it had A/D conversion. Actuator has the function such as on/off of circuit breaker as process IED. Backup IED performs the same function of DUT.

4.2 Simulation: DUT Diagnosis

In this simulation, we consider two types of fault such as communication and IED hardware part. We assume that the fault in communication part is LAN cable broken and the fault in IED hardware part is power off.

We perform the test that TM can detect the fault. After the DUT's function replaces with backup IED, TM sent the test input pattern and received the response from DUT. Figure 6 shows the GOOSE message from DUT as response.



```
No.   Time       Source             Destination        Protocol  Info
 5895 76.879924  00:0a:5e:65:8b:db  01:0c:cd:04:00:02  IECSMV    SMV 9-2 Publish
 5896 76.880047  00:0a:5e:65:8b:db  01:0c:cd:04:00:02  IECSMV    SMV 9-2 Publish
 5897 76.891621  10.0.1.101         01:0c:cd:04:00:01  IECSMV    SMV 9-2 Publish
 5898 76.895623  00:0a:5e:65:8b:db  01:0c:cd:04:00:02  IECSMV    SMV 9-2 Publish
 5899 76.895736  00:0a:5e:65:8b:db  01:0c:cd:04:00:02  IECSMV    SMV 9-2 Publish
 5900 76.907194  10.0.1.101         01:0c:cd:04:00:01  IECSMV    SMV 9-2 Publish
 5901 76.911203  00:0a:5e:65:8b:db  01:0c:cd:04:00:02  IECSMV    SMV 9-2 Publish
 5902 76.911312  00:0a:5e:65:8b:db  01:0c:cd:04:00:02  IECSMV    SMV 9-2 Publish
 5903 76.922835  10.0.1.101         01:0c:cd:04:00:01  IECSMV    SMV 9-2 Publish
 5904 76.925764  00:0a:5e:65:8a:fd  01:0c:cd:01:00:06  IECGOOSE  GOOSE Request
 5905 76.926794  00:0a:5e:65:8b:db  01:0c:cd:04:00:02  IECSMV    SMV 9-2 Publish
 5906 76.927306  00:0a:5e:65:8b:db  01:0c:cd:04:00:02  IECSMV    SMV 9-2 Publish
 5907 76.938556  10.0.1.101         01:0c:cd:04:00:01  IECSMV    SMV 9-2 Publish
 5908 76.943106  00:0a:5e:65:8b:db  01:0c:cd:04:00:02  IECSMV    SMV 9-2 Publish
 5909 76.943249  00:0a:5e:65:8b:db  01:0c:cd:04:00:02  IECSMV    SMV 9-2 Publish
 5910 76.954057  10.0.1.101         01:0c:cd:04:00:01  IECSMV    SMV 9-2 Publish

        TimeAllowedToLive (length=2): 2048 msec
        DataSet Reference (length=19): E1Q1B2C1/LLN0$Trips
        Application ID(length=11): gcbTestTrip
        Event Timestamp: 1970-01-01 00:00.0.000000  Timequality: 00
        State Change Number (length=1): 2

0020  4f 24 67 63 62 54 65 73  74 54 72 69 70 81 02 08   O$gcbTes tTrip...
0030  00 82 13 45 31 51 31 42  32 43 31 2f 4c 4c 4e 30   ...E1Q1B 2C1/LLN0
0040  24 54 72 69 70 73 83 0b  67 63 62 54 65 73 74 54   $Trips.. gcbTestT
0050  72 69 70 84 08 00 00 00  00 00 00 00 00 85 01 02   rip..... ........
0060  86 01 0c 87 01 00 88 01  20 89 01 00 8a 01 01 ab   ........ ........

P: 5989 D: 5989 M: 0
```

Figure 6: Response message from DUT.

5 CONCLUSION

A novel multi agent-based ODS can be applied in the IEC61850-based SA system is proposed in this paper. The technique presented in this paper detects the fault of IED automatically and it replaces the failed IED by backup IED to tolerate the fault in system level. Through the implementation and the case studies of the multi agent-based ODS, the availability is proven.

ACKNOWLEDGEMENTS

This work was supported by the ERC program of MOST/KOSEF (Next-generation Power Technology Center).

REFERENCES

Bertsch, J. *et al.* 2003, "Experiences with and perspectives of the system for wide area monitoring of power systems," *CIGRE/IEEE PES International Symposium, Quality and Security of Electric Power Delivery Systems*, 8-10 Oct. pp.5-9.

Yu, X. Z. *et al.*, 2000, "Condition Monitoring of a power station," International Conference on Power System Technology, Vol.2, 4-7 Dec., pp.1029-1033.

Ding M. et al., 2004, "Reliability analysis of digital relay," *Eighth IEE International Conference on Developments in Power System Protection*, Volume 1, 5-8 April, Vol.1, pp. 268 – 271.

IEC 61850, 2002-2005, *Communication networks and systems in substations*.

HyperSim, http://www.transenergie-tech.com/en/

SCADA WEB
Remote Supervision and Maintenance of Industrial Processes

José Ramón Janeiro, Eduardo J. Moya, David García, Oscar Calvo and Clemente Cárdenas
Fundación CARTIF, Parque Tecnológico de Boecillo. Parcela 205. 47151 Boecillo, Valladolid, Spain
josjan@cartif.es, edumoy@cartif.es, davgar@cartif.es, osccal@cartif.es, clecar@cartif.es

Keywords: Remote Supervision, Control, Maintenance, Industrial Process, PLC, OPC, OPC-DA, SCADA, Web, Internet, Java, GSM, GPRS, SMTP, PLC.

Abstract: This article explains a SCADA System called SCADA Web developed by CARTIF Foundation. This SCADA System makes possible remote monitoring and control from the process controlled by PLC using Web Technology. SCADA Web has been developed on platform Java EE and provides visualization and control signals, trend charts, alarms, historical data and alarm reports.

1 INTRODUCTION

Internet, together with the use of technologies such as GMS/GPRS/UMTS (International Engineering Consortium), provides communication mechanisms with a great number of possibilities (GeneraLynx). These technologies are present more and more in the industry.

Nowadays, a large extent of automation industrial solutions demand maintenance and supervision systems because of the high complexity and the need of managing a large volume of data.

Computers are the main support for supervision and maintenance systems due to the high performances and informations that they can process.

The main reasons to develop a SCADA Web (Janeiro, J. R., 2006) have been the growing use of the supervision and maintenance system on computers and the features that support the technologies GSM/GPRS/UMTS and Internet.

This paper is organized as follows. In Section 2 the characteristics that a remote supervision and maintenance system have to support are presented.

The SCADA Web Application arquitecture necessary to supervise a process in a remote way is presented in Section 3.

In Section 4 the main features and an operation manual of the SCADA Web Application is presented.

Conclusions and open issues for future research are discussed in Section 5.

2 REMOTE SUPERVISION AND MAINTENANCE OF INDUSTRIAL PROCESSES

Nowadays, more and more systems are being supervised in a remote manner in order to keep the process running in a correct state. The most important applications to be carried out are maintenance actions, although this depends on the nature of the process.

The traditional arquitecture of these systems in relation to remote supervision is composed by a Web Server that, after a correct authentication, allows access to remote supervision services of an industrial process. The supervision systems have to fulfill the next requirements:

- It is only necessary one computer and an Internet connection to supervise the system
- There is not need of special software installation in the remote computer, except for some specific plug in.
- Independence of the remote computer operative system.
- The access to applications must be controlled to avoid risks, due the that these remote supervision tasks can be accessible through a network like internet.
- Be careful with the exchange between remote computer and the process due to security problems. One possible solution is the use of safe communication protocols like HTTPS.

Maintenance systems send alarms or warning by means of SMS and e-mail. These methods supply an efficient notification reducing failure correction times and therefore reducing process costs.

3 SCADA WEB ARCHITECTURE

The acronym of SCADA Web Application comes from SCADA (Supervisory Control And Data Acquisition) and WEB (World Wide Web). This application makes remote supervision and maintenance of any industrial process possible. The above-mentioned process can be controlled by PLC´s of different brands by means of a web browser, an e-mail client and a mobile.

This application offers a number of features beyond the usual characteristics of a simple SCADA system. It contains elements focused towards industrial process maintenance like SMS messaging (Short Message Service) using technology GSM/GPRS (Global System for Mobile Communications /General Packet Radio Service) and e-mails.

Using OPC-DA(OPC-Foundation) protocol (OLE for Process Control-Data Access) based on OLE/COM/DCOM (Object Linking and Embedding/Component Object Model/Distributed Component Object Model) it is possible to access the variables.

In order to avoid the use of DCOM technology due to problems with the firewalls (OPC Programmers' Connection), we have developed an application called OPC gateway to supervise the process without limiting the structures of control. This application uses OPC-DA implementation in its interface "Automation" allowing the SCADA Web Application to access the variables by means of strings UNICODE based on Sockets. Although the OPC gateway application support DCOM, it is expected to be installed in the same computer where the OPC-DA server is located so that it is only necessary to use COM technology.

The use of OPC-XDA(OPC Foundation) (OLE For Process Control- XML Data Access) instead of OPC-DA was ruled out because of the generally scarce in utilization of this protocol by PLC´s manufacturers. Even so, it is desirable that the application supports both protocols in future versions.

In fact, the SCADA Web Application is a Web application developed using the platform Java EE (Java Enterprise Edition) (Sun Microsystems). Apache Tomcat (The Apache Software Foundation) has been the server used. SCADA Web Application is based on the next technologies: JSP, Servlet, Applet, Beans, JDBC, JavaMail, JAF and JAXB. The use of these technologies leads to a high development cost of the application, a null or low acquisition cost and high flexibilility and scalability compared, for example, to commercial SCADA.

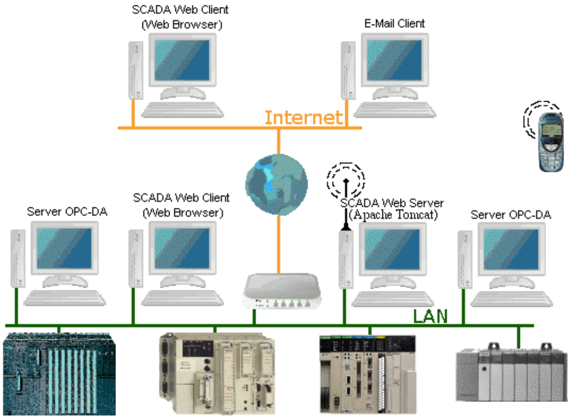


Figure 1: SCADA Web Application Structure.

The configuration parameters of SCADA Web Application are inserted in an XML file called *scadaweb.xml*. This file has to keep the xml schema which is defined in the file *scadaweb.xsd* The use of an XML file to configure the application allows a high degree of flexibility and scalability in the parameters configuration, as well as a simplification of the configuration process to the user.

Thanks to JAXB (Java Architecture for XML Binding) the schema XML is mapped to JAVA classes providing a simple way to access the configuration file *scadaweb.xml*. The SCADA Web Application is divided in two modules called Supervision and Maintenace modules. Next, these modules will be explained.

3.1 Supervision Module

This module has the typical characteristics of a comercial SCADA. An industrial process can be remotely supervised via WEB thanks to this module.

Although this module was designed for remote supervision via WEB, it can be used for local supervision of a process. Furthermore, thanks to its generic design, it's possible to supervise the process using informatics nets so complex as desired (LAN, Internet, Routers, ...).

The Supervision Module of SCADA Web Application is a multiuser system that lets to several remote users supervise the same process at the same time.

This module is based on a client-server architecture (Applet-Servet/JSP) which allows to distribute the computacional load of the application between Tomcat Apache server and remote machines.

An Applet is an application, written on Java, which is executed on a Web browser. Using an applet allows us to have a more complete interface from the point of view of the client.

A Servlet is a Java class that is carried out on an application server in order to answer clients requests. The clients are Applet´s in the case of SCADA Web.

A page JSP is a file that combines HTML targets with script targets which belong to Java Server Pages specification. The page JSP has a similar aspect to HTML targets, but they are changed in Servlet in order to compile and generate the appropriated class files. The resulting file from the transformation of page JSP is a combination of HTML code cointained in JSP page and dynamic content which is especified by JSP specification targets. The exchange of information between Applet and Servlet is carried out by HTTP tunneling, in other words, transmitting binary information using HTTP protocol.

Due to security obsession this protocol is usually the only possible method of information transmission.

3.2 Maintenance Module

This module is executed as a daemon checking permanently that alarms have been configured to take place. Alarm notification is carried out by maintenance workers using two diferent methods depending on configuration.

The first method consist of an alarm notification using GSM/GPRS technology by SMS. It is necessary that the computer where SCADA Web application is installed, is connected to a GSM/GPRS modem, normally through a serial port. Communication between modem and SCADA Web is made by standard AT commands. The library command used to operate the computer ports is one from the RXTX organization (RXTX Serial and parallel I/O libraries supporting Sun's CommAPI).

The second method consist of alarm notification using e-mails. A SMTP server which is configurable in SCADA Web Application is used. In this case the library used is JavaMail from Sun Microsystems.

4 SCADA WEB

SCADA Web Application is executed within applications servers. This application allows an interaction between user and process by means of three interfaces, HTTP, GSM/GPRS and e-mails.

The first interface, HTTP, makes it possible to supervise the process in a remote manner from a web browser.

The interface GSM/GPRS is oriented to offer an instantaneous warning procedure reducing failure correction times and the costs of industrial processes.

Finally, the e-mail interface carries out the same functions that GSM/GPRS interface, although it allows to send more information. The main reason is to provide redundant alarm warning procedures, so that the user is notified as soon as possible. E-mail, unlike GSM/GPRS, is used to notify less critical alarms. The HTTP interface has been grouped inside what has been called Supervision Module, while GSM/GPRS interface and e-mails are in Maintenance. Next, these modules are explained.

4.1 SCADA Web Configuration

As in other software applications, the configuration of SCADA Web Application is done by an XML file according to XML Schema called *scadaweb.xsd*.

This configuration file has four blocks well differentiated: Parameters, Maintenance, Protocols, and Users.

The reading frecuency of PLC´s variables in "Paramaters Block" is normally about half a second. The update frequency of the different databases of the supervisory system is also configurated These are called reports or historical data.

Inside the "Maintenance Block" the neccesary basic parameters required by the above-mentionated block are configurated. There are two sections in this block: SMS and E-MAIL. In SMS Section the port is configurated. It contains the modem GSM/GPRS the PIN, the SMS Server phone and the alarms to detect besides text message and the phone where it will be sent.

The SMTP server (address and port), the sender address, alarms to detect together with the content and message addressee are configurated in E-MAIL Section.

In Protocol Block the available protocols, in this case OPC-DA, are added together with the adress and port of OPC-DA server.

In Users Block passwords, usernames and access permission to the different parts of the program are configurated.

4.2 Supervision Module

The necessary requirements to access SCADA Web application are Internet connection and a computer.

After previously stated requirements are fulfilled it is necessary to open a web browser window and to connect to the server where the SCADA Web application is installed. A similar window is showed in Figure 2.

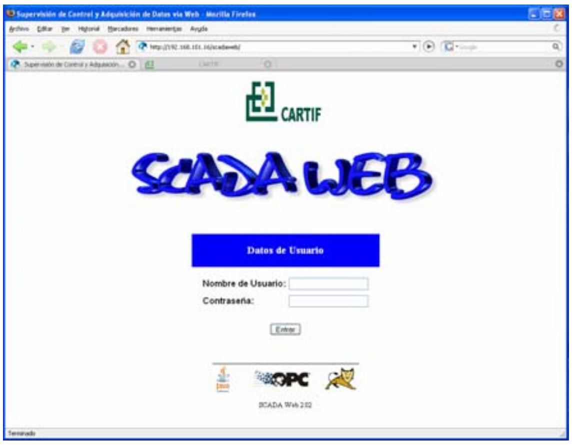


Figure 2: Initial Window of SCADA Web.

In this window it is neccesary to authenticate in order to access to the application. After a correct authentication, it is possible to access the applications: “Builder” and “Viewfinder”.

The “Builder Application” is going to allow the creation of a series of windows with components thanks to a high level graphic interface with the user.

Linking ones with the others it is possible to build a SCADA system of a specific installation.

In the figure 3 an image of “Builder Application is showed”.

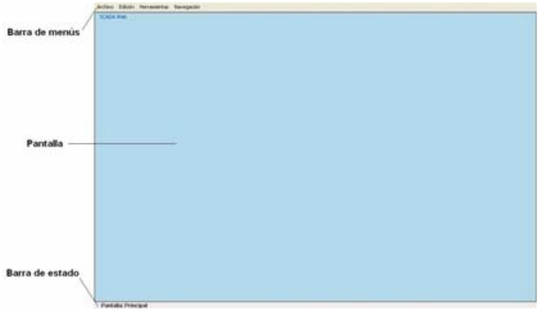


Figure 3: SCADA Web Application: “Constructor”.

The application allows to add windows so that a SCADA system can be built with so many windows as desired. Each window can be associated to a wallpaper.

Hereafter the available components of the application will be explained. Every component has a properties dialogue box associated that allows to realize its configuration.

Some of the components that can be added (Besides of labels, rectangles and image components) are:

- **Text Field** displays analog and digital variables.

Figure 4: Text Field Component.

- **LED** can be associated with one or more digital variables. This element makes it possible, by means of colors, to show the state of the variables. A typical example is failure-non failure state monitoring and on/off motor state

Figure 5: LED Component.

- **Button** is used to control digital or analog variables. When the button is pressed, depressed or is pressed during a time period, the variable changes of value.

Figure 6: Button Component.

- **Vertical Bar Graph** simulates a level and it is associated with analog and digital variables.

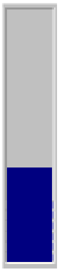
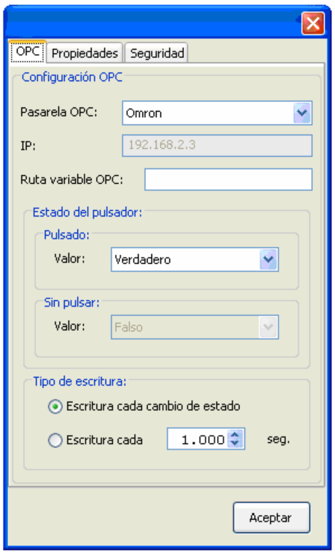


Figure 7: Vertical Bar Graph.

- **Toggle** allows to supervise digital variables. The monitoring of variables is carried out by the state (pressed/depressed) in which the toggle can be found. When the user presses, the toggle the variable value is modified. In Figure 9 the dialogue window to configure the component is depicted.

Figure 8: Toggle Component.



Configuración OPC

Pasarela OPC: Omron

IP: 192.168.2.3

Ruta variable OPC:

Estado del pulsador:

Pulsado: Valor: Verdadero

Sin pulsar: Valor: Falso

Tipo de escritura:

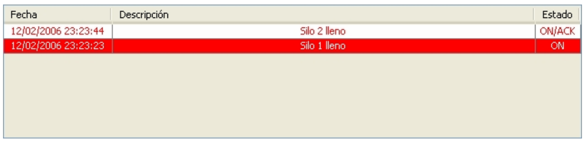
● Escritura cada cambio de estado

○ Escritura cada 1.000 seg.

Figure 9: Dialogue window of a toggle.

Alarms. In the SCADA Web Application it is also allowed to configure alarms. It is possible the monitoring and recognition of alarms. Alarms are administered in the server. Because of this, the consistency of all alarms by users can be saved. In this way we can supervise remotly and simultaneously the process.

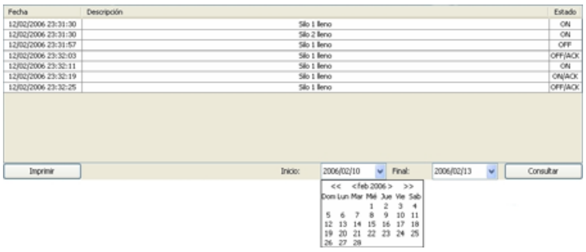
Moreover, in order to avoid lacking firmness in relation to dates, every happened event is registered according to server time. All these events are registered in a database to a subsequent consultation. In Figure 10 the alarms component is showed.



Fecha	Descripción	Estado
12/02/2006 23:23:44	Silo 2 lleno	ON/ACK
12/02/2006 23:23:23	Silo 1 lleno	ON

Figure 10: Alarms Component.

Alarm Reports. Alarms are registered in a database which can be looked up using the “Report Component”. The generated reports can be printed.

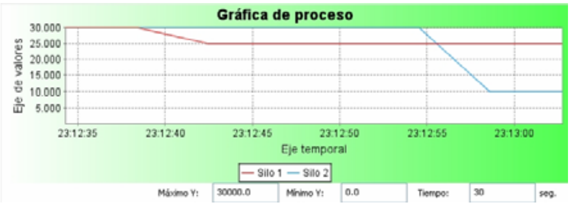


Fecha	Descripción	Estado
12/02/2006 23:31:30	Silo 1 lleno	ON
12/02/2006 23:31:30	Silo 2 lleno	ON
12/02/2006 23:31:57	Silo 1 lleno	OFF
12/02/2006 23:32:03	Silo 1 lleno	OFF/ACK
12/02/2006 23:32:11	Silo 1 lleno	ON
12/02/2006 23:32:19	Silo 1 lleno	ON/ACK
12/02/2006 23:32:25	Silo 1 lleno	OFF/ACK

Inicio: 2006/02/10 Final: 2006/02/13

Figure 11: Reports Component.

Trend Charts. Besides monitoring process variables using controls explained before is possible to obtain the trend of these variables by graphics using their corresponding component. Their value can be also scaled as it is showed in the Figure 12(JFree.org).

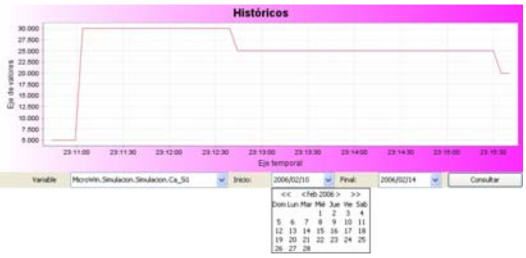


Máximo Y: 30000.0 Mínimo Y: 0.0 Tiempo: 30 seg.

Figure 12: Graphics Component.

Historical Data In this last component the data to register in a database are configurated. Later, during the process supervision it will be possible to do consultations of these data by graphics or accessing directly to the database to obtain the real value. An example is depicted in Figure 13 (JFree.org).

Besides, is necessary to configure users that could use this SCADA system and the permission level of each one. Due to this and to the fact that each component has a permission level associated is possible to control the access as well as to adapt the SCADA system according to permission level of the authenticated user.



Variable: MicroWin.Simulacion.Simulacion.Ca_S1 Inicio: 2006/02/10 Final: 2006/02/14

Figure 13: Historical Component.

After having built the SCADA system of a particular installation, then it will be possible to keep in a local hard disk or in the server. In the last instance it is possible to do a remote supervision of the process if you have a computer and internet connection.

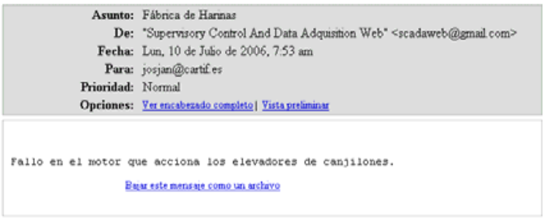
"Viewfinder Application" makes it possible to carry out the SCADA system created with the "Builder Application". In order to configurate these configurations it is necessary to open them and after a correct authentication the variable update frequency will be fixed.

The configuration of update variable frequency make it possible to adapt the refresh time to the necessities of the person that wants to supervise the system. A clear example happens in industry when the system goes to be supervised simultaneously by workers in the monitoring process and quality control.

4.3 Maintenance Module

This module sends alarms of the industrial process by SMS messages and e-mails. The complete configuration is done by XML configuration file.

Thanks to this module the alarm notification time generated is decreased and the costs in the industrial process associated with alarms are reduced. Furthermore, all alarm events and e-mail and SMS messages associated are registrated in SMS and e-mail databases.



Asunto: Fábrica de Harinas
De: "Supervisory Control And Data Adquisition Web" <scadaweb@gmail.com>
Fecha: Lun, 10 de Julio de 2006, 7:53 am
Para: josjan@cartif.es
Prioridad: Normal
Opciones: Ver encabezado completo | Vista preliminar

```
Fallo en el motor que acciona los elevadores de canjilones.
```

Bajar este mensaje como un archivo

Figure 14: An alarm e-mail.

5 CONCLUSIONS

A system SCADA for remote monitoring and control of procceses is suggested. This SCADA Web Application consists of a sub-application called "Builder" that allows to make SCADA systems for an industrial plant and a sub-application "Viewfinder" whose function is supervise an industrial process based on SCADA systems built by the "Builder".

The process variables of an industrial plant controlled by different PLC´s brands can be remotely supervised by this application. It is only necessary to have a computer and Internet connection. The SCADA system can also represent by graphics the current evolution of process variables and the configuration of alarms to warn when a failure takes place. Besides, these alarms are registered and can be recognized by user.

There is a possibility of printing and the monitoring reports by SCADA system of industrial plant.

The SCADA system allows to create historical data in order to a later consulting in a visual way.

In the SCADA permission levels to adapt the posibilites of supervision tasks to worker permission level who supervises the installation are configurated.

Finally, it shoud be remarked that thanks to use an object orientation programming language and an organization of application in components it is possible easily add new elements in this application.

ACKNOWLEDGEMENTS

This work was supported in part by "Programa de Fomento de la Investigación Técnica para los Centros Tecnológicos", (PROFIT grant FIT 330220-2005-138) from the Spanish Education and Culture Ministry.

REFERENCES

International Engineering Consortium. Global System for Mobile Communication, <http://www.iec.org/online/tutorials/gsm/> (last accessed on January 10, 2007)

Janeiro, J. R., 2006. Supervisión remota de procesos industriales controlados por Autómatas Programables, University of Valladolid. Spain, 1nd edition.

JFree.org. JFreeChart, <http://www.jfree.org/jfrechart/> (last accessed on January 10, 2007)

OPC Foundation. Specifications,http://ww.opcfoundation.org (last accessed on January 10, 2007)

OPC Programmers' Connection. OLE for Process Control, http://ethernet.industrial-networking.com/ieb/article display.asp?id=21 (last accessed on January 10, 2007)

RXTX. Serial and parallel I/O libraries supporting Sun's CommAPI, <http://ww.rxtx.org> (last accessed on January 10, 2007)

Sun Microsystems. Java Enterprise Edition, <http://java.sun.com/javaee/> (last accessed on January 10, 2007)

The Apache Software Foundation. Apache Tomcat, <http://tomcat.apache.org> (last accessed on January 10, 2007)

GeneraLynx. Remote supervision and control by WAP, <www.euroines.com/down/DemoDoc/WapScada%20DD.pdf> (last accessed on January 10, 2007)

PROPERTY SERVICE ARCHITECTURE FOR DISTRIBUTED ROBOTIC AND SENSOR SYSTEMS

Antti Tikanmäki and Juha Röning
Computer Engineering laboratory, University of Oulu, antti.tikanmaki@ee.oulu.fi, jjr@ee.oulu.fi

Keywords: Distributed robots and systems, multi-robot systems, sensor networks, Property Service architecture.

Abstract: This paper presents a general architecture for creating complex distributed software systems, called Property Service architecture. The system may contain resources like robots, sensors, and different kinds of system services, such as controller units, data storages, or a collective model of the environment. This architecture contains several solutions and distributed system design methods for developing advanced and complex systems. It also provides the possibility to add new resources to the system easily and during operation. Each service has a very simple generalized interface. This meets the requirements of distributed robotic applications, such as remote operation, multi-robot cooperation, and the robot's internal operation. The simplicity of the interface also provides a possibility to scale down the service even on the low-cost, low-performance microcontrollers used in small-sized robots. The main features of the architecture are the dynamic properties of the resources, automatic reconfiguration, and the high level of reusability of the implemented methods and algorithms.

1 INTRODUCTION

Creating a distributed system from several kinds of devices and functionalities is a very demanding task. Since, for example, a robot may contain many different kinds of functionalities, actuators, and sensors, the process of controlling is complex and the interface becomes very complex. Different systems cannot communicate with each other if there is no single commonly used communication standard. Figure 1 shows an example of a set of resources that a distributed system might have. The operating environment might have several sensors, lights and cameras. Different robots might need to be used, some of them might be legged, and they might contain several kinds of sensors. The human operator has a graphical user interface where tasks can be selected, and the system might have several automatic functionalities. Each device is used in different ways, and they can communicate in different ways. The lighting of the room is connected to a special control protocol, room cameras can be controlled using a serial port, and mobile robots use a wireless LAN. As the number of resources in the system increases, for example when a new kind of robot is developed to the system, the process of maintaining and updating the interfaces becomes very challenging.

For heterogeneous devices that take part in a larger system, a common problem is how to define an interface for each service in the system. This becomes even harder in multi-robot systems, where different kinds of robots with different capabilities. As the system improves, the interface must be changed and the whole system must be recompiled.

In real world applications when the communication channel between robots also unreliable, a common solution is to reduce communication and increase autonomous functionalities. Instead of sending drive commands to a robot, a target of movement is delivered to the robot. Increasing the capabilities of the robot leads to an increase in features that must be accessible through the interface. This sets either dynamic requirements for the interface or a need for continuous updating of interfaces.

To be able to achieve inter operation of a versatile set of robots and devices, very simple but flexible middleware is needed. The main requirement for the middleware is that it is possible to scale down to small-sized devices with a very limited set of capabilities. On the other hand, it must be able to provide an interface to varying features of resources in the system and expand as new resources are added to the system. This paper proves that it is possible to implement flexible application layer that can be implemented on all kinds of transportation layers while providing all the capabilities of state-of-the-art distributed architectures.

Our solution, called Property Service architecture, has been developed to be a generalized interface for all kinds of devices or resources in a distributed system. Property Service provides the possibility to easily create and add new resources to a distributed system. The main criteria for the architecture design were simplicity, scalability, dynamics and expandability, and high reusability of system components. Dynamics and expandability make it possible to add new features and functionalities to services even during operation, as well as to integrate the results of different research topics into a larger system. This is essential to be able to build robotic systems with multiple capabilities.

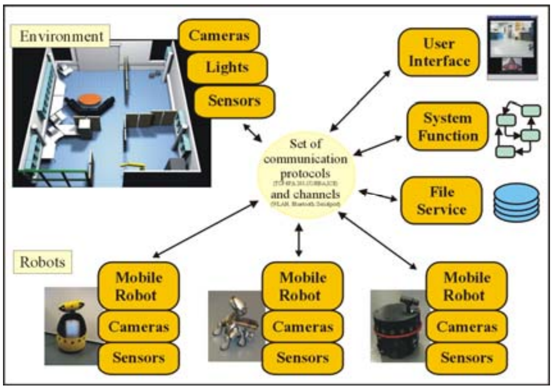


Figure 1: Set of resources that might be a part of a distributed system.

This paper describes the whole architecture that has been partly introduced on several papers previously. It will describe system-level services that improve the creation and management of distributed systems and heterogeneous multi-robot systems.

2 RELATED WORK

Many efforts have been made to create a more generally useful interface for networked robots. In many cases, communication is based on commonly used technologies like TCP/IP protocols and middleware like CORBA, Soap, etc. In multi robot architectures, several of these are based on CORBA or a real-time CORBA (OMG, 2002) extension. Examples of robotic architectures using RT-CORBA are Mobility software (Rea, 1999), the Player/Stage project (Gerkev et. al. 2003), Miro (Utz et. al. 2002) and ORCA (Brooks 2005), which is based on a CORBA component model. Later in ORCA2, CORBA has been replaced with Ice (Hemming, 2004) middleware, which provides improved features compared with CORBA

Wang et al. presented a COM-based architecture for the fusion of logical sensors (Wang et al. 2001). However their approach has some drawbacks, including platform dependency and a lack of network transparency. DDX architecture (Corke 2004) for a distributed system has been developed using UDP/IP to transfer data from remote sensors to a central data store. However their support for only lower-level data structures increase the amount of data transferred during operation and lacks expandability.

Algorithms and remote control code on heterogeneous robotics systems have also been developed with higher reusability in mind. CARMEN (Montemerlo, 2003) uses IPC to communicate between system components and provides a reusable navigation toolkit for multiple robotics. URBI (URBI www) scripting language support several kinds of robots and robot simulators. URBI also provides client/server-type networking for remote control of robots, in-language parallel processing and support for several commercial mobile robots.

Even though several possible solutions exist already, they have several disadvantages. Relying on a single communication protocol or middleware limits the possible platforms on which the service can run. As new features are added to the resource (e.g. a robot or sensor), the interface must be modified, which causes system-wide changes in the remote software that uses the interface. General middleware also requires a lot from the CPU, memory, and disc space, and it cannot be run on robots with an embedded CPU.

3 PROPERTY SERVICE

Property Service architecture provides tools for building new networked systems in just a few hours. It provides standardized data types and service types that can operate on different kinds of platforms. Property Service has been successfully used for remote operation of mobile robots (Tikanmäki, 2003), multi-robot cooperation (Mäenpää et al. 2004), remote operation of a swarm of robots (Tikanmäki, 2004), and to create a dynamic interface for a modular robot (Tikanmäki and Vallius 2004). This paper presents the complete architecture and several system services, such as a storage service, a resource service, and grouping services, and explains how they can be used to build more complex distributed systems containing robots and other devices and system resources.

Properties are the features of each service, such as the sensors of a robot, or they are functional properties related to the operation of the service. Functional properties include, for example, the autonomous features of robots; or the tracking capabilities of a sensing device. Each property is a paired name and value and it can be set or requested. The value of each property is transferred as character arrays through a communication channel, and structured data is represented in XML format.

Each service has a special property called "*properties*", which contains a list of the properties currently available on the service. The value of "properties" changes during operation as the set of properties of the service changes, providing dynamic interface functionality.

Some properties may contain sub-properties. This hierarchic representation provides the possibility to request complex representations or, on the other hand, only special features of a property. The format is the same as that used in several programming languages for separating fields of structures. For example, by requesting the "location" property from a robot service, the client receives a 6D vector of the current location of the robot. The client can also request the "location.x" property, which returns only the x coordinate of the robot's current location.

A request for certain properties may also contain some parameters that specify, for example, the context of the return value. A good example of this is a request for robot's location, which might be requested in several coordinate systems. The client can request possible parameters using the ".parameters" extension on property name.

3.1 Service Interface

Property Service has a simple interface, which contains only two methods, "SET" and "GET", for getting and setting the properties of a service. For example, a connected client can GET the velocity of a robot through Property Service or SET the velocity of the robot. The terms client and server are misleading, as both the client side and the server side implement the Property Service interface. The term 'client service' is used when talking about user interfaces or other services that request properties from other services. Figure 2 shows the general principle of the architecture. In the Property Service architecture, user interface components can be thought of as a sensor that senses user inputs and provides these as properties. Therefore, the user interface can have properties that other services can set or get. This feature is used in the listening mechanism. A client service can request listening of a certain property of the other service. This service registers information concerning the connection of the client service to the requested property. Each time the value of the property changes in the service, an event is sent to the client service. The main advantage of this is that the client service does not need to continuously request value changes, thus reducing the amount of required communication. This is especially useful when changes in the property value occur over a long period.

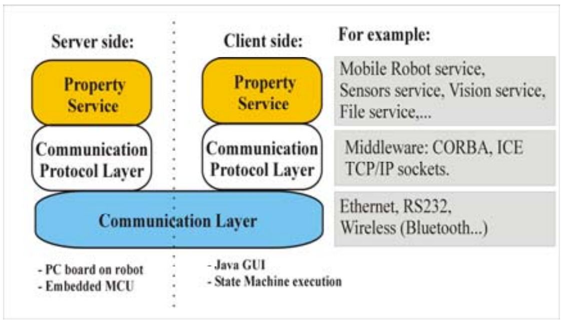

Server side:
Client side:
For example:
Property Service
Property Service
Mobile Robot service, Sensors service, Vision service, File service,...
Communication Protocol Layer
Communication Protocol Layer
Middleware: CORBA, ICE TCP/IP sockets.
Communication Layer
Ethernet, RS232, Wireless (Bluetooth...)
- PC board on robot
- Embedded MCU
- Java GUI
- State Machine execution

Figure 2: Principle and examples of property service layers.

3.2 Data Types

Property Service has a simple interface, which Several commonly used data types have been standardized to provide compatibility between services. The most commonly used ones are vector, list and markers. These data types are briefly introduced in the following.

A *vector* is an array of decimal numbers and its length can vary. For example, a request for a robot's location returns a 6-dimensional vector containing x, y, and z translations and a, b, and c rotations along each axis. A vector is also used to deliver measurements from various sensors, like distance sensors. A vector is also used to represent sequences, histograms, etc.

A *list* is a set of data types. It is used, for example, to list the properties that a service has. A list can wrap several kinds of data types, including a robot's path, which is presented as a list of locations (presented as vector types). The shape of an edge of an object can also be presented as a list of vectors containing image coordinates.

A *marker* is a structure that stores information about detected and tracked objects in the environment. For example, a robot's vision system provides markers for objects currently in the field of view. The marker structure contains location information and a set of detected features that the object has. For example, a ball recognized from the camera view can have recognized features like shape and color. Markers provided by different services can be collected into a model that represents current knowledge about the current environment. This provides the possibility to easily build sensor networks and swarm robotic systems.

Markers are used as input parameters for the behaviors of various services. In addition to physical objects they can also represent abstract entities that can be used to create new behaviors. For example, a target marker for a "move to" behavior that stays in front of a robot causes the robot to move forward. The measurements of each sensor can be presented as markers, which is highly useful in creating distributed sensing systems. For example, a swarm of robots produces a large amount of markers of the objects they detect. These markers are collected into one place to create an environmental model based on where the swarm operates.

Table I shows an example of each of the basic data types. As it can be seen, for example, the color of the marker is represented with a vector type. In most cases these data types are used, but each developer can also define their own data structures.

Even the interface remains the same; it is possible to make dynamic "interfaces" by changing the set of properties provided by a service. New properties can be attached to a service during operation, and they are shown when the client requests "properties" again. This feature is used in the Qutie robot (Tikanmäki and Vallius 2004), where the properties of a new attached hardware component appear on the list of properties as soon as they are available.

Table 1: Examples of commonly used data types.

Data type	Example
vector	(1.0 0.1 0.1)
list	((1.0 0.0) (2.0 0.1))
marker	<marker> <location>(1.0 0.1)</location> </marker>

As the value of the property in communication is an array of characters, a service can be implemented over any communication protocol containing the possibility to transfer data characters. The current implementation contains RS232, TCP/IP sockets, HTTP as well as several middleware like CORBA and ICE. To be able to communicate between various protocols, special proxies have been made to transfer a call from one media to another. As an example, the Property Service in Sony's AIBO robot communicates through a TCP/IP socket and a Wireless LAN with a PC that contains a wrapper to convert property calls to other protocols, such as ICE or CORBA calls. This provides the possibility to control AIBO from any property client without knowledge of how the AIBO is actually connected to the system. The implementation of a proxy is simple, as it simply delivers property calls from one protocol to another.

3.3 GUI as a Property

A service may contain special properties that can be used to create user interfaces automatically. Several of properties may have a ".gui" extension property that returns the preferred GUI component for the property. The service can also provide a class that contains the compiled code for the GUI component. For example, the "movement.gui.java" property returns the java class that can operate property movement on the service. If a new version of the GUI component is then developed, it can be received from the service without a need to change the client side user interface. Instead of requesting the GUI components from each service they can also be requested from a special “GUIPropertyService”, whose purpose is to provide standard GUI components for various property types

3.4 Data Transfer Principles

The ideology of Property Service is to always reduce the need for data delivery if possible. One way to do this is by implementing several automatic features for services. Instead of continuously sending moving commands to a robot, we prefer to send only target location or several route points. Instead of sending raw image data, we prefer to request only the targets that vision system can recognize from the view. If raw images are needed for remote driving of a robot, a commonly used compression format, such as JPEG, is used.

Several standard sets of properties have been used for different services. As each service delivers data in the same format, and understands the same commands, different services can be used by the client service. A good example of this is to use different robots with the same user interface.

3.5 Platforms

Currently services are implemented using C, C++, Java, PHP or Python. Services can be compiled and run with Windows and different Unix systems. A Property Service for AIBO's operating system Open-R has also been developed.

As one criterion for the architecture was simplicity, services can also be implemented on devices containing low calculation power. In many types of robotics or sensor nodes on sensor networks, it is reasonable to use embedded microcontrollers instead of a PC. Property Service has also been implemented using standard C and can be used on many embedded platforms, for example, in “Atomi” boards containing an Atmel 8-bit AVR microcontroller (Tikanmäki and Vallius 2004) and ARM-based embedded computers running embedded Linux. These two platforms are good examples of systems that cannot run complex middleware.

3.6 Implementing New Resources

The implementations of Property Service include several functions and classes that facilitate the creation of new services. Automatic features, such as a dynamic set of properties, are ready for use in the PS core library. One of the most useful tools is called Class Property Service. Class Property Service is a class that creates a service automatically from given class. The automatic service creation is available for Java and C++ classes. The main advantage of this service is that it facilitates the reuse of previously built classes. For example, researchers may want to be able to use their old manipulator remotely. The manipulator has a control box that can be used from a PC through a serial port. By using Class Property Service, a previously made control class can be used as a part of the distributed system, as the Class Property Service creates properties from methods and attributes of the class according to defined rules.

In addition to robots and sensors, Property Service can be used to control various other devices in the environment. Since the interface is simple and can also be used in simple microcontrollers, different kinds of devices, like electric locks, motion sensors or room light control units, can be included in the distributed system. This provides the possibility for a robot to use these resources, which increases the possibilities of new applications.

3.7 Standard Services

Some commonly used types of services have been standardized, so that each service in the type category provides at least a certain set of properties. The main advantage of this is the possibility to use several kinds of services (or system resources) without a need to modify the control program. A good example of this is that the user can change the robot into a different kind of robot to do the same task without modifying the control program.

3.7.1 Mobile Robot Service

The main standardized service is a mobile robot service. A mobile robot service contains properties related to movement, sensing and physical dimensions. Each robot's property service provides the same properties. Table 2 shows a basic set of properties of a mobile robot service.

Table 2: Examples of commonly used data types.

location	6D global location of the robot, containing x, y, z and rotations around the axis
velocity	6D movement of the robot, including translations and rotations
actuators	List of actuators on the robot, contains sub-properties for each actuator
sensors	List of sensors, with several subproperties
behaviors	List of the robot’s primitive functionalities. All robots must have at least a "move to" behavior.

Standard properties are the same regardless of the moving mechanism of the robot. When the robot is requested to move forward one meter (by using a “*behaviors.moveto*” functional property with a marker as a parameter), a walking Aibo or a wheeled Scout robot moves one meter forward according to its moving capabilities. If a robot contains additional actuators, such as arms or legs, properties for them are added to a list of *actuators* properties, but the basic functionality remains the same. The main advantage of this is the possibility to control all kinds of mobile robots with the same control software or user interface while also providing special features of each robot.

3.7.2 Sensor Service

One main type of service is sensor. A sensor service includes different sensors, such as sonar and cameras, as well as different controlling devices, e.g. joysticks, GPS devices and touch pads. User interfaces are also like sensors, because they sense the operational requests of the human user. The basic properties of sensor services are "location", which contains the location of the sensor in its context, and "data", which returns the raw data captured by the sensor. The default coordinate system for sensors location is relative to its base, for example, the robots origin. By using request parameters, different coordinate systems can also be used. In addition to raw sensor data, various kinds of refined information can be attached to the sensor's properties. The most advanced system is the vision sensor's property set.

Vision service is the most advanced sensor service and a good way to integrate several kinds of machine vision methods to a robotic system. Several different methods can be activated separately, and the result of image processing can be requested from the service. Results can be requested in several forms. For example, a remote client can request an edge-detected image or segments of edges from the vision service. The vision service is the interface to one or several cameras, but it can also process images sent by other services. As the interface is always the same, different kinds of cameras can be used in the same way. All targets detected by the vision sensor can be received in marker format.

3.7.3 Resource Service

To be able to find the available resources in s distributed system, a special Resource Property Service has been introduced. The properties of this service are references to services available in the system. New resources can contact it to notify of their presence. When a service contacts the resource service, it gives information how it can be contacted, like its address and protocol(s). Services are then shown as properties of Resource Property Service and each property contains the name of the service and contact information. For example, CORBA-based services' contact information is the IOR of the service and TCP/IP or ICE services indicate their IP address and the port where they can be reached.

The resource service actively checks the existence of services and removes the resources that are not available any more. Resource Property Service also contains several functional properties that can be used to search services using parameters. For example, a client might want to find all available mobile robots located in a certain room. For the search request, Resource Property Service searches for services that are mobile robots and whose "location" property matches the client’s request. The resource service also starts up new system services when necessary.

3.7.4 Grouping Services

Another architectural solution is to group several services into one Property Service. A good example of this is to group a set of simple miniature robots into one Swarm Property Service (Tikanmäki 2004). Miniature robots with low computational power are commanded through a common radio channel. Each robot's properties are shown in a hierarchical set of properties of the swarm service, shown in Table 3. The properties of a single robot can be reached through these properties, and each property of the mobile robot service interface becomes a sub-property of Group Property Service with a “robots.<name>” prefix. The “<name>” parameter changes according the robot currently used. In addition to a single robot's properties, Grouping Property Service contains various functional properties that can simultaneously control a larger set of robots. For example, the client wants a swarm of robots to move to a certain destination. Grouping Property Service can create a sub set from selected robots and order each individual to move to the destination. In the case of group of mobile robots, a groups of robots are controlled by properties similar to those used to control an individual robot. For example, Swarm Property Service has “behaviors” properties that control the whole group.

Table 3: Examples of properties of swarm service.

robots	robots currently belonging to this swarm each robot has sub-properties included in robot service properties
location	location of the swarm
behaviors	List of the primitive functionalities. Similarly to single robots, group can be controlled with same behaviors like "move to" behavior.

Resource Property Service and Group Property Service have several similarities, and Resource Property Service can be expanded to act as a grouping service. In this case, the resource service reroutes the property request. Instead of giving the reference of the service to the client, it acts as a proxy and asks for the property value from the service and delivers the reply to the client. This feature is useful in, for example, a long latency system and a long request time in some cases, because the grouping service is already connected to the service, and therefore the client does not need to make a direct connection. These services also provide full transparency to the system, as the client does not need to know the address or even the number of robots taking a part in the action.

3.7.5 State Machine Service

In addition to user interface client services, some control services have also been created. One of the main services is State Machine Service. It provides a resource that can execute RDF-formed state machines, which can control complex conditional sequences, being able to provide multiple property services at the same time. This has been described previously in (Mäenpää 2004). State Machine Service can control a large set of services simultaneously. Currently, State Machine Service also has a visual user interface, where the user can create and modify state machines graphically and during operation. By using Storage Service, introduced below, the state machine or part of it can be stored into a database for further usage or execution.

3.7.6 Storage Service

Another architectural service is Storage Service. It is a combination of file transfer systems and databases. The service can be started for a specified directory in operating systems and all (or only selected) files are shown as properties of the service. A remote client can request a property, named according to the filename, and receive the value of the property, which is the content of the file. The same procedure can be used to store new files and data into a service, by setting a new property in the service. For example, a robot can send captured images to a storage service, which stores the images in the memory (or on a hard disc) of the service. As in Resource Property Service, Storage Service may also contain several intelligent search methods, and the client can request files according to specified features. Storage Service has been used to store captured data such as images, to upload and download control sequences and state machines, and to request GUI components remotely.

3.7.7 Environment Model as a Service

To be able to operate, a distributed system must have knowledge of the environment where the services, such as robots and sensors, are operating. If the system is not designed to be fully reactive, it is reasonable to collect information received by services. The system can have one or several environment model services. If only one model is used, all the services send their measurements to this service, and the model service performs data fusion and updates the model. Each service might also have its own model, which is a local representation of the service’s way of receiving information from the environment. For example, a ceiling camera might have an environment model service that provides information received using the camera. Markers are used in most communication and data storing, but other methods are also possible. For example, a model can contain a grid of temperature values gathered by a swarm of small robots or a sensor network. In this case, the temperature map is updated according to the locations of the measurements.

4 COMPARISON

The overall performance of the system depends on the computational power of the platform and the speed of the communication channel used. However, in a comparison of the development times of a new service and cooperation between different middleware, Property Service provides a great advantage. The amount of required lines of code is very small compared with, for example, CORBA-based robot middleware that provides the same features and functionalities of the resource. As the capabilities of a robot or other service increase, it becomes even more useful that the actual interface does not need to be changed.

As the code is highly reused, new services are fast to implement and all system services like state machines, and storage services are available for use immediately. For example, the “move to” behavior is common to all robots; no matter if they are legged, wheeled, or even a manipulator. System services also provide great advantages in building applications for distributed systems. This provides even more advantages compared with other systems.

Passing messages as text strings is expensive compared with other formats (like language-specific byte arrays). The performance of communication has been improved by sending only essential information and by using refined data instead of continuous control commands or raw sensor data.

Several applications and example systems have been created using Property Services. Property Service has been successfully used for remote control of several mobile robots that provide audio and video data, and receive several moving commands (Tikanmäki 2003) using CORBA middleware and a wireless LAN. It has also been used to create remote operation for a swarm of simulated robots (Tikanmäki 2004). Both are good examples of applications where quick response between services is an essential requirement. Multi-robot cooperation and designing of operation using state machines has been demonstrated in reference (Mäenpää 2004).

5 CONCLUSION

The main advantage of Property Service in compared with competitive architectures is the ease of adding new resources to a distributed robotic system. Using Class Property Service, classes that are already available can be used remotely and connected to a larger system quickly and without a need to implement a code related to communication between services. As Property Service can be implemented on various communication channels and platforms, different kinds of new resources can be attached to the system. The usability of the architecture is not limited to robotics, it can also be used in other distributed systems, for example, in home automation systems, sensor networks, and industrial automation. As these devices become part of the architecture, they can be operated remotely or by the robot's control software, and robots easily became a part of the system. Using Resource Property Service, the robot can, for example, search for the light service of a room and switch on the light on the room upon entering the room. Complex applications built using state machines are easy to change, and the user can monitor their operation online using state machine visualization.

ACKNOWLEDGEMENTS

This work has been partly funded by Finnish Academy.

REFERENCES

OMG. *Real-Time CORBA Specification. Object Management Group, Inc.,* 1.1 edition, August 2002.

Rea 1999. *Mobility 1.1, Robot Integration Software, User’s Guide,* iRobot Corporation, MobilityTM software

Gerkey, B., Vaughan, R. T. & Howard, A. 2003, *The player/stage project: Tools for multi-robot and distributed sensor systems,* in `Proceedings of the 11th International Conference on Advanced Robotics (ICAR'03)', Coimbra, Portugal, pp. 317–323.

Utz H., Sablatn¨og S., Enderle S., and Kraetzschmar G. 2002. *Miro - middleware for mobile robot applications.* IEEE Transactions on Robotics and Automation, 18(4), August 2002.

Brooks A., Kaupp T., Makarenko A., Orebäck A. and Williams S. 2005. *"Towards Component-Based Robotics".* IEEE/RSJ International Conference on Intelligent Robots and Systems 2005.

Henning M., Spruiell M. 2004 *Distributed Programming with Ice,* ZeroC Inc., http://www.zeroc.com

Wang J., Su J., and Xi Y., 2001 *“COM-based software architecture for multisensor fusion system,”* Information Fusion, vol. 2, no. 4, pp. 261– 270,.

Corke P., Sikka P., Roberts J., E. Duff, *"DDX: A Distributed Software Architecture for Robotic Systems",* Australasian Conference on Robotics and Automation 2004

Montemerlo M., Roy N., and Thrun S.. *Perspectives on standardization in mobile robot programming: The carnegie mellon navigation (carmen) toolkit.* In IEEE/RSJ Intl. Workshop on Intelligent Robots and Systems, 2003.

URBI *http://www.urbiforge.com/*

Tikanmäki A., Röning J., Riekki J. 2003 *Remote operation of robotics systems using WLAN- and CORBA based architecture,* SPIE / Intelligent Robots and Computer Vision XXI, Oct 27 - 31, Rhode Island, Providence, RI USA, 2003

Mäenpää T., Tikanmäki A., Riekki J. and Röning J., , 2004 *A Distributed Architecture for Executing Complex Tasks with Multiple Robots,* IEEE 2004 ICRA, International Conference on Robotics and Automation,Apr 26 - May 1, New Orleans, LA, USA

Tikanmäki A., Röning J. 2004 *Advanced Remote Operation of swarms of Robots,* SPIE / Intelligent Robots and Computer Vision XXII, Philadelphia, USA

Tikanmäki A., Vallius T., Röning J., 2004 *Qutie - Modular methods for building complex mechatronic systems,* ICMA - International Conference on Machine Automation, Nov. 24.-26., Osaka, Japan

ON GENERATING GROUND-TRUTH TIME-LAPSE IMAGE SEQUENCES AND FLOW FIELDS

Vladimír Ulman and Jan Hubený
Centre for Biomedical Image Analysis, Masaryk University, Botanická 68a, Brno 602 00, Czech Republic
xulman@fi.muni.cz, xhubeny@fi.muni.cz

Keywords: Optical flow evaluation, ground-truth flow field.

Abstract: The availability of time-lapse image sequencies accompanied with appropriate ground-truth flow fields is crucial for quantitative evaluation of any optical flow computation method. Moreover, since these methods are often part of automatic object-tracking or motion-detection solutions used mainly in robotics and computer vision, an artificially generated high-fidelity test data is obviously needed. In this paper, we present a framework that allows for automatic generation of such image sequences based on real-world model image together with an artificial flow field. The framework benefits of a two-layered approach in which user-selected foreground is locally moved and inserted into an artificially generated background. The background is visually similar to the input real image while the foreground is extracted from it and so its fidelity is guaranteed. The framework is capable of generating 2D and 3D image sequences of arbitrary length. A brief discussion as well as an example of application in optical microscopy imaging is presented.

1 INTRODUCTION

The growing importance of image processing methods is unquestionable in the field of automation and robotics. Especially, optical flow computation methods are often involved in solutions adopted in, for instance, autonomous systems and agents, vehicle control applications, surveillance or live-cell microscopy. The outcome of these methods is often not the final product. It is usually further analyzed by object-tracking or motion-detection methods (Cédras and Shah, 1995; Gerlich et al., 2003; Eils and Athale, 2003).

Important is also thorough testing of a particular method before its application. This is even more evident due to the continuous development of image acquisition devices, since the usability of given image processing method is changing with the nature of examined image data. Verification is, therefore, an obvious need.

For the purpose of fully automatic testing one has to have a large data set together with correct results prepared. Or, the dataset should be generated online reasonably fast. A dataset consisting of real images is obviously the ideal choice. Unfortunately, the real images do not explicitly provide the ground-truth information about the motion expressed in the data.

There exist methods that extract such motion information, for example other, than currently tested, optical flow methods (Horn and Schunck, 1981; Barron et al., 1994) or image registration techniques (Zitová and Flusser, 2003). Unfortunately, these practically always incur some sort of error or imprecision in the flow field. The same holds for manually processed data (Webb et al., 2003), not mentioning the tedious extraction of ground-truth motion information.

We decided to automatically generate vast amount of artificial test images with the stress on their near-perfect visual similarity to the real data of the application in mind. The aim was to confidently test the reliability of the given optical flow computation method using this data. Moreover, we wanted to generate image sequences together with associated flow fields reasonably fast (i.e. faster than the execution of an optical flow computation method) to be able to simulate the behaviour of real-time decision system incorporating optical flow computation. Tracking of a live cell in microscopy can be taken as an example of such a decision-making system. Due to technology limits the cell can be acquired with only restricted (small) surroundings and needs on-line 2D or 3D tracking (moving the stage with the cell based on its motion).

The next section gives a motivation to the adopted solution by means of brief overview of possible approaches. The third section describes the proposed framework which automatically generates 2D and 3D image sequences of arbitrary length. It is followed by the section in which behaviour and sample image data for the case of optical microscopy is presented. A 3D image is considered as a stack of 2D images in this paper.

2 MOTIVATION

Basically, there are just two possible approaches to obtain image sequences with ground-truth flow fields. One may inspect the real data and manually determine the flow field. Despite the bias (Webb et al., 2003) and possible errors, this usually leads to a tedious work, especially, when inspecting 3D image sequences from a microscope. The other way is to generate sequences of artificial images from scratch by exploiting some prior knowledge of a generated scene. This is usually accomplished by taking 2D snapshots of a changing 3D scene (Galvin et al., 1998; Barron et al., 1994; Beauchemin and Barron, 1995). The prior knowledge is encoded in models which control everything from the shape of objects, movements, generation of textures, noise simulation, etc. (Lehmussola et al., 2005; Young, 1996). This may involve a determination of many parameters as well as proper understanding of the modeled system. Once the two consecutive images are created, the information about movement between these two can be extracted from the underlying model and represented in a flow field.

We have adopted the approach in which we rather modify an existing real sample image in order to generate an image sequence. This enabled us to avoid most of the modeling process as we shall see later. Moreover, we could easily create a flow field we were interested in. Consecutive images from the sample image could be then transformed by using either backward or forward transformations (Lin and Barron, 1994). Both transformations are possible. Nevertheless, we observed that forward transformation was substantially slower. Hence, we described the framework based only on backward transformation in this paper.

The backward transformation moves the content of an image with respect to the input flow field. The flow field assigns a vector to each voxel in the image. When generating a sequence, the voxel value is expected to move along its vector into the following image. The backward transformation works in the opposite direction: the preceding image is always created. Basically, the voxel at vector's origin is fetched into an output image from vector's end in the input image. An interpolation in voxel values often occurs due to real numbers in vectors.

A few drawbacks of the backward transformation must be taken into account when used. Owing to the interpolation the transformed image is blurred. The severity depends on the input flow field as well as interpolation method used. Moreover, the blur becomes more apparent after a few iterative transformations of the same image. Thus, the number of transformations should be as low as possible. Another issue appears when the flow field is not continuous. In this case, two (or even more) vectors may end up in the same position which copies the same voxel into distinct places in the output image. Unfortunately, non-continuous flow field is the case when local movements are to be simulated. Both drawbacks are demonstrated in the example in Fig. 1.

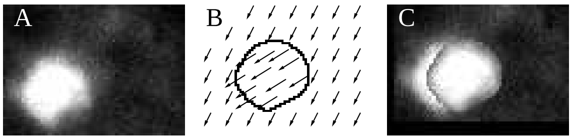


Figure 1: Backward transformation. A) An input image. B) Visualization of the input flow field with two homogeneous regions. C) A transformed image. Notice the blurred corona as well as the partial copy of the moved object. Images were enhanced to be seen better.

3 THE FRAMEWORK

In this section we described the framework based on two-layered component-by-component backward transformation. The input to the framework was an real-world sample image, a background mask, a foreground mask and a movements mask. The background mask denoted what, as a whole, should be moved in the sample image. The foreground mask denoted independent regions (components) in the sample image that were subjects to local movements. The movements of components had to remain inside the movements mask. The output of the framework was a sequence of artificially generated images together with appropriate flow fields. The schema of the framework is displayed in Fig. 2. The foreground and background masks were results of advanced segmentation method (Hubený and Matula, 2006) which was

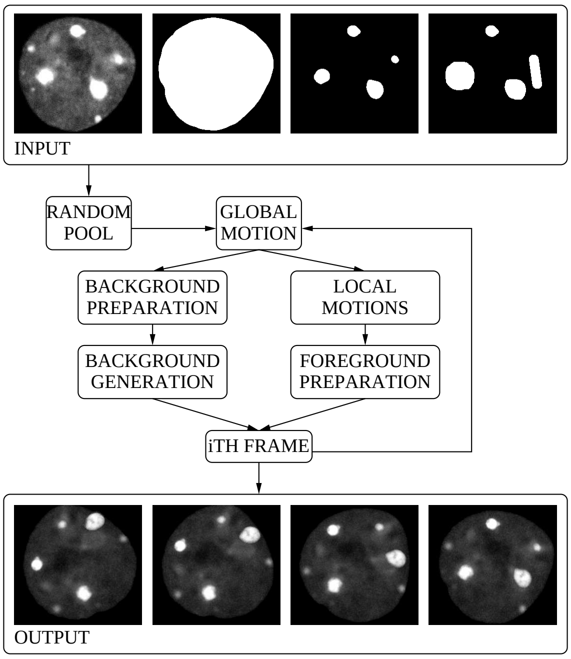


Figure 2: The schema of the framework. From left to right in INPUT: sample image, background mask, foreground mask, movements mask; in OUTPUT: examples of 1st, 10th, 20th and 30th image of a generated sequence, respectively. Images were enhanced to be seen better.

initiated with manually thresholded mask images.

The framework was aimed against two obstacles. Firstly, when foreground component was moved away from its place, the empty region had to be filled in. Therefore, only image regions corresponding to foreground components could be directly used. The whole background had to be artificially generated. Secondly, several transformations of the same image data was not acceptable. In order to generate long sequences without increasing blur in generated images, we developed a concept of generating *i*th image directly from the sample image instead of the $i+1$th image. Last but not least, owing to the backward transformation property the framework generated image sequence from the last to the first image.

The *random pool* was the initiating step of the process. Here, specific voxels from the sample image were collected and stored into a separate image. Voxels had to lay inside the background mask and outside the foreground mask. The mean intensity value μ over these voxels was computed. The selection was then even restricted. In the separate image, the pool, remained only such voxel which intensity value was inside the interval $(\mu-\sigma, \mu+k\sigma)$ where σ and k were supplied manually. We set $\sigma = 11$ and $k = 3/2$ to fit the real histogram better. This will probably change when different sort of images is generated.

A simulation of some global movement of the entire sample image was achieved in the unit *global motion*. In this unit, the flow field for the *i*th frame was formed for the first time. The foreground and background masks as well as movements mask were transformed according to this flow field. There was a zero flow field created when processing the last image of the sequence, i.e. the first image created by the framework.

The generation of background started from preparation of the sample image. The sample image had to be positioned to fit the background mask. We made use of a special flow field for that purpose. This flow field was concatenated to the flow created in the previous unit and the result was kept until the next sequence image is considered. Copy of sample image was transformed according to this special flow field. Note that the backward transformation may be used for concatenation of flow fields if we store flow vector's elements in separate images. The concatenated flow is transformed and added to the flow.

A new similar background was created in two steps. The foreground components were removed from the transformed copy of sample image and the holes were filled in as described in Fig. 3. The result was filtered in order to estimate local averages. We used the 9×9 filter of ones which still reflected local intensity values sensitively, yet the filtered image was smooth. In the second step, the background mask was filled in with randomly chosen values from the pool created in the *random pool* unit. Finally, corresponding local average subtracted to the value of the mean μ was added to each voxel within the background mask. This ensured the high fidelity of the generated texture.

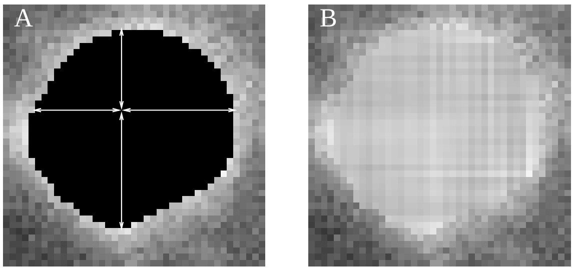


Figure 3: The filling of removed foreground regions. A) For each examined voxel, nearest voxel in each direction outside the black region is found and the distance is determined. A weighted average of 1/distance-based values is supplied. B) The result of such filling.

The foreground mask was first decomposed into independent components in the *local motions* unit. Each one is treated separately. A translating motion vector was randomly chosen from all such vectors that keep the component within the movement mask. We also made use of user supplied parameter for maximum allowed length of motion vector which enabled us to control the magnitude of independent local motions. A temporary flow field was created and uniformly filled with this vector. The mask of this component only and a copy of the *i*th flow field were transformed according to this uniform flow. This moved the component mask and prepared the concatenation of the corresponding region of the flow field. The concatenation was finished by pasting this region followed by addition of chosen flow vector to each vector inside the region into the *i*th flow. Note that more complex foreground movement may be used by substituting any smooth flow field for the uniform one as well as corresponding vectors should be added instead of constantly adding the chosen one. After all, new foreground mask was created by merging all single locally moved masks.

In the *foreground preparation* unit, similarly to the background preparation, another special flow field was used. It was again concatenated to the current *i*th flow and the result was stored for the next framework's iteration. Another copy of sample image was transformed according to this another special flow field to position the foreground texture.

In the *ith frame* unit, the foreground regions were extracted from the transformed copy of sample images. The extraction was driven by the new foreground mask which was dilated (extended) only for that purpose beforehand. Finally, the *i*th image was finished by weighted insertion (for details refer to (Ulman, 2005)) of the extracted foreground into the artificially generated background. The weights were computed by thresholding the distance transformed (we used (Saito and Toriwaki, 1994)) foreground mask. An illustration of the whole process is shown in Fig. 4.

4 RESULTS

We implemented and tested the presented framework in C++ and in two versions. The first version created only image pairs while the second version created arbitrarily long image sequences. It was implemented with both backward and forward transformations. We observed that for 2D images the forward variant was up to two orders of magnitude slower than the backward variant. Therefore, the second version was implemented based only on backward transformation. The program required less then 5 minutes on Pentium4 2.6GHz for computation of a sequence of 50 images with 10 independent foreground regions.

The generator was tested on several different 2D real-world images and one real-world 3D image. All generated images were inspected. The framework generates every last image in the sequence as a replacement for the sample image. Thus, we computed correlation coefficient (Corr.), average absolute difference (Avg. diff.) and root mean squared (RMS) differences. The results are summarized in Table 1. The generator achieved minimal value of 0.98 for correlation. This quantitatively supports our observations that generated images are very close to their originals. The suggested framework also guarantees exactly one transformation of the sample image, hence the quality of the foreground texture is best possible thorough the sequence. Refer to Fig. 5 for example of 3 images of a 50 images long sequence. A decent improvement was also observed when an artificial background of 3D image was formed in a slice-by-slice manner, see rows C and D in Table 1. In the case of row D, a separate random pools and mean values were used for each slice of the 3D image.

Inappropriately created foreground mask may emphasize the borders of extracted foreground when inserted into artificial background. The weighted foreground insertion was observed to give visually better results. Table 1 quantitatively supports our claim: merging the foreground components according to twice dilated foreground mask was comparable to the plain overlaying of foreground components according to non-modified masks.

The use of user-supplied movements mask prevented the foreground components from moving into regions where there were not supposed to appear, e.g. outside the cell. The masks are simple to create, for example by extending the foreground mask into demanded directions. The generated sequences then became even more real. Anyway, randomness of components' movements prohibited their movements consistency. Pre-programming the movements would enable the consistency. Clearly, the movement mask wouldn't be necessary in this case.

5 CONCLUSION

We have described the framework for generating time-lapse pseudo-real images together with unbiased flow fields. The aim was to automatically generate a large dataset in order to automatically evaluate methods for optical flow computation. However, one may discard the generated flow fields and use just the image sequence.

The framework allows for the synthesis of 2D and 3D image sequences of arbitrary length. By suppling real-world sample image and carefully created masks for foreground and background, we could force im-

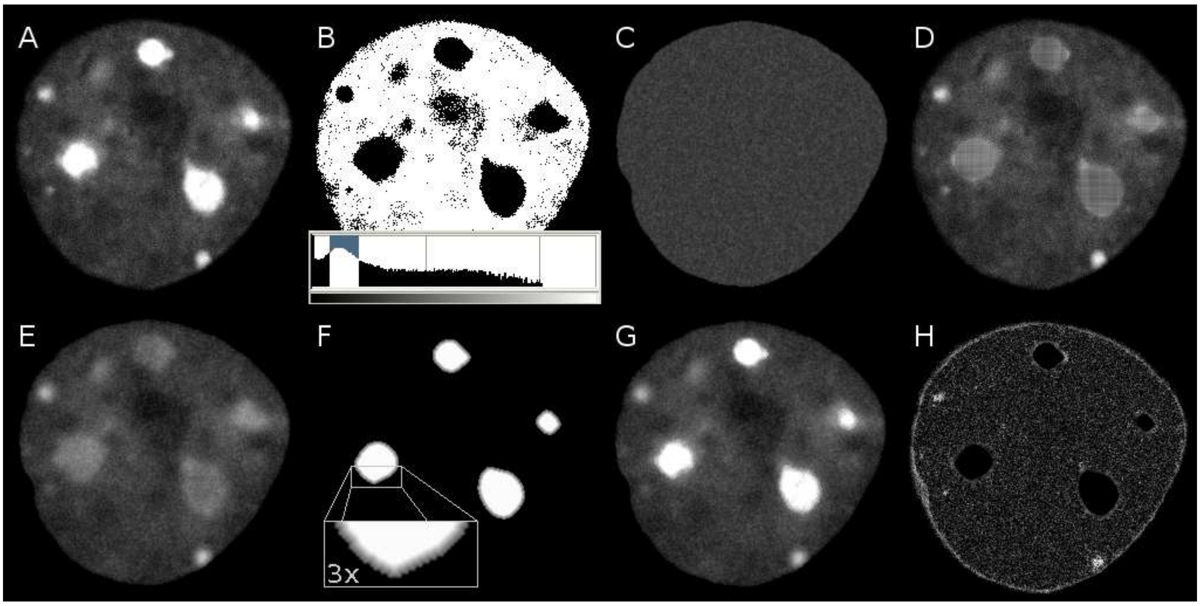


Figure 4: Example of image formation. A) A sample image. B) Its intensity histogram and thresholded image with thresholds set as shown in the histogram. C) The background filled with randomly chosen values. D) The sample image with foreground regions filled in. E) The same image after the averaging filter. F) The weights used together with the extended foreground mask, brighter intensity shows higher weight. G) The artificial image (the last image in the sequence). H) A map of intensity differences between A) and G), maximum brightness is at value of 30. Note that the highest errors were due to erroneous segmentation of the background. All images were enhanced for the purpose of displaying.

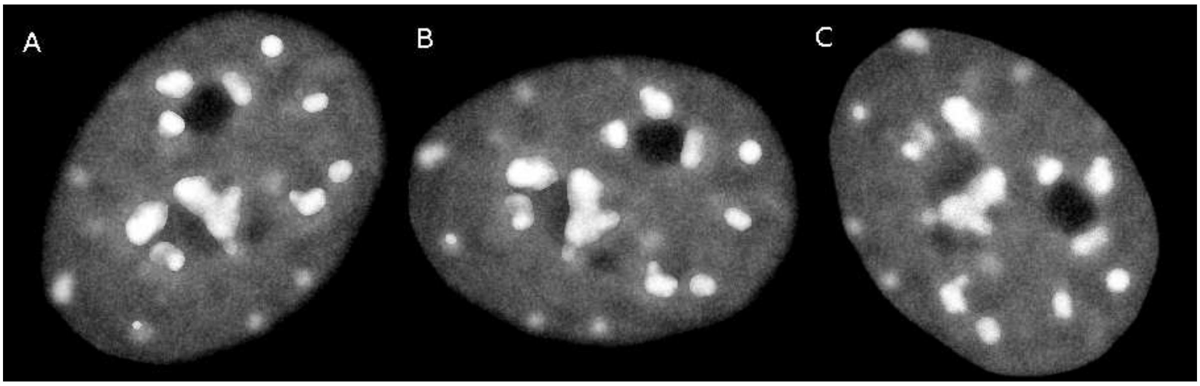


Figure 5: Example of 3 frames from image sequence. A) The first frame (the last generated). B) The middle frame. C) The last frame (the first generated). All images were enhanced for the purpose of displaying.

ages in the sequence to look more realistic. We made use of rotation and translation transformations for global motion (of the entire cell) and only translations for independent local movements of foreground components (selected intracellular structures). The transformations used can be arbitrary, owing to the formalism of the flow field, provided they are continuous because of limitation of both transformation methods. Seamless overlaying of the foreground was achieved by the weighted insertion of foreground which improved the robustness to any imprecision in the foreground segmentation. We also made use of local movements mask which gave us ultimate control over the independent foreground movements.

We believe that the framework is applicable to other fields as well. In some circumstances, image processing subroutines may differ as well as different foreground movements may be desired. The requirement is that images should be separable into just two layers and that the background should be reasonably easy to generate. For instance, in the vehicle control applications one may meet the requirement: observ-

Table 1: Similarity comparison from several aspects. The column "Ext." shows the number of dilations performed on the input foreground mask beforehand. The upper indices denote whether the foreground regions were simply overlaid^1 or merged^2 into the background. A) and B) Comparisons over 2D images. C) Comparison over a 3D image. D) Comparison over the same 3D image, separate pools of voxel intensities were used for each slice during the formation of the artificial background.

	Ext.	Corr.^{1}	Avg. diff.^{1}	RMS^1	Corr.^{2}	Avg. diff.^{2}	RMS^2
A	0	0.989	3.87	5.13	0.989	3.87	5.12
	1	0.989	3.80	5.03	0.989	3.85	5.05
	2	0.989	3.73	4.94	0.989	3.82	5.00
	3	0.989	3.68	4.90	0.989	3.83	4.98
B	0	0.992	2.76	3.83	0.992	2.77	3.85
	1	0.992	2.62	3.69	0.992	2.74	3.75
	2	0.993	2.41	3.46	0.992	2.62	3.58
	3	0.993	2.33	3.40	0.992	2.64	3.57
C	0	0.980	3.67	4.79	0.980	3.67	4.79
	1	0.980	3.73	4.89	0.980	3.81	4.92
	2	0.981	3.53	4.69	0.981	3.70	4.77
	3	0.981	3.42	4.59	0.981	3.66	4.72
D	0	0.982	3.15	4.16	0.982	3.16	4.17
	1	0.983	3.07	4.08	0.982	3.13	4.11
	2	0.983	3.00	4.03	0.983	3.11	4.08
	3	0.984	2.92	3.96	0.983	3.10	4.05

ing an image of a car on the road can be split into the car foreground and rather uniform road background.

ACKNOWLEDGEMENTS

The presented work has been supported by the Ministry of Education of the Czech Republic (Grants No. MSM0021622419, LC535 and 2B06052).

REFERENCES

Barron, J. L., Fleet, D. J., and Beauchemin, S. S. (1994). Performance of optical flow techniques. *Int. J. Comput. Vision*, 12(1):43–77.

Beauchemin, S. S. and Barron, J. L. (1995). The computation of optical flow. *ACM Comput. Surv.*, 27(3):433–466.

Cédras, C. and Shah, M. A. (1995). Motion based recognition: A survey. *Image and Vision Computing*, 13(2):129–155.

Eils, R. and Athale, C. (2003). Computational imaging in cell biology. *The Journal of Cell Biology*, 161:447–481.

Galvin, B., McCane, B., Novins, K., Mason, D., and Mills, S. (1998). Recovering motion fields: An evaluation of eight optical flow algorithms. In *In Proc. of the 9th British Mach. Vis. Conf. (BMVC '98)*, volume 1, pages 195–204.

Gerlich, D., Mattes, J., and Eils, R. (2003). Quantitative motion analysis and visualization of cellular structures. *Methods*, 29(1):3–13.

Horn, B. K. P. and Schunck, B. G. (1981). Determining optical flow. *Artificial Intelligence*, 17:185–203.

Hubený, J. and Matula, P. (2006). Fast and robust segmentation of low contrast biomedical images. In *In Proceedings of the Sixth IASTED International Conference VIIP*, page 8.

Lehmussola, A., Selinummi, J., Ruusuvuori, P., Niemisto, A., and Yli-Harja, O. (2005). Simulating fluorescent microscope images of cell populations. In *IEEE Engineering in Medicine and Biology 27th Annual Conference*, pages 3153–3156.

Lin, T. and Barron, J. (1994). Image reconstruction error for optical flow. In *Vision Interface*, pages 73–80.

Saito, T. and Toriwaki, J. I. (1994). New algorithms for Euclidean distance transformations of an *n*-dimensional digitized picture with applications. *Pattern Recognition*, 27:1551–1565.

Ulman, V. (2005). Mosaicking of high-resolution biomedical images acquired from wide-field optical microscope. In *EMBEC'05: Proceedings of the 3rd European Medical & Biological Engineering Conference*, volume 11.

Webb, D., Hamilton, M. A., Harkin, G. J., Lawrence, S., Camper, A. K., and Lewandowski, Z. (2003). Assessing technician effects when extracting quantities from microscope images. *Journal of Microbiological Methods*, 53(1):97–106.

Young, I. (1996). Quantitative microscopy. *IEEE Engineering in Medicine and Biology Magazine*, 15(1):59–66.

Zitová, B. and Flusser, J. (2003). Image registration methods: a survey. *IVC*, 21(11):977–1000.

HELPING INSTEAD OF REPLACING
Towards A Shared Task Allocation Architecture

Foad Ghaderi and Majid Nili Ahmadabadi
Control and Intelligent Processing Center of Excellence, Mobile Robot Lab
Dept. of Elect. and Comp. Eng., Faculty of Eng., University of Tehran
f.ghaderi@ikcard.com, mnili@ut.ac.ir

Keywords: Cooperative robotics, shared task allocation, fault tolerance.

Abstract: Some failures cause the robots to loss parts of their capabilities, so that they cannot perform their assigned tasks. Considering requirements of typical robotic teams during different missions, a distributed behavior based control architecture is introduced in this paper. This architecture is based on an enhanced version of ALLIANCE, and provides the robots the ability of performing shared tasks based on help requests. The architecture contains a mechanism for adaptive action selection and a communication protocol for information and task sharing which are required for coordination of team members. The proposed architecture is used in a box pushing mission where heterogeneous robots push several boxes with different masses.

1 INTRODUCTION

Sensitivity of distributed robotic systems to changes in working environments and common failures in their mechanical and electrical components is a barrier to their wide physical implementation (Ahmadabadi, 2001). So a dynamic task allocation mechanism that supports fault tolerance behavior is a mandatory requirement for cooperative robotics solutions (Ghaderi, 2002).

Mataric et al. showed empirically that there is no optimal task allocation strategy for all domains, and even it is difficult to identify the optimal task allocation strategy for a particular task (Mataric, 2003). In their framework (Gerkey, 2003) only one robot could be assigned to a task, and no redundancies were allowed.

In (Ahmadabadi, 2004) two distributed and cooperative methods for load reallocation among some object lifting robots without requiring them to change their grasp positions are introduced.

Vig et al. provided RACHNA (Vig, 2005) which is a market-based architecture for allocating multi-robot tasks based on individual robot capabilities. In this architecture, if a robot can not support the team, it is more likely that another robot with similar capabilities replaces this robot.

Some advanced behavior based control architectures are introduced that allow adaptive task allocation. Parker's architecture (ALLIANCE) is based on Subsumption architecture (Brooks, 1986). It supports adaptive action selection for faulty robot replacement (Parker, 1998), (Parker, 1994).

Kasahara has considered reorganization in an organizational learning model (Kasahara, 1998). Whenever a fault occurs, agents try to compose new organizations using learning techniques. This process can lead the team to its goal.

In this paper, a new task allocation method is introduced which is based on ALLIANCE and supports help in a team of cooperative robots. In this method partially faulty robots do not leave the team, but the team tries to redistribute task among other members in order to use the whole capabilities of the robots. The suggested architecture supports adaptive action selection with help request processing and allows the group to perform its mission coordinated.

In the next section importance of help in a cooperative team of robots is discussed. The help supporting method is introduced in section 3. Box pushing problem is reported in section 4. Conclusions are presented in section 5.

2 HELP IMPORTANCE IN COOPERATIVE MISSIONS

Most of the real world mobile robots applications are performed in dynamic environments. Missions of rescue robots, cleaning robots, robots used for defense purposes and every other mission that robots are used instead of human in order to decrease dangers are examples of these applications. In these applications a lot of changes may occur in environment by time, and sometimes these changes cause a team to fail in some (or all) of its tasks. If the task selection mechanism is static and without flexibility, then there is no way to complete the tasks, except waiting for some robots to complete their tasks and replacing them with those robots that are not able to continue. Here we refer to an action selection mechanism as static when robots do not change their tasks even if there are some tasks in the team with higher priorities.

In this case if there is redundancy in the team, higher priority tasks are assigned to idle robots, otherwise these tasks would be assigned after a time interval that is unknown. Obviously such a task assignment may cause a disaster for the group or environment if the critical tasks are not assigned and performed in an acceptable time. We define that a task in a cooperative mission is unassigned, if no robots have selected it ever, or if the selecting robot(s) is (are) not able to perform the task.

Disability of a robot in performing a task may have two reasons. First, the robot is faulty and hence it can not complete the task, and second the robot is not faulty but its capabilities are not enough to complete the task.

Sometimes replacing the disabled robot with a new one provides a new chance to the team to achieve its goal. If there is redundancy in quantity of the robots, problem is solved easily, and otherwise one of the robots must ignore its task and perform the uncompleted task. But there are some situations that this method does not acquire mission's goals.

We divide these missions into two categories:

❒ **None of the team members can finish the uncompleted task alone.** In the other words, performing the task requires efforts of more than one of the existing robots and the task is not divisible to simpler subtasks. This kind of tasks is called "shared task" which requires closed coordination and real time action of a team of robots. Transferring of an injured person in a rescue mission is a good example of shared tasks. In this mission, none of the robots can complete the task alone and since some member's actions may disturb efforts of the others, a close coordination between robots is mandatory.

❒ **Performing the task depends on some robot's capabilities.** These capabilities may include some special mechanical mechanisms, processing power, knowledge and etc. Therefore, the replacement would be possible if there is another robot with at least the same capabilities.

It is clear that the traditional way of replacing robots will not have considerable effect in performance of the team. In this case, the best way to achieve the group's goal is helping the weak robots.

3 THE HELP SUPPORTING METHOD

There are two important issues to be considered in any architecture that supports help for faulty agents. Let's review a scenario first. Assume that some robots are cooperating to perform a mission. During their action, one of them senses that it is not possible to complete its task lonely and so broadcasts a help request. Receiving this message, other robots must decide about helping disabled robot. After making decision, the helping robots must cooperate to complete the disabled robot's task. During lots of cooperative help tasks a closed coordination between robots is required. So the help supporting architecture must include appropriate mechanisms to do action selection and closed coordination for a cooperative task.

3.1 Action Selection

The way in which robots process the help request and make decision has great effects on functionality and performance of the team. In addition, there are many factors to be considered when processing a help request. Some of them are listed below:

- Distance to the disabled robot (in general the cost to reach to the position in which the helper robot can help the disabled robot),
- Cost of performing the task,
- Having useful mechanical capabilities, knowledge and experience,
- Priority of robot's current task compared with the shared task,
- Criticality of the disabled robot's task,
- Progress of the task of the team members (including the faulty robot).

These are general parameters that can affect performance of a help process. Besides these, other parameters might be chosen according to specific nature of the mission.

We used ALLIANCE architecture (Parker, 1998) as a base for adaptive action selection. This behavior based control architecture has adaptive action selection capabilities that can be used in different missions, but this architecture does not support help. So we enhanced ALLIANCE to address the mentioned requirements.

Considering motivation function and it’s parameters in ALLIANCE architecture, it’s apparent that if a robot selects a task, (except in some special conditions), it will not ignore it till the task is finished. So while a robot is performing a task and receives a help request it is not possible to change its behavior and select a new task. In this case the help request is not answered if there is no idle robot in the team.

We supposed that each task is allowed to be shared just between two robots. In order to achieve adaptive action selection, some changes were applied to ALLIANCE. In fact, we designed a two dimensional motivation vector for each robot. Two axes of this vector represent team members and tasks of the team. So m_{ijk} in robot i is motivation function corresponding to behavior j, while robot k broadcasts a help request. The new motivation function for the shared task j in robot i is defined by the following equation:

$$\begin{aligned} & \mathrm{m}_{\mathrm{ijk}}(0)=0 \\ & \mathrm{m}_{\mathrm{ijk}}(\mathrm{t})=[\mathrm{m}_{\mathrm{ijk}}(\mathrm{t}-1)+\mathrm{impatience}_{\mathrm{ijk}}(\mathrm{t})] \\ & \qquad \times \mathrm{sensory\_feedback}_{\mathrm{ijk}}(\mathrm{t}) \\ & \qquad \times \mathrm{activity\_suppression}_{\mathrm{ij}}(\mathrm{t}) \\ & \qquad \times \mathrm{impatience\_reset}_{\mathrm{ijk}}(\mathrm{t}) \\ & \qquad \times (1+\mathrm{Help\_Request}_{\mathrm{ijk}}(\mathrm{t})) \\ & \qquad \times \mathrm{acquiescence}_{\mathrm{ij}}(\mathrm{t}) \end{aligned} \tag{1}$$

The parameters are defined below:

Impatience$_{ijk}$(t): This parameter shows robot's impatience for interfering other robot's tasks. If task j is not selected by any other robot, *Impatience* will have a large value; otherwise its quantity depends on the type of task, environmental conditions, and progress of the task. In shared task, some more factors such as the criticality of the task, the efficiency of the robot in performing the task and type of the help request are important.

Sensory_feedback$_{ijk}$(t): This parameter indicates whether behavior j is required at the time to reach the goals of the team or not. Often, physical robots sensors produce this information, but in practical robotics applications it's possible to use virtual sensors such as memory (as a flag), and communication lines to indicate this parameter. If robot k needs some help to complete task j and robot i can cooperate in this task, *sensory_feedback* is one, otherwise it is equal to zero.

Activity_Suppression$_{ij}$(t): When a behavior is active in a robot, this parameter suppresses behaviors with lower priorities to be activated. This suppression is not applied to higher priority tasks and so permits the robot to change its active behavior in order to increase team's efficiency or help other robots in critical tasks.

If priority of task j is higher than active behavior in robot i, this parameter is equal to 1, otherwise it's 0. In Parker's architecture (Parker, 1994), (Parker, 1998), the criticality of the tasks is not taken into account in evaluating this parameter. So if any behavior is active, others would be suppressed, regardless of the criticality of the others.

Impatience_Reset$_{ijk}$(t): The default value of this parameter is 1. When a robot selects behavior j to help robot k, it broadcasts its decision. As a result, *Impatience_Reset* in the other robots becomes zero for a defined period of time, and then increases to 1 again, and the robots motivation to participate in task j resets consequently.

Help_request$_{ijk}$(t): Default value of this parameter is zero and takes a positive value for a predefined interval, whenever robot i receives a help request from robot k requesting to cooperate in task j. If in this period the value of *impatience_reset* is not zero, the motivation value will be multiplied by a value bigger than one. (Notice that faulty robot broadcasts help request periodically until its task is finished.)

Acquiesence$_{ij}$(t): The same as that in ALLIANCE architecture.

Initial values of the parameters and their changes deeply depend on the general requirements of the mission, robots type and different performance indexes used. Therefore, the initial values must be optimized for each task.

3.2 Robots Coordination in Help Process

After deciding to help, robots must coordinate to perform the shared task. The coordination strategy depends on the special characteristics of the task and there is no general method to do this. But, the coordination methods are generally based on the information about activities of the other robots. Then, we used the communication method used in (Asama, 1992) to obtain information about others tasks, activities, and their progress in order to coordinate the robots.

In Asama’s protocol, nine procedures are defined to communicate information (Asama, 1992). We selected seven of them to coordinate the team:

- Report for ready state,
- Declaration of task selection,
- Request for Cooperative task,
- Acceptance of Cooperative task,
- Report of completion of preparation for cooperation,
- Command for start of cooperative task,
- Report of task completion.

4 EVALUATING THE ARCHITECTURE

Task assignment and adaptive behavior of a cooperative team that uses help mechanism is evaluated in a box pushing problem. Box pushing is a common test bed in the field of cooperative robotics. It is assumed that some boxes are distributed in an environment and some robots must push them to the front wall.

There are two kinds of boxes. Some of them are light and can be moved by a single robot. Others are heavy such that one robot is not able to transfer them alone. Each robot selects a box to transfer while it has no information about weight of the box. Whenever the robot detects that the selected box is heavy and it's not possible to move it alone, it will broadcast a help request.

Robots have some inexact information about the position of the boxes, so they must search for them. At the beginning, the robots assume that all of the boxes are light. So after selecting a box, the robot goes towards it and tries to move it. If the box is heavy, the robot broadcasts a help request, and waits for other's responses.

Experiments show that the team can manage existing resources to complete the mission in cases that some robots are not able to perform their assigned task.

5 SUMMARY

In this paper, we have introduced a help supporting architecture that focuses on task allocation in cases that some of the team members are not able to complete their tasks. This architecture supports fault tolerance in cooperative missions that have various tasks with different criticalities. In this method, the team tries to redistribute the tasks among members by processing help requests from disabled robots in order to use all robots capabilities. The suggested architecture supports adaptive action selection and let's the group to perform its mission in cooperation. In our method the robots are committed unless some critical tasks are not assigned and they are individualistic unless some robots require help. The architecture is evaluated in a box pushing mission and results show acceptable performance.

ACKNOWLEDGEMENTS

This work is partially supported by Control and Intelligent Processing Center of Excellence, University of Tehran.

REFERENCES

Ahmadabadi, M. N. and Nakano E., "A Constrain and Move approach to distributed object manipulation", *IEEE Trans. on Robotics and Automation*, Vol. 17, No. 2, pp. 157-172 , April 2001.

Ghaderi, F., Ahmadabadi, M. N. "Distributed Cooperative Fault Tolerance In A Team Of Object Lifting Robots", In *IEEE Proc. of 1996 Int. Conf. on Intelligent Systems and Robots*, 2002, pp. 2721-2727.

Mataric, M. J., Sukhatme, G. S., Østergaard, E. H., "Multi-Robot Task Allocation in Uncertain Environments", *Autonomous Robots* 14, 255–263, 2003, Kluwer Academic Publishers.

Gerkey, B. P., Mataric, M. J., "A Framework for studying multi robot task allocation", In *Multi-Robot Systems: From Swarms to Intelligent Automata*, Volume II, Pages 15-26, the Netherlands, 2003, Kluwer Academic Publishers.

Ahmadabadi, M. N., Ghaderi, F. "Distributed cooperative load redistribution for fault tolerance in a team of four object-lifting robots", *Advanced Robotics*, Vol. 18, No. 1, pp. 61–81, 2004.

Vig, L., Adams, J. A., "A Framework for Multi-Robot Coalition Formation" In *Proc. of the 2nd Indian International Conference on Artificial Intelligence*, 2005, India.

Brooks, R. A., "A Robust Layered Control System for a Mobile Robot", *IEEE Journal of Robotics and Automation*, Vol.2, No.1, March 1986 pp. 14-23.

Parker, L. E., "ALLIANCE: An Architecture for Fault tolerant Multi-robot Cooperation", *IEEE Trans. robotics and automation* vol. 14, No. 2, pp. 220-240, April 1998.

Parker, L. E., "Heterogeneous Multi-Robot Cooperation", *Ph.D. Thesis*, Massachusetts Institute of Technology, Cambridge, MA, Feb.1994.

Kasahara, H., Takadama, K. , Nakasuka, S., Shimohara, K.,"Fault Tolerance in a Multiple Robots Organization Based on an Organizaional Learning Model" *The IEEE 1998 International Conference On Systems, Man and Cybernetics (SMC'98)*, pp. 2261-2266, 1998.

Asama H., Ozaki K., Matsumoto A., Ishida Y., and Endo I. , "Development of task assignment system using communication for multiple autonomous robots", *Journal of Robotics and Mechatronics*, pp. 122-127, 1992.

FUZZY LOGIC ALGORITHM FOR MOBILE ROBOT CONTROL

Viorel Stoian and Cristina Pana
University of Craiova, Mechatronics Department, Decebal Street No. 107, Craiova, Romania
stoian@robotics.ucv.ro, cristina@robotics.ucv.ro

Keywords: Fuzzy logic algorithm, mobile robots, obstacole avoidance, trajectory controller.

Abstract: This paper presents a fuzzy control algorithm for mobile robots which are moving next to the obstacle boundaries, avoiding the collisions with them. Four motion cycles (programs) depending on the proximity levels and followed by the mobile robot on the trajectory (P1, P2, P3, and P4) are shown. The directions of the movements corresponding to every cycle, for every reached proximity level are presented. The sequence of the programs depending on the reached proximity levels is indicated. The motion control algorithm is presented by a flowchart showing the evolution of the functional cycles (programs). The fuzzy rules for evolution (transition) of the programs and for the motion on X-axis and Y-axis respectively are described. Finally, some simulations are presented.

1 INTRODUCTION

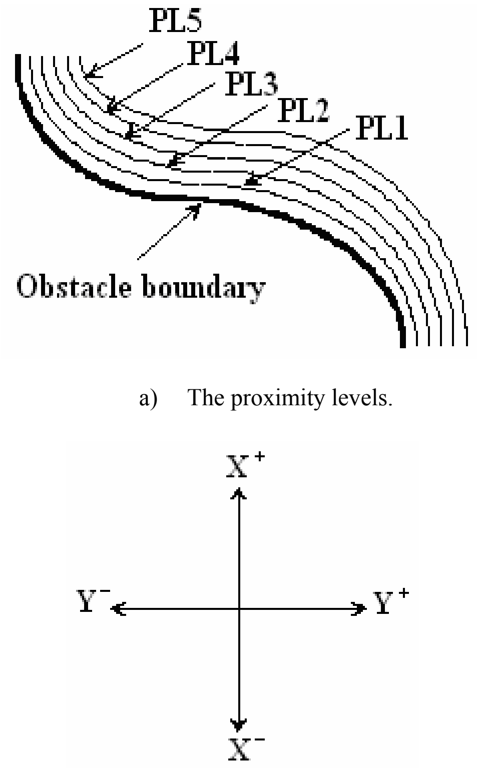
Fuzzy set theory, originally developed by Lotfi Zadeh in the 1960's, has become a popular tool for control applications in recent years (Zadeh, 1965).

Fuzzy control has been used extensively in applications such as servomotor and process control. One of its main benefits is that it can incorporate a human being's expert knowledge about how to control a system, without that a person need to have a mathematical description of the problem.

Many robots in the literature have used fuzzy logic (Song, 1992, Khatib, 1986, Yan, Ryan, Power, 1989 …). Computer simulations by Ishikawa feature a mobile robot that navigates using a planned path and fuzzy logic. Fuzzy logic is used to keep the robot on the path, except when the danger of collision arises. In this case, a fuzzy controller for obstacle avoidance takes over.

Konolige, et al. use fuzzy control in conjunction with modeling and planning techniques to provide reactive guidance of their robot. Sonar is used by robot to construct a cellular map of its environment.

Sugeno developed a fuzzy control system for a model car capable of driving inside a fenced-in track. Ultrasonic sensors mounted on a pivoting frame measured the car's orientation and distance to the fences. Fuzzy rules were used to guide the car parallel to the fence and turn corners (Sugeno et al., 1989).



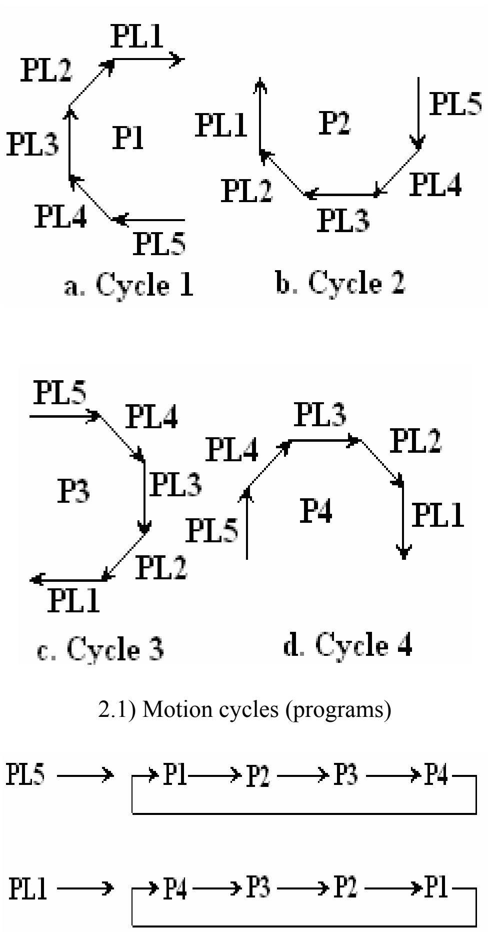
a) The proximity levels.

b) The two degrees of freedom of the locomotion system of the mobile robot.

Figure 1: The proximity levels and the degrees of freedom of the robot motion.

2 CONTROL ALGORITHM

The mobile robot is equipped with a sensorial system to measure the distance between the robot and object that permits to detect 5 proximity levels (PL): PL1, PL2, PL3, PL4, and PL5. Figure 1a presents the obstacle (object) boundary and the five proximity levels and Figure 1b presents the two degrees of freedom of the locomotion system of the mobile robot. This can move either on the two rectangular directions or on the diagonals (if the two degrees of freedom work instantaneous).



2.1) Motion cycles (programs)

2.2) The sequence of the programs

Figure 2: The sequences of the motion.

The goal of the proposed control algorithm is to move the robot near the object boundary with collision avoidance. Figure 2a shows four motion cycles (programs) which are followed by the mobile robot on the trajectory (P1, P2, P3, and P4). Inside every cycle are presented the directions of the movements (with arrows) for every reached proximity level. For example, if the mobile robot is moving inside first motion cycle (cycle 1 or program P1) and is reached PL3, the direction is on Y-axis (sense plus) (see Figure 1b, too). In Figure 2b we can see the sequence of the programs.

One program is changed when are reached the proximity levels PL1 or PL5. If PL5 is reached the order of changing is: P1→P2→P3→P4→ P1→

If PL1 is reached the sequence of changing becomes: P4→P3→P2→P1→ P4→

The motion control algorithm is presented in Figure 3 by a flowchart of the evolution of the functional cycles (programs). We can see that if inside a program the proximity levels PL2, PL3 or PL4 are reached, the program is not changed. If PL1 or PL5 proximity levels are reached, the program is changed. The flowchart is built on the base of the rules presented in Figure 2.1 and Figure 2.2.

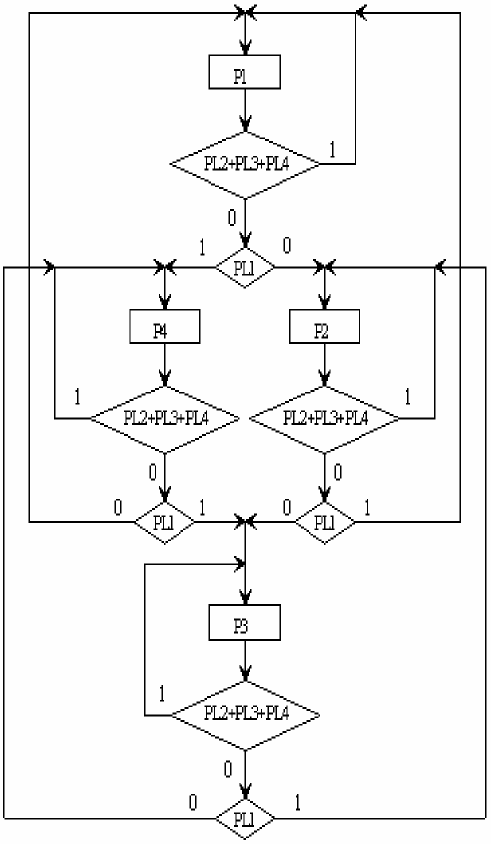


Figure 3: The flowchart of the evolution of the functional cycles (programs).

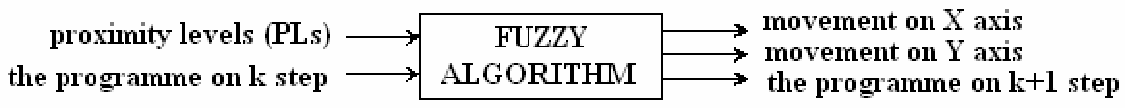
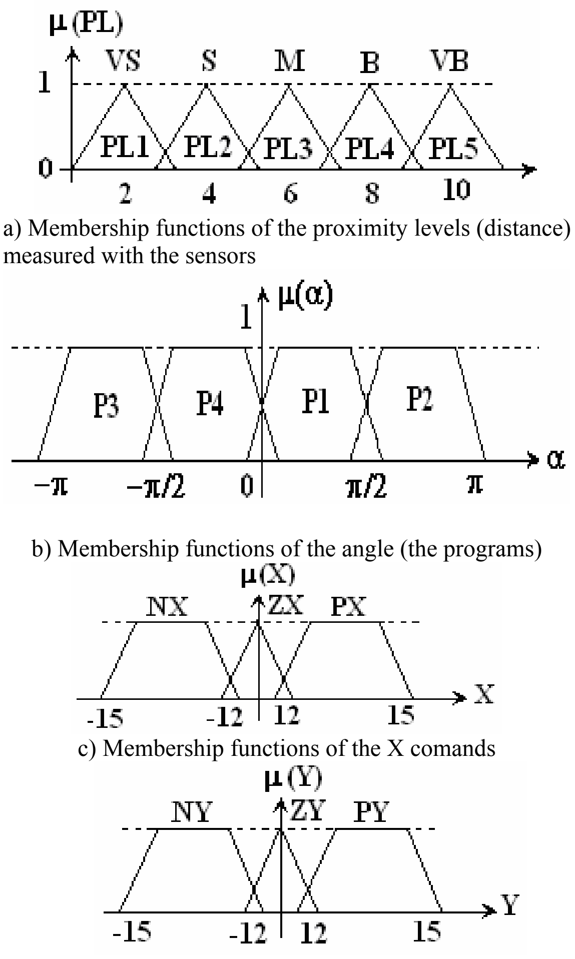


Figure 4: The inputs and outputs of the fuzzy algorithm.

3 FUZZY ALGORITHM



a) Membership functions of the proximity levels (distance) measured with the sensors

b) Membership functions of the angle (the programs)

c) Membership functions of the X comands

d) Membership functions of the Y comands

Figure 5: Membership functions of the I/O variables.

The fuzzy controller for the mobile robots based on the algorithm presented above is simple. Most fuzzy control applications, such as servo controllers, feature only two or three inputs to the rule base. This makes the control surface simple enough for the programmer to define explicitly with the fuzzy rules.

The above robot example uses this principle, in order to explore the feasibility of using fuzzy control for its tasks. Figure 4 presents the inputs (distance-proximity levels and the program on k step) and the outputs (movement on X and Y-axes and the program on k+1 step) of the fuzzy algorithm.

For the linguistic variable "distance proximity level" we establish to follow five linguistic terms: "VS-very small", "S-small", "M-medium", "B-big", and "VB-very big". Figure 5a shows the membership functions of the proximity levels (distance) measured with the sensors and Figure 5b shows the membership functions of the angle (the programs). If the object is like a circle every program is proper for a quarter of the circle.

Figure 5c and Figure 5d present the membership functions of the X, respectively Y commands (linguistic variables). The linguistic terms are: NX-negative X, ZX-zero X, PX-positive X, and NY, ZY, PY respectively.

Table 1: Fuzzy rules for evolution of the programs.

	VS	S	M	B	VB
P1	P4	P1	P1	P1	P2
P2	P1	P2	P2	P2	P3
P3	P2	P3	P3	P3	P4
P4	P3	P4	P4	P4	P1

Table 2: Fuzzy rules for the motion on X-axis.

	VS	S	M	B	VB
P1	PX	PX	ZX	NX	NX
P2	ZX	NX	NX	NX	ZX
P3	NX	NX	ZX	PX	PX
P4	ZX	PX	PX	PX	ZX

Table 3: Fuzzy rules for the motion on Y-axis.

	VS	S	M	B	VB
P1	ZY	PY	PY	PY	ZY
P2	PY	PY	ZY	NY	NY
P3	ZY	NY	NY	NY	ZY
P4	NY	NY	ZY	PY	PY

Table 1 describes the fuzzy rules for evolution (transition) of the programs and Table 2 and Table 3 describe the fuzzy rules for the motion on X-axis and Y-axis, respectively. Table 1 implements the sequence of the programs (see Figure 2.2 and Figure 3) and Table 2 and Table 3 implement the motion cycles (see Figure 2.1 and Figure 3).

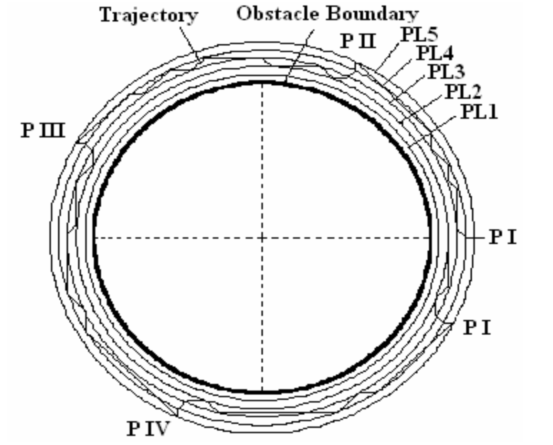


Figure 6: The trajectory of the mobile robot around a circular obstacle.

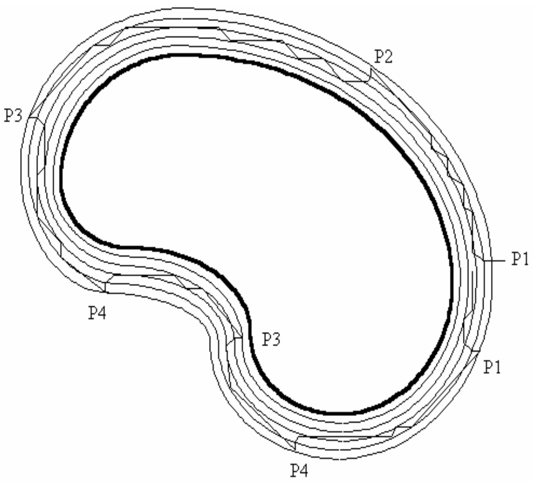


Figure 7: The trajectory of the mobile robot around a irregular obstacle.

4 SIMULATIONS

In the simulations can be seen the mobile robot trajectory around an obstacle (object) with circular boundaries (Figure 6) and around an obstacle (object) with irregular boundaries (Figure 7). One program is changed when are reached the proximity levels PL1 or PL5. If PL5 is reached the order of changing becomes as follows: P1→P2→P3→P4→... If PL1 is reached the order of changing is becomes follows: P4→P3→P2→P1→ P4→ ……

5 CONCLUSIONS

This paper presents a fuzzy control algorithm for mobile robots which are moving next to the obstacle boundaries, avoiding the collisions with them. Four motion cycles (programs) depending on the proximity levels and followed by the mobile robot on the trajectory (P1, P2, P3, and P4) are shown. The directions of the movements corresponding to every cycle, for every reached proximity level are presented. The sequence of the programs depending on the reached proximity levels is indicated. The motion control algorithm is presented by a flowchart showing the evolution of the functional cycles (programs). The fuzzy rules for evolution (transition) of the programs and for the motion on X-axis and Y-axis respectively are described**.** The fuzzy controller for the mobile robots based on the algorithm presented above is simple. Finally, some simulations are presented**.** If the object is like a circle, every program is proper for a quarter of the circle.

REFERENCES

Zadeh, L. A., 1965. Fuzzy Sets, *Information and Control*, No 8, pp. 338-353.

Sugeno, M., Murofushi, T., Mori, T., Tatematasu, T., and Tanaka, J., 1989. Fuzzy Algorithmic Control of a Model Car by Oral Instructions, *Fuzzy Sets and Systems,* No. 32, pp. 207-219.

Song, K.Y. and Tai, J. C., 1992. Fuzzy Navigation of a Mobile Robot, *Proceedings of the 1992 IEEE/RSJ Intern. Conference on Intelligent Robots and Systems*, Raleigh, North Carolina.

Khatib, O., 1986. Real-Time Obstacle Avoidance for Manipulators and Mobile Robots, *International Journal of Robotics Research*, Vol. 5, No.1, pp. 90-98.

Boreinstein, J. and Koren, Y., 1989. Real-time Obstacle Avoidance for Fast Mobile Robots, *IEEE Transactions on Systems, Man., and Cybernetics*, Vol. 19, No. 5, Sept/Oct. pp. 1179-1187.

Jamshidi, M., Vadiee, N. and Ross, T. J., 1993. *Fuzzy Logic and Control. Software and Hardware Applications*, PTR, Prentice Hall, New Jersey, USA.

Yan, J., Ryan, M., and Power, J., 1994. *Using Fuzzy Logic. Towards intelligent systems*, Prentice Hall, New York.

ON THE BALANCING CONTROL OF HUMANOID ROBOT

Youngjin Choi
School of electrical engineering and computer science, Hanyang University, Ansan, 426-791, Republic of Korea
cyj@hanyang.ac.kr

Doik Kim
Intelligent Robotics Research Center, Korea Institute of Science and Technology (KIST), Seoul, 136-791, Republic of Korea
doikkim@kist.re.kr

Keywords: WBC (whole body coordination), Humanoid robot, Posture control, CoM (center of mass) Jacobian.

Abstract: This paper proposes the kinematic resolution method of CoM(center of mass) Jacobian with embedded motions and the design method of posture/walking controller for humanoid robots. The kinematic resolution of CoM Jacobian with embedded motions makes a humanoid robot balanced automatically during movement of all other limbs. Actually, it offers an ability of WBC(whole body coordination) to humanoid robot. Also, we prove that the proposed posture/walking controller brings the ISS(disturbance input-to-state stability) for the simplified bipedal walking robot model.

1 INTRODUCTION

Recently, there have been many researches about humanoid motion control, for example, walking control(Choi et al., 2006; Kajita et al., 2001), and whole body coordination(Sugihara and Nakamura, 2002). Especially, the WBC(whole body coordination) algorithm with good performance becomes the essential part in the development of humanoid robot because it offers the enhanced stability and flexibility to the humanoid motion planning. In this paper, we suggest the kinematic resolution method of CoM Jacobian with embedded motions, actually, which offers the ability of WBC to humanoid robot. For example, if humanoid robot stretches two arms forward, then the position of CoM(center of mass) of humanoid robot moves forward and its ZMP(zero moment point) swings back and forth. In this case, the proposed kinematic resolution method of CoM Jacobian with embedded (stretching arms) motion offers the joint configurations of supporting limb(s) calculated automatically to maintain the position of CoM fixed at one point.

Also, we will simplify the dynamics of bipedal robot as the equation of motion of a point mass concentrated on the position of CoM. First, let us assume that the motion of CoM is constrained on the surface $z = c_z$, then the rolling sphere model with the concentrated point mass m can be obtained as the simplified model for bipedal robot as shown in Fig. 1. The motion of the rolling sphere on a massless plate is described by the position of CoM, $c = [c_x, c_y, c_z]^T$, and the ZMP is described by the position on the ground, $p = [p_x, p_y, 0]^T$. Second, let us take the moments about origin on the ground of the linear equations of motion for the rolling sphere (with a point mass = m) confined to motion on a plane $z = c_z$ as shown in Fig. 1, then the following equations are obtained:

$$\begin{aligned} \tau_x &= mgc_y - m\ddot{c}_y c_z && (1)\\ \tau_y &= -mgc_x + m\ddot{c}_x c_z && (2)\\ \tau_z &= -m\ddot{c}_x c_y + m\ddot{c}_y c_x && (3) \end{aligned}$$

where g is the acceleration of gravity, c_z is a height constant of constraint plane and τ_i is the moment about i-coordinate axis, for $i = x, y, z$. Now, if we introduce the conventional definition of ZMP as following forms:

$$p_x \triangleq -\frac{\tau_y}{mg} \quad \text{and} \quad p_y \triangleq \frac{\tau_x}{mg}$$

to two equations (1) and (2), then ZMP equations can be obtained as two differential equations:

$$p_i = c_i - \frac{1}{\omega_n^2}\ddot{c}_i \qquad \text{for } i = x, y \qquad (4)$$

where $\omega_n \triangleq \sqrt{g/c_z}$ is the natural radian frequency of the simplified biped walking robot system. Above

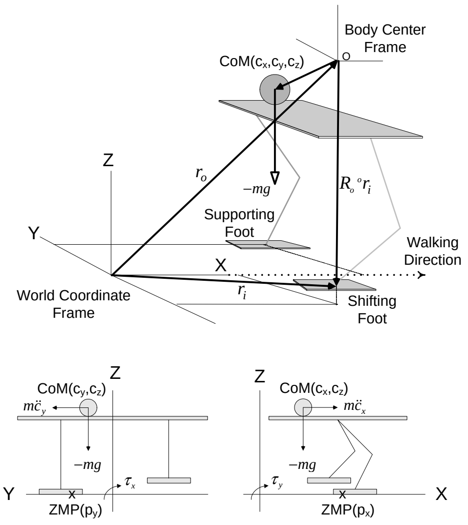


Figure 1: Rolling Sphere Model for Dynamic Walking.

equations will be used to prove the stability of the posture/walking controller in the following sections.

2 KINEMATIC RESOLUTION

Let a robot has n limbs and the first limb be the base limb. The base limb can be any limb but it should be on the ground to support the body. Each limb of a robot is hereafter considered as an independent limb. In general, the i-th limb has the following relation:

$$ {}^o\dot{x}_i = {}^oJ_i\dot{q}_i \tag{5}$$

for $i = 1, 2, \cdots, n$, where ${}^o\dot{x}_i \in \Re^6$ is the velocity of the end point of i-th limb, $\dot{q}_i \in \Re^{n_i}$ is the joint velocity of i-th limb, ${}^oJ_i \in \Re^{6\times n_i}$ is the usual Jacobian matrix of i-th limb, and n_i means the number of active links of i-th limb. The leading superscript o implies that the elements are represented on the body center coordinate system shown in Fig. 1, which is fixed on a humanoid robot.

In the humanoid robot, the body center is floating, and thus the end point motion of i-th limb about the world coordinate system is written as follows:

$$\dot{x}_i = X_i^{-1}\dot{x}_o + X_o{}^oJ_i\dot{q}_i \tag{6}$$

where $\dot{x}_o = [\dot{r}_o^T; \omega_o{}^T]^T \in \Re^6$ is the velocity of the body center represented on the world coordinate system, and

$$X_i = \begin{bmatrix} I_3 & [R_o{}^or_i\times] \\ 0_3 & I_3 \end{bmatrix} \in \Re^{6\times6} \tag{7}$$

is a (6×6) matrix which relates the body center velocity and the i-th limb velocity. I_3 and 0_3 are an (3×3) identity and zero matrix, respectively. $R_o{}^or_i$ is the position vector from the body center to the end point of the i-th limb represented on the world coordinate frame. $[(\cdot)\times]$ is a skew-symmetric matrix for the cross product. The transformation matrix X_o is

$$X_o = \begin{bmatrix} R_o & 0_3 \\ 0_3 & R_o \end{bmatrix} \in \Re^{6\times6} \tag{8}$$

where $R_o \in \Re^{3\times3}$ is the orientation of the body center represented on the world coordinate frame, and hereafter, we will use the relation $J_i \overset{\triangle}{=} X_o{}^oJ_i$.

All the limbs in a robot should have the same body center velocity, in other words, from Eq. (6), we can see that all the limbs should satisfy the compatibility condition that the body center velocity is the same, and thus, i-th limb and j-th limb should satisfy the following relation:

$$X_i(\dot{x}_i - J_i\dot{q}_i) = X_j(\dot{x}_j - J_j\dot{q}_j). \tag{9}$$

From Eq. (9), the joint velocity of any limb can be represented by the joint velocity of the base limb and cartesian motions of limbs. Actually, the base limb should be chosen to be the support leg in single support phase or one of both legs in double support phase. Let us express the base limb with the subscript 1, then the joint velocity of i-th limb is expressed as:

$$\dot{q}_i = J_i^+\dot{x}_i - J_i^+X_{i1}(\dot{x}_1 - J_1\dot{q}_1), \tag{10}$$

for $i = 2, \cdots, n$, where J_i^+ means the Moore-Penrose pseudoinverse of J_i and

$$X_{i1} \overset{\triangle}{=} X_i^{-1}X_1 = \begin{bmatrix} I_3 & [R_o({}^or_1 - {}^or_i)\times] \\ 0_3 & I_3 \end{bmatrix}. \tag{11}$$

The position of CoM represented on the world coordinate frame, in Fig. 1, is given by

$$c = r_o + \sum_{i=1}^{n} R_o{}^oc_i \tag{12}$$

where n is the number of limbs, c is the position vector of CoM represented on the world coordinate system, and ${}^oc_i$ means the CoM position vector of i-th limb represented on the body center coordinate frame which is composed of n_i active links. Now, let us differentiate Eq. (12), then the it is obtained as follows:

$$\dot{c} = \dot{r}_o + \omega_o \times (c - r_o) + \sum_{i=1}^{n} R_o{}^oJ_{ci}\dot{q}_i. \tag{13}$$

where ${}^oJ_{c_i} \in \Re^{3\times n_i}$ means CoM Jacobian matrix of i-th limb represented on the body center coordinate frame, and hereafter, we will use the relation $J_{c_i} \overset{\triangle}{=} R_o{}^oJ_{c_i}$.

Remark **1** *The CoM Jacobian matrix of i-th limb represented on the body center frame is expressed by*

$$ {}^{o}J_{c_i} \overset{\triangle}{=} \sum_{k=1}^{n_i} \mu_{i,k} \frac{\partial\, {}^{o}c_{i,k}}{\partial q_i}, \tag{14}$$

where ${}^{o}c_{i,k} \in \Re^3$ means the position vector of center of mass of k-th link in i-th limb represented on the body center frame and the mass influence coefficient of k-th link in i-th limb is defined as follow:

$$\mu_{i,k} \overset{\triangle}{=} \frac{\textit{mass of k-th link in i-th limb}}{\textit{total mass}}. \tag{15}$$

The motion of body center frame can be obtained by using Eq. (6) for the base limb as follows:

$$\begin{aligned} \dot{x}_o = {} & X_1\{\dot{x}_1 - J_1\dot{q}_1\} \\ \begin{bmatrix} \dot{r}_o \\ \omega_o \end{bmatrix} = {} & \begin{bmatrix} I_3 & [R_o {}^{o}r_1 \times] \\ 0_3 & I_3 \end{bmatrix} \left\{ \begin{bmatrix} \dot{r}_1 \\ \omega_1 \end{bmatrix} - \begin{bmatrix} J_{v_1} \\ J_{\omega_1} \end{bmatrix} \dot{q}_1 \right\}, \end{aligned} \tag{16}$$

where J_{v_1} and J_{ω_1} are the linear and angular velocity part of the base limb Jacobian J_1 expressed on the world coordinate frame, respectively. Now, if Eq. (10) is applied to Eq. (13) for all limbs except the base limb with subscript 1, the CoM motion is rearranged as follows:

$$\begin{aligned} \dot{c} = {} & \dot{r}_o + \omega_o \times (c - r_o) + J_{c_1}\dot{q}_1 \\ & + \sum_{i=2}^{n} J_{c_i} J_i^{+} (\dot{x}_i - X_{i1}\dot{x}_1) + \sum_{i=2}^{n} J_{c_i} J_i^{+} X_{i1} J_1 \dot{q}_1. \end{aligned} \tag{17}$$

Here, if Eq. (16) is applied to Eq. (17), then the CoM motion is only related with the motion of base limb. Also, if the base limb has the face contact with the ground (the end-point of base limb represented on world coordinate frame is fixed, $\dot{x}_1 = 0$, namely, $\dot{r}_1 = 0$, $\omega_1 = 0$), then Eq. (17) is simplified as follows:

$$\begin{aligned} \dot{c} - \sum_{i=2}^{n} J_{c_i} J_i^{+} \dot{x}_i = {} & -J_{v_1}\dot{q}_1 + r_{c1} \times J_{\omega_1}\dot{q}_1 + J_{c_1}\dot{q}_1 \\ & + \sum_{i=2}^{n} J_{c_i} J_i^{+} X_{i1} J_1 \dot{q}_1. \end{aligned} \tag{18}$$

where $r_{c1} = c - r_1$.

Finally, $3 \times n_1$ CoM Jacobian matrix with embedded motions can be rewritten like usual kinematic Jacobian of base limb:

$$\dot{c}_{\text{fsem}} = J_{\text{fsem}}\, \dot{q}_1, \tag{19}$$

where

$$\dot{c}_{\text{fsem}} \overset{\triangle}{=} \dot{c} - \sum_{i=2}^{n} J_{c_i} J_i^{+} \dot{x}_i, \tag{20}$$

$$J_{\text{fsem}} \overset{\triangle}{=} -J_{v_1} + r_{c1} \times J_{\omega_1} + J_{c_1} + \sum_{i=2}^{n} J_{c_i} J_i^{+} X_{i1} J_1. \tag{21}$$

Here, if the CoM Jacobian is augmented with the orientation Jacobian of body center ($\omega_o = -J_{\omega_1} q_1$) and all desired cartesian motions are embedded in Eq. (20), then the desired joint configurations of base limb (support limb) are resolved as follows:

$$\dot{q}_{1,d} = \begin{bmatrix} J_{\text{fsem}} \\ -J_{\omega_1} \end{bmatrix}^{+} \begin{bmatrix} \dot{c}_{\text{fsem},d} \\ \omega_{o,d} \end{bmatrix}, \tag{22}$$

where the subscript d means the desired motion and

$$\dot{c}_{\text{fsem},d} = \dot{c}_d - \textstyle\sum_{i=2}^{n} J_{c_i} J_i^{+} \dot{x}_{i,d}. \tag{23}$$

All the given desired limb motions, $\dot{x}_{i,d}$ are embedded in the relation of CoM Jacobian, thus the effect of the CoM movement generated by the given limb motion is compensated by the base limb. The CoM motion with fully specified embedded motions,

After solving Eq. (22), the desired joint motion of the base limb is obtained. The resulting base limb motion makes a humanoid robot balanced automatically during the movement of the all other limbs. With the desired joint motion of base limb, the desired joint motions of all other limbs can be obtained by Eq. (10) as follow:

$$\dot{q}_{i,d} = J_i^{+} (\dot{x}_{i,d} + X_{i1} J_1 \dot{q}_{1,d}), \quad \text{for } i = 2, \cdots, n. \tag{24}$$

The resulting motion follows the given desired motions, regardless of balancing motion by base limb. In other words, the suggested kinematic resolution method of CoM Jacobian with embedded motion offers the WBC(whole body coordination) function to the humanoid robot automatically.

3 STABILITY

The control system is said to be disturbance input-to-state stable (ISS), if there exists a smooth positive definite radially unbounded function $V(e,t)$, a class $\mathcal{K}_\infty$ function γ_1 and a class $\mathcal{K}$ function γ_2 such that the following dissipativity inequality is satisfied:

$$\dot{V} \leq -\gamma_1(|e|) + \gamma_2(|\varepsilon|), \tag{25}$$

where $\dot{V}$ represents the total derivative for Lyapunov function, e the error state vector and ε disturbance input vector.

In this section, we propose the posture/walking controller for bipedal robot systems as shown in Fig. 2. In this figure, first, the ZMP Planer and CoM Planer generate the desired trajectories satisfying the following differential equation:

$$p_{i,d} = c_{i,d} - 1/\omega_n^2 \ddot{c}_{i,d} \quad \text{for} \quad i = x, y. \tag{26}$$

Second, the simplified model for the real bipedal walking robot has the following dynamics:

$$\begin{aligned} \dot{c}_i &= u_i + \varepsilon_i \\ p_i &= c_i - 1/\omega_n^2 \ddot{c}_i \quad \text{for} \quad i = x, y, \end{aligned} \tag{27}$$

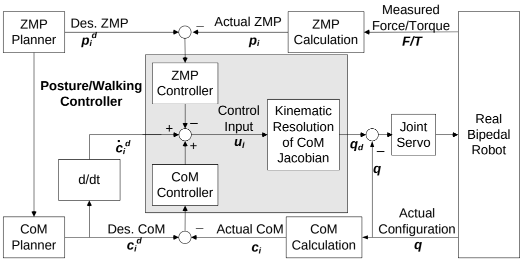


Figure 2: Posture/Walking Controller for Humanoid Robot.

where ε_i is the disturbance input produced by actual control error, u_i is the control input, c_i and p_i are the actual positions of CoM and ZMP measured from the real bipedal robot, respectively. Here, we assume that the the disturbance produced by control error is bounded and its differentiation is also bounded, namely, $|\varepsilon_i| < a$ and $|\dot{\varepsilon}_i| < b$ with any positive constants a and b. Also, we should notice that the control error always exists in real robot systems and its magnitude depends on the performance of embedded local joint servos. The following theorem proves the stability of the posture/walking controller to be suggested for the simplified robot model.

Theorem **1** *Let us define the ZMP and CoM error for the simplified bipedal robot control system (27) as follows:*

$$e_{p,i} \overset{\triangle}{=} p_{i,d} - p_i \tag{28}$$

$$e_{c,i} \overset{\triangle}{=} c_{i,d} - c_i \quad \text{for} \quad i = x, y. \tag{29}$$

If the posture/walking control input u_i in Fig. 2 has the following form:

$$u_i = \dot{c}_i^d - k_{p,i}e_{p,i} + k_{c,i}e_{c,i} \tag{30}$$

under the gain conditions:

$$k_{c,i} > \omega_n \quad \text{and} \quad 0 < k_{p,i} < \left(\frac{\omega_n^2 - \beta^2}{\omega_n} - \gamma^2\right) \tag{31}$$

satisfying the following conditions:

$$\beta < \omega_n \quad \text{and} \quad \gamma < \sqrt{\frac{\omega_n^2 - \beta^2}{\omega_n}}, \tag{32}$$

then the posture/walking controller gives the disturbance input($\varepsilon_i, \dot{\varepsilon}_i$)-to-state($e_{p,i}, e_{c,i}$) stability (ISS) to a simplified bipedal robot, where, the $k_{p,i}$ is the proportional gain of ZMP controller and $k_{c,i}$ is that of CoM controller in Fig. 2.

Proof. First, we get the error dynamics from Eq. (26) and (27) as follows:

$$\ddot{e}_{c,i} = \omega_n^2(e_{c,i} - e_{p,i}). \tag{33}$$

Second, another error dynamics is obtained by using Eq. (27) and (30) as follows:

$$k_{p,i}e_{p,i} = \dot{e}_{c,i} + k_{c,i}e_{c,i} + \varepsilon_i, \tag{34}$$

also, this equation can be rearranged for $\dot{e}_c$:

$$\dot{e}_{c,i} = k_{p,i}e_{p,i} - k_{c,i}e_{c,i} - \varepsilon_i. \tag{35}$$

Third, by differentiating the equation (34) and by using equations (33) and (35), we get the following:

$$\begin{aligned} \dot{e}_{p,i} &= 1/k_{p,i}\,(\ddot{e}_{c,i} + k_{c,i}\dot{e}_{c,i} + \dot{\varepsilon}_i) \\ &= \omega_n^2/k_{p,i}(e_{c,i} - e_{p,i}) \\ &\quad + k_{c,i}/k_{p,i}(k_{p,i}e_{p,i} - k_{c,i}e_{c,i} - \varepsilon_i) + (1/k_{p,i})\dot{\varepsilon}_i \\ &= \left(\frac{\omega_n^2 - k_{c,i}^2}{k_{p,i}}\right)e_{c,i} - \left(\frac{\omega_n^2 - k_{p,i}k_{c,i}}{k_{p,i}}\right)e_{p,i} \\ &\quad + \frac{1}{k_{p,i}}(\dot{\varepsilon}_i - k_{c,i}\varepsilon_i). \end{aligned} \tag{36}$$

Fourth, let us consider the following Lyapunov function:

$$V(e_{c,i}, e_{p,i}) \overset{\triangle}{=} \frac{1}{2}\left[(k_{c,i}^2 - \omega_n^2)e_{c,i}^2 + k_{p,i}^2e_{p,i}^2\right], \tag{37}$$

where $V(e_c, e_p)$ is the positive definite function for $k_{p,i} > 0$ and $k_{c,i} > \omega_n$, except $e_{c,i} = 0$ and $e_{p,i} = 0$. Now, let us differentiate the above Lyapunov function, then we can get the following:

$$\begin{aligned} \dot{V} \leq\ & -(k_{c,i} - \alpha^2)(k_{c,i}^2 - \omega_n^2)e_{c,i}^2 \\ & -k_{p,i}[\omega_n^2 - (k_{p,i} + \gamma^2)k_{c,i} - \beta^2]e_{p,i}^2 \\ & + \left[\frac{(k_{c,i}^2 - \omega_n^2)}{4\alpha^2} + \frac{k_{p,i}k_{c,i}}{4\gamma^2}\right]\varepsilon_i^2 + \frac{k_{p,i}}{4\beta^2}\dot{\varepsilon}_i^2 \end{aligned} \tag{38}$$

where $e_{c,i}^2$ term is negative definite with any positive constant satisfying $\alpha < \sqrt{\omega_n}$ and $e_{p,i}^2$ term is negative definite under the given conditions (31). Here, since the inequality (38) follows the ISS property (25), we concludes that the proposed posture/walking controller gives the disturbance input($\varepsilon_i, \dot{\varepsilon}_i$)-to-state($e_{p,i}, e_{c,i}$) stability (ISS) to the simplified control system model of bipedal robot. □

To make active use of the suggested control scheme, the control input u of Eq. (30) suggested in Theorem 1 is applied to the place of the term $\dot{c}_d$ in Eq. (23). In other words, equation (23)is modified to include the ZMP and CoM controllers as following forms:

$$\dot{c}_{\text{fsem},d} = u - \sum_{i=2}^{n} J_{c_i}J_i^+\dot{x}_{i,d} \tag{39}$$

where $u \overset{\triangle}{=} \dot{c}_d - k_pe_p + k_ce_c$. And then, the suggested kinematic resolution method of Eq. (22) and (24) are utilized to obtain the desired base limb and other limb motions in the joint space as shown in Fig. 2.

ACKNOWLEDGEMENTS

This work was supported in part by the Korea Research Foundation Grant funded by the Korea Government (MOEHRD, Basic Research Promotion Fund) (KRF-2006-003-D00186) and in part by MIC & IITA through IT leading R&D support project, Republic of Korea.

REFERENCES

Choi, Y., Kim, D., and You, B. J. (2006). On the walking control for humanoid robot based on the kinematic resolution of com jacobian with embedded motion. *Proc. of IEEE Int. Conf. on Robotics and Automation*, pages 2655–2660.

Kajita, S., ad M. Saigo, K. Y., and Tanie, K. (2001). Balancing a humanoid robot using backdrive concerned torque control and direct angular momentum feedback. *Proc. of IEEE Int. Conf. on Robotics and Automation*, pages 3376–3382.

Sugihara, T. and Nakamura, Y. (2002). Whole-body cooperative balancing of humanoid robot uisng COG jacobian. *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 2575–2580.

ON COMPUTING MULTI-FINGER FORCE-CLOSURE GRASPS OF 2D OBJECTS

Belkacem Bounab [*] [**], Daniel Sidobre [**] [‡]
[*] *Mechanical Laboratory of Structures, EMP, Algiers, Algeria*
[**] *LAAS-CNRS, Toulouse, France*
[‡] *Paul Sabatier University, Toulouse, France*
bbounab@laas.fr, daniel.sidobre@laas.fr

Abdelouhab Zaatri
Advanced Technologies Laboratory, Mentouri University, Constantine, Algeria
zaatri@hotmail.com

Keywords: Multi-Fingered grasps, Force-Closure, Central Axis, Grasp Wrenches.

Abstract: In this paper, we develop a new algorithm for computing force-closure grasps of two-dimensional (2D) objects using multifingred hand. Assuming hard-finger point contact with Coulomb friction, we present a new condition for multi-finger to form force-closure grasps. Based on the central axis of contact wrenches, an easily computable algorithm for force-closure grasps has been implemented and its efficiency has been demonstrated by examples.

1 INTRODUCTION

Grasping remains one of the fundamental problems in robotics. Research has been directed towards the design and control of multifingred dexterous robot hand to increase robot dexterity and adaptability (Li J-W., Jin M-H. and Liu H. 2003).

A main property of a multi-finger stable grasp is force-closure. It's the ability to balance any external object wrenches by applying appropriate finger wrenches at the contact points. In other words, a grasp on an object is force-closure if and only if arbitrary force and torque can be exerted on the object through the fingers (Yan-Bin Jia 2004). It's complicated to assure that the applied finger forces remain in the friction cone at all times so as to avoid fingers slippage on the object surface (Murray R., Li Z. and Sastry S. 1994).

Human can use more than three/four fingers of his hand to manipulate objects. During such tasks, there exists a lot of contact points between the hand and grasped object. The question is: how can we evaluate or compute force-closure of such grasps?

In this paper, we are focused on the problem of computing force-closure of multifingered grasps of 2D objects. We develop a new approach for force-closure test independently of fingers's number. This quality is obtained using the mechanical properties of the grasp wrench.

2 RELATED WORK

Force-closure test is an essential problem in grasping. However, The notion of force-closure does not directly yield a method for force-closure test (Sudsang A. and Phoka T. 2005). Some necessary and sufficient conditions for force-closure were formulated in order to derive force-closure tests. A commonly used necessary and suffecient force-closure condition given by (Salisbury J.K. and Roth B. 1982) allowed a force-closure test to be performed by checking whether the origin is strictly inside the convex hull of the primitive contact wrenches. This test also provided an underlying idea to recent work in grasping (D. Ding, Y-H Liu, and S. Wang 2001). Nguyen (Nguyen, V.D. 1988) formally demonstrated for 2-fingered grasps that non-marginal equilibrium grasps achieve force-closure. Recently, (Li J-W., Jin M-H. and Liu H. 2003) proposed a necessary and suffecient condition for 3-fingered force-closure grasps based on (Ponce J. and Faverjon B. 1995) and developed an algorithm for three-finger force-closure test. Their method begins by the processing of friction cones using an operation called disposition H, then, they attack the problem of determining the intersection of the three double-side friction cones.

The rest of the paper is organized as follow, in section 3, we present the background of grasp wrenches central axes and the relationship between these axes and grasp force-closure. In section 4, we propose a new multi-finger force-closure condition. Hence, a novel algorithm is presented, which its implementation needs little geometric computations. In section 5, we present some multi-finger grasps examples. Finally, we conclude with future works.

3 CENTRAL AXIS OF GRASP WRENCHES

Based on Coulomb friction model, a contact force is constrained to lie in a friction cone centered about the internal surface normal at contact point.

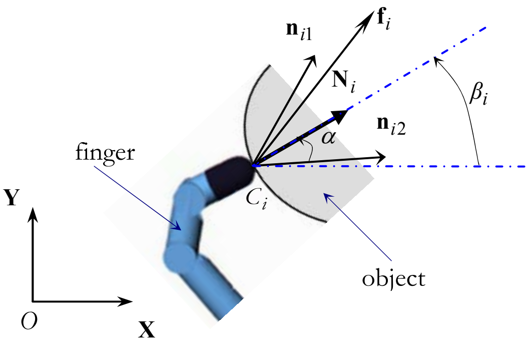


Figure 1: Contact between the finger and an object showing friction cone.

As shown in figure 1, a friction cone at C_i is bounded by vectors $\mathbf{n}_{i1}$ and $\mathbf{n}_{i2}$, and any force $\mathbf{f}_i$ is a nonnegative combination of these two vectors. In 2D case, contact forces are

$$\mathbf{f}_i = a_{i1}\,\mathbf{n}_{i1} + a_{i2}\,\mathbf{n}_{i2} \quad i = 1,\ldots,m \tag{1}$$

With $a_{i1} \geq 0, a_{i2} \geq 0$ to avoid fingertips slippage. m is the number of contact points.

If $\mathbf{N}_i$ is the surface normal at the contact point C_i and α is the friction angle that depends on materials in contact (finger and object) then,

$$\beta_i = Atan\,2(\mathbf{N}_i\,\mathbf{Y}, N_i\,\mathbf{X}) \tag{2}$$

and

$$\mathbf{n}_{i1} = \begin{pmatrix} Cos(\beta_i + \alpha) \\ Sin(\beta_i + \alpha) \end{pmatrix} ;\ \mathbf{n}_{i2} = \begin{pmatrix} Cos(\beta_i - \alpha) \\ Sin(\beta_i - \alpha) \end{pmatrix} \tag{3}$$

The contact wrench produced by $\mathbf{f}_i$ reduced at the point O is defined by

$$\begin{aligned} \mathbf{w}_i &= \begin{bmatrix} \mathbf{f}_i \\ \mathbf{OC}_i \times \mathbf{f}_i \end{bmatrix} \\ &= \begin{bmatrix} a_{i1}\,\mathbf{n}_{i1} + a_{i2}\,\mathbf{n}_{i2} \\ \mathbf{OC}_i \times \left(a_{i1}\,\mathbf{n}_{i1} + a_{i2}\,\mathbf{n}_{i2}\right) \end{bmatrix} \end{aligned} \tag{4}$$

The external wrench applied by the robotic hand on the grasped object is given by

$$\mathbf{W}_{c/o} = \sum_{i=1}^{m} \mathbf{w}_i = \begin{bmatrix} \mathbf{F}_c \\ \boldsymbol{\tau}_{c/o} \end{bmatrix} \tag{5}$$

With

$$\begin{aligned} \mathbf{F}_c &= \sum_{i=1}^{m} \mathbf{f}_i = \sum_{i=1}^{m} \left(a_{i1}\,\mathbf{n}_{i1} + a_{i2}\,\mathbf{n}_{i2}\right) \\ \boldsymbol{\tau}_{c/o} &= \sum_{i=1}^{m} \left(\mathbf{OC}_i \times \left(a_{i1}\,\mathbf{n}_{i1} + a_{i2}\,\mathbf{n}_{i2}\right)\right) \end{aligned} \tag{6}$$

In two-dimensional grasps case, we have

$$\mathbf{F}_c = \left(F_{cx}, F_{cy}, 0\right)^T \text{ and } \boldsymbol{\tau}_{c/o} = \left(0, 0, \tau_{z/o}\right)^T \tag{7}$$

With

$$\begin{cases} F_{cx} = \mathbf{X} \cdot \displaystyle\sum_{i=1}^{m} \left(a_{i1}\,\mathbf{n}_{i1} + a_{i2}\,\mathbf{n}_{i2}\right) \\ F_{cy} = \mathbf{Y} \cdot \displaystyle\sum_{i=1}^{m} \left(a_{i1}\,\mathbf{n}_{i1} + a_{i2}\,\mathbf{n}_{i2}\right) \\ \tau_{z/o} = \mathbf{Z} \cdot \displaystyle\sum_{i=1}^{m} \left(\mathbf{OC}_i \wedge \left(a_{i1}\,\mathbf{n}_{i1} + a_{i2}\,\mathbf{n}_{i2}\right)\right) \end{cases} \tag{8}$$

Poinsot's theorem: "*Every collection of wrenches applied to a rigid body is equivalent to a force applied along a fixed axis (central axis) and a torque around the same axis*" (Murray R., Li Z., Sastry S. 1994). Using this theorem, points of the central axis Δ_C of contact wrench are given by

$$\Delta_C = \begin{cases} \dfrac{\mathbf{F}_c \times \boldsymbol{\tau}_{c/o}}{\mathbf{F}_c^{\,2}} + \lambda\,\mathbf{F}_c & if\ \mathbf{F}_c \neq \mathbf{0} \\ \lambda\,\boldsymbol{\tau}_{c/o} & if\ \mathbf{F}_c = \mathbf{0} \end{cases} \ : \left(\lambda \in \Re\right) \tag{9}$$

The axis Δ_C is a directed line through a point. For $\mathbf{F}_c \neq \mathbf{0}$, the central axis is a line in the

$\mathbf{F}_c$ direction going through the point Q_0 such as $\mathbf{OQ}_0 = (\mathbf{F}_c \times \boldsymbol{\tau}_{c/o}) / \mathbf{F}_c{}^2$.

For $\mathbf{F}_c = \mathbf{0}$, the axis is a line in the $\boldsymbol{\tau}_{c/o}$ direction going through the origin (Murray R., Li Z., Sastry S. 1994).

In two-dimentional case with non null forces ($\mathbf{F}_c \neq \mathbf{0}$), the torque around the central axis is zero. The force $\mathbf{F}_c$ is an invariant vector and always parallel to Δ_C. Figure 2 shows the central axis in 2D grasps when $\mathbf{F}_c \neq \mathbf{0}$. it is Characterized by the following equation

$$y = \left(\frac{F_{cy}}{F_{cx}}\right) \cdot x + \left(\frac{\tau_{z/o}}{F_{cx}}\right) \qquad (10)$$

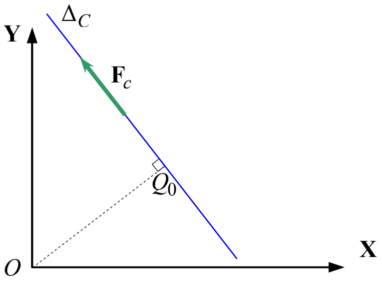
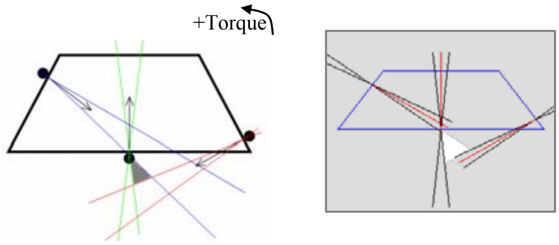


Figure 2: Central axis parameters.

In figure 3-a, we present a first example of three-finger 2D grasp. By varying forces $\mathbf{f}_i$ randomly (in orientation and amplitude) inside friction cones (the friction angle), figure 3-b illustrates all possible central axes of grasp wrenches.

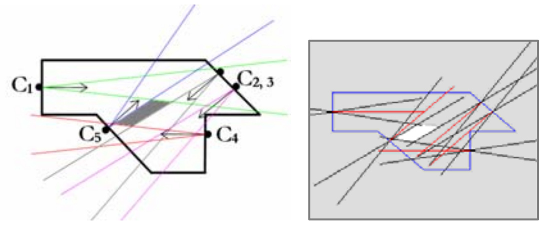
There is no central axis passing through the gray region. In this region, positive torque can't be exerted on the object through the finger contacts. This grasp can't achieve force-closure. Exactly, the grasp can not achieve torque-closure because the object turn around the gray region in figure 3-a.



(a) (b)

Figure 3: a) no force-closure 2D grasp, b) all central axes of grasp wrenches ($\alpha = 5°$).

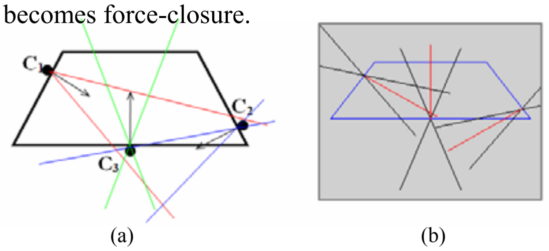
A second example is shown in figure 4; we present a non-force-closure grasp using five contact points. This grasp is instable and the object turn around $\mathbf{Z}$ axis in the gray region (figure 4-a).



(a) (b)

Figure 4: a) non-force-closure five-fingers grasp; b) central axes of grasp wrenches ($\alpha = 10°$).

When a grasp is force closure, the central axes of grasp wrenches can wholly sweep the plan $(\mathbf{X}, \mathbf{Y})$. In the third example, shown in figure 5, we use the same finger's configuration as figure 3 but we change the friction angle $\alpha = 20°$. The grasp becomes force-closure.



(a) (b)

Figure 5: a) three fingers force-closure 2D grasp, b) central axes of grasp wrenches ($\alpha = 20°$).

According to these three examples, we can conclude that the distribution of central axes can confirm if a grasp is force-closure or not (for any contact points number).

4 FORCE-CLOSURE AND EQUILIBRIUM CONDITION

In 2D grasps and based on Poinsot's theorem, we can give the following definition.

Definition 1: Any external wrench applied by the robotic hand on a 2D object is equivalent to a force along a central axis of this wrench. When the force is equal to zero, the external wrench is equivalent to a torque about the grasp plan normal.

Definition 2: A grasp on an object is force-closure if and only if any arbitrary force and torque can be exerted on the object through the finger contacts (Yan-Bin Jia 2004). There are other force-closure definitions, but this one is more useful for our deduction.

Definition 3: A grasp is said to achieve equilibrium when it exists forces (not all of them being zero) in the friction cones at the fingertips such that the sum of the corresonding wrenches is zero (Sudsang A., Phoka T. 2005).

4.1 Equilibrium Condition

During objects grasp operations there exist two kinds of external wrenches applied on the manipulated object, task wrench (applied by the environment) and contact wrench (applied by the robotic hand fingers). Based on *definitions 1* and *definitions 3*, we derive a new proposed necessary and sufficient equilibrium condition.

Proposition 1: A multifingers grasp is said to achieve equilibrium if and only if the central lines of contact wrench and task wrench have the same support and opposite direction.

Proof:

i) *Sufficient Condition*:

the external contact wrench given by equation (5) and task wrench is given by

$$\mathbf{W}_{t/o} = \begin{bmatrix} \mathbf{F}_t \\ \boldsymbol{\tau}_{t/o} \end{bmatrix} \qquad (11)$$

The object is in equilibrium if:

$$\mathbf{W}_{c/o} + \mathbf{W}_{t/o} = \{\mathbf{0}\} \Rightarrow \begin{cases} \mathbf{F}_c = -\,\mathbf{F}_t \\ \boldsymbol{\tau}_{c/o} = -\,\boldsymbol{\tau}_{t/o} \end{cases} \qquad (12)$$

From Relation (9), the central axis of contact wrench is defined by

$$\mathbf{OP}_c = \frac{\mathbf{F}_c \times \boldsymbol{\tau}_{c/o}}{\mathbf{F}_c^{\,2}} + \lambda_c \,\mathbf{F}_c \quad (\lambda_c \in \Re) \qquad (13)$$

Substuting (12) in (13) lead to

$$\mathbf{OP}_c = \frac{\mathbf{F}_t \times \boldsymbol{\tau}_{t/o}}{\mathbf{F}_t^{\,2}} + \lambda_t \,\mathbf{F}_t \quad (\lambda_t = -\lambda_c \in \Re) \qquad (14)$$

Relation (14) defines the central axis of task wrench given by

$$\mathbf{OP}_t = \frac{\mathbf{F}_t \times \boldsymbol{\tau}_{t/o}}{\mathbf{F}_t^{\,2}} + \lambda_t \,\mathbf{F}_t \quad (\lambda_t \in \Re) \qquad (15)$$

In the case $\mathbf{F}_t = \mathbf{0}$, the points P_c are given by

$$\mathbf{OP}_c = \lambda_c \,\boldsymbol{\tau}_{c/o} = \lambda_t \,\boldsymbol{\tau}_{t/o} \;:\; (\lambda_t = -\lambda_c \in \Re) \qquad (16)$$

In both cases, Relations (14) and (16), the two wrenches (contact and task) should have the same central line with opposite directions. ■

ii) *Necessary Condition:*

Now, if we consider two wrenches reduced at the same point O and they have the same central line with opposite directions. We have two cases:

a) if $\mathbf{F}_t = \mathbf{0}$ then the central axis of the task wrench is defined by the unit vector $\mathbf{U}_t$ where:

$$\mathbf{U}_t = \frac{\boldsymbol{\tau}_{t/o}}{\left\|\boldsymbol{\tau}_{t/o}\right\|}$$

If the two wrenches have the same central line with opposite direction then the contact central axis is defined by the following unit vector:

$$\mathbf{U}_c = -\mathbf{U}_t = \frac{\boldsymbol{\tau}_{c/o}}{\left\|\boldsymbol{\tau}_{c/o}\right\|}$$

We conclude that:

$$Sgn(\boldsymbol{\tau}_{c/o} \cdot \mathbf{U}_c)\, Sgn(\boldsymbol{\tau}_{t/o} \cdot \mathbf{U}_c) < 0 \qquad (17)$$

Sgn is the sign function that computes the sign of the leading coefficient of expression.

b) if $\mathbf{F}_t \neq \mathbf{0}$, having the same central axis with opposite direction implies

$$Sgn(\mathbf{F}_c \cdot \mathbf{U}_c)\, Sgn(\mathbf{F}_t \cdot \mathbf{U}_c) < 0 \qquad (18)$$

Where $\mathbf{U}_c$ and $\mathbf{U}_t$ define the unit vectors of the two central axes. We have:

$$\mathbf{U}_c = \frac{\mathbf{F}_c}{\left\|\mathbf{F}_c\right\|} \;\; ; \;\; \mathbf{U}_t = \frac{\mathbf{F}_t}{\left\|\mathbf{F}_t\right\|}$$

Using hypothesis that there is one central line and form relation (9), we have

$$\frac{\mathbf{F}_c \times \boldsymbol{\tau}_{c/o}}{\mathbf{F}_c^{\,2}} = \frac{\mathbf{F}_t \times \boldsymbol{\tau}_{t/o}}{\mathbf{F}_t^{\,2}} \qquad (19)$$

Then, replacing $\mathbf{F}_c = \left\|\mathbf{F}_c\right\| \mathbf{U}_c$; $\mathbf{F}_t = -\left\|\mathbf{F}_t\right\| \mathbf{U}_c$ in Relation (19) we obtain

$$\left(\frac{\boldsymbol{\tau}_{c/o}}{\left\|\mathbf{F}_c\right\|} + \frac{\boldsymbol{\tau}_{t/o}}{\left\|\mathbf{F}_t\right\|} \right) = \lambda \cdot \mathbf{U}_c \;:\; (\lambda \in \Re) \qquad (20)$$

In 2D case, the equation (20) can be only satisfied when $\lambda = 0$. therefore, the two torques have opposite signes:

$$Sgn(\boldsymbol{\tau}_{c/o} \cdot \mathbf{Z})\, Sgn(\boldsymbol{\tau}_{t/o} \cdot \mathbf{Z}) < 0 \qquad (21)$$

Relations (17, 18 and 21) imply that the contact wrench can generate grasp wrenches that opposite the external task wrench. Robotic hand can control its fingers force to produce the appropriate force/torque magnitude that achieving equilibrium.■

4.2 Force-Closure Condition

In particular, force-closure implies equilibrium, but there are wrench systems that achieve equilibrium but not force closure (Li Jia-Wei. and Cia He-Gao, 2003).

Using force-closure condition in *definition 2*, we can derive this definition

Definition 4: A grasp is force closed, if and only if it is in equilibrium for any arbitrary wrench (Bicchi A., Kumar V. 2000, Nguyen, V.D. 1988). Thus, force closure implies, fingers contact wrenches can balance any external task wrenches.

According to *proposition 1* and *definition 4*, we propose a new force-closure necessary and sufficient condition.

Proposition 2: A multifingred grasp of 2D objects is said to achieve force-closure if and only if the central axis of the fingers contact wrenches can sweep the grasp plan at any direction.

4.3 Force-Closure Test Algorithm

According to the *proposition 2*, we present a new algorithm for computing 2D multi-fingers grasps of arbitrary object.

Based on the central axis equation defined in relation (10), this central line can sweep the plan in all directions if

$$\forall (k_1, k_2) \in \Re^2, \exists\, \Delta_c \text{ Satisfy } y = k_1 \cdot x + k_2$$

Where

$$k_1 = \left(\frac{F_{cy}}{F_{cx}}\right) \quad ; \quad k_1 = \left(\frac{\tau_{z/o}}{F_{cx}}\right)$$

In other word, for any axis on the $(\mathbf{X}, \mathbf{Y})$ plan or along the vertical $\mathbf{Z}$, this axis must be one of the grasp wrench central axes.

This condition implies that the quantities k_1 and k_2 must take all real number, therefore

$$\begin{cases} F_{cx} \in [-\infty, +\infty] \\ F_{cy} \in [-\infty, +\infty] \\ \tau_{c/o} \in [-\infty, +\infty] \quad \forall O \end{cases} \tag{21}$$

The third sub-condition is function of the reduced point of the torque, to cover the entire grasp plan; we test this condition at all the vertices of the intersection of the m double-side friction cones (named B_k). In general case of m contact points, the number of intersection points is given by

$$N_{B_k} = 4 \cdot \sum_{k=1}^{m-1} (m-k) \tag{22}$$

Hence, a multifingred 2D grasp is said to achieve force-closure if each of these inequalities are true.

$$\left| \mathbf{X} \cdot \sum_{i=1}^{m} (\mathbf{n}_{i1} + \mathbf{n}_{i2}) \right| < \sum_{i=1}^{m} \left(|\mathbf{X} \cdot \mathbf{n}_{i1}| + |\mathbf{X} \cdot \mathbf{n}_{i2}| \right) \tag{23-1}$$

$$\left| \mathbf{Y} \cdot \sum_{i=1}^{m} (\mathbf{n}_{i1} + \mathbf{n}_{i2}) \right| < \sum_{i=1}^{m} \left(|\mathbf{Y} \cdot \mathbf{n}_{i1}| + |\mathbf{Y} \cdot \mathbf{n}_{i2}| \right) \tag{23-2}$$

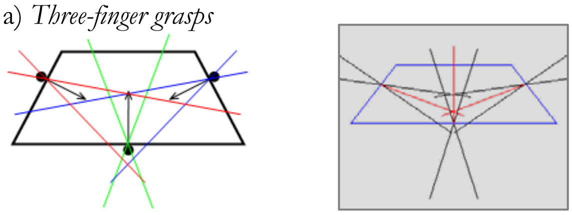
$$\left| \sum_{i=1}^{m} \mathbf{Z} \cdot \left(\mathbf{B}_k \mathbf{C}_i \wedge (\mathbf{n}_{i1} + \mathbf{n}_{i2}) \right) \right| < \sum_{i=1}^{m} \left(|\mathbf{Z} \cdot (\mathbf{B}_k \mathbf{C}_i \wedge \mathbf{n}_{i1})| + |\mathbf{Z} \cdot (\mathbf{B}_k \mathbf{C}_i \wedge \mathbf{n}_{i2})| \right) \tag{23-3}$$

From mechanical viewpoint, inequality (23-1) implies that fingers can generate forces along $\mathbf{X}$ and $-\mathbf{X}$, (23-2) means that fingers can exert force on the object along $\mathbf{Y}$ and $-\mathbf{Y}$. If the last inequality (23-3) is true $\left(\text{for } k = 1 \ldots N_{B_k}\right)$ then the finger can exert torque on object about the vertical axis $\mathbf{Z}$ in both directions.

5 EXAMPLES

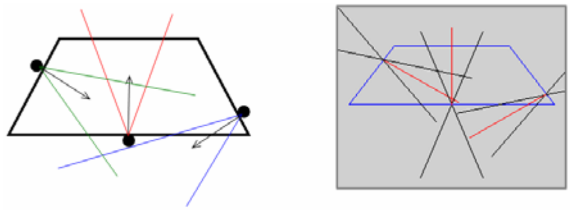
We present bellow some grasp examples using three, four and five fingers. In both cases (force-closure and no force-closure), we show the distribution of grasp wrench central axes.

a) *Three-finger grasps*



(a) (b)

Figure 6: a) a three-finger force-closure 2D grasp, b) central axes of grasp wrenches ($\alpha = 15°$).



(a) (b)

Figure 7: a) a three-finger force-closure 2D grasp, b) central axes of grasp wrenches ($\alpha = 20°$).

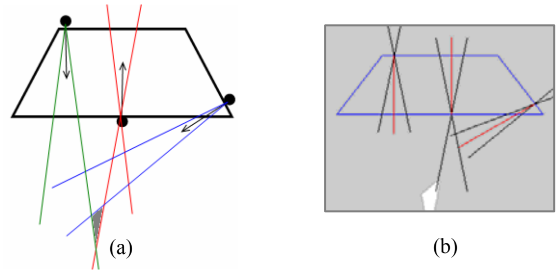


Figure 8: a) no force-closure 2D grasp, b) central axes of grasp wrenches ($\alpha = 10°$). Grasp wrenches can't generate a negative torque in grey zone.

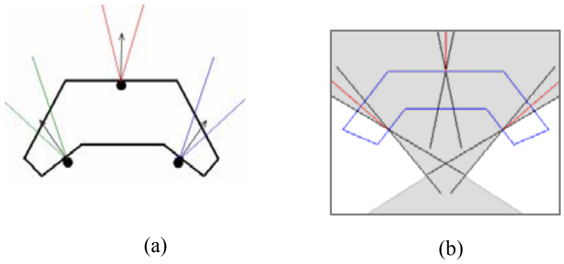


Figure 9: a) no force-closure 2D grasp, b) central axes of grasp wrenches ($\alpha = 10°$).Grasp wrenches can't exert a force along $(-\mathbf{Y})$ axis and can't generate torques in two-direction in unreachable zones in (b).

b) *Four-finger grasps*

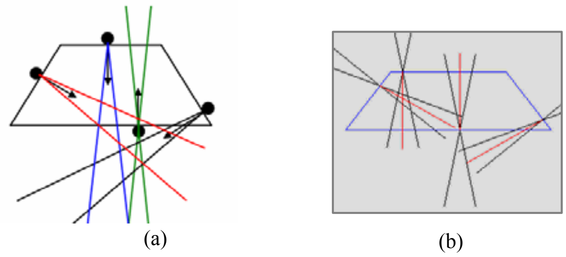


Figure 10: a) four-finger force-closure 2D grasp, b) central axes of grasp wrenches ($\alpha = 10°$).

c) *Five-finger grasps*

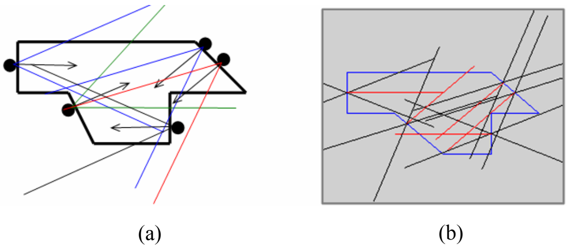


Figure 11: a) five-finger force-closure 2D grasp, b) central axes of grasp wrenches ($\alpha = 25°$).

6 CONCLUSION AND FUTURE WORK

We have presented a new equilibrium and force-closure conditions for multifingred 2D grasps. A novel algorithm for computing 2D multi-finger force-closure grasps of arbitrary objects was developed, which is very simple and needs little geometric computations. Therefore, it can be implemented in real-time multifingred grasp programming. Our future work will be concentrated on the extending of this algorithm to the 3D grasps and the quality measurement of grasps.

REFERENCES

Bicchi A., Kumar V. (2000). Robotic grasping and Contact: A review. In *Proc. IEEE ICRA*, pp. 348–352.

Ding D., Liu Y-H, and S. Wang (2001). Computation of 3-d form-closure grasps. In *IEEE Transactions on Robotics and Automation*, 515-522.

Li J-W., Cia He-Gao (2003). On Computing Three-Finger Force-Closure Grasps of 2-D and 3-D Objects. *Transaction on Robotics & Automation.* IEEE Proceeding, 155-161.

Li J-W., Jin M-H., Liu H. (2003). A New Algorithm for Three-finger Force-Closure Grasp of Polygonal Objects. In *International Conference on Robotics & Automation.* IEEE Proceeding, 1800-1804.

Murray R., Li Z., Sastry S. (1994). *A Mathematical Introduction to Robotic Manipulation,* Boca Raton, FL: CRC press.

Nguyen, V.D. (1988). Constructing force-closure grasps. In *Journal of Robotics Research,* vol. 7, no. 3, pp. 3-16.

Ponce J. and Faverjon B.. (1995). On computing three-finger force-closure grasps of polygonal objects. In *IEEE Transactions on Robotics and Automation*, 868-881.

Salisbury J.K. and Roth B. (1982) Kinematic and force analysis of articulated hands. In *Journal of Mechanisms, Transmissions, and Automation in Design,* 105, 33-41.

Sudsang A., Phoka T. (2005). Geometric Reformulation of 3-Fingred Force-Closure Condition. In *International Conference on Robotics & Automation.* IEEE Proceeding, 2338-2343.

Yan-Bin Jia (2004). Computation on Parametric Curves with an Application in Grasping. In *the International Journal of Robotics Research,* vol. 23, No. 7-8, 825-855.

PREDICTIVE CONTROL BY LOCAL VISUAL DATA
Mobile Robot Model Predictive Control Strategies Using Local Visual Information and Odometer Data

Lluis Pacheco and Ningsu Luo
Institute of Informatics and Applications, University of Girona, Av. Ll. Santaló s/n, Girona, Spain
lluispa@eia.udg.es, ningsu@eia.udg.es

Keywords: Autonomous mobile robots, computer vision control, system identification, model based control, predictive control, trajectory planning, obstacle avoidance, robot vision.

Abstract: Nowadays, the local visual perception research, applied to autonomous mobile robots, has succeeded in some important objectives, such as feasible obstacle detection and structure knowledge. This work relates the on-robot visual perception and odometer system information with the nonlinear mobile robot control system, consisting in a differential driven robot with a free rotating wheel. The description of the proposed algorithms can be considered as an interesting aspect of this report. It is developed an easily portable methodology to plan the goal achievement by using the visual data as an available source of positions. Moreover, the dynamic interactions of the robotic system arise from the knowledge of a set of experimental robot models that allow the development of model predictive control strategies based on the mobile robot platform PRIM available in the Laboratory of Robotics and Computer Vision. The meaningful contribution is the use of the local visual information as an occupancy grid where a local trajectory approaches the robot to the final desired configuration, while avoiding obstacle collisions. Hence, the research is focused on the experimental aspects. Finally, conclusions on the overall work are drawn.

1 INTRODUCTION

The research presented in this paper addresses to a kind of differential driven WMRs (wheeled mobile robots). Nowadays, the computer vision techniques applied to WMR have solved the problem of obstacle detection by using different methods as stereo vision systems, optical flow or DFF (depth from focus). Stereo vision systems seem to provide the easiest cues to infer scene depth (Horn, 1998). The optical flow techniques used in WMR result in several applications as i.e. structure knowledge, obstacle avoidance, or visual servoing (Campbell, et al., 2004). The DFF methods are also suitable for WMR. For example, three different focused images were used, with almost the same scene, acquired with three different cameras (Nourbakhsh, et al., 1997). In this work, it is supposed that available obstacle positions are provided by using computer vision systems. In this context, the allowed navigation control signals should achieve the obstacle avoidance as well as the final desired coordinates. Scientific community has developed several studies in this field. Based on the dynamic window approach with available robot speeds, the reactive avoidance collisions, safety stop and goal can be achieved using the dynamic constraints of WMR (Fox, et al., 1997). Rimon and Koditschek (1992) presented the methodologies for the exact motion planning and control, based on the artificial potential fields where the complete information about the free space and goal are encoded. Some approaches on mobile robots propose the use of potential fields, which satisfy the stability in a Lyapunov sense, in a short prediction horizon (Ögren and Leonard, 2005). The main contribution of this paper is the use of the visual information as a dynamic window where the collision avoidance and safety stop can be planned. Thus, local visual data, instead of artificial potential fields, are used in order to achieve the Lyapunov stability. The use of MPC (model predictive control) with available on-robot information is possible. Moreover, the local visual information is used as an occupancy grid that allows planning feasible trajectories towards the desired objective. The knowledge of the objective allows the optimal solution of the local desired coordinates based on the acquired images. The sensor fusion is done using visual perception, as the meaningful source of information in order to accomplish with the robot tasks. Other data provided by the encoder-based odometer system are also considered.

This paper is organized as follows: Section 1 gives a brief presentation about the aim of the present work. In the Section 2, the platform is introduced as an electromechanical system. This section also describes the experiments to be realized in order to find the parametric model of the robot suitable for designing and implementing MPC methods. In the Section 3, the use of visual data is presented as a horizon where optimal trajectories can be planned. Section 4 presents the MPC strategies used for achieving the path following of the reference trajectories. In the Section 5, some conclusions are made with special attention paid into the future research works.

2 ROBOT AND BASIC CONTROL METHODS

This section gives some description on the main robot electromechanical and sensorial systems of the platform tested in this work. Hence, the WMR PRIM, available in our lab, has been used in order to test and orient the research. The experimental modelling methodologies as well as the model predictive control are also introduced.

2.1 Electromechanical and Sensorial System of the Robot

Figure 1 shows the robot PRIM used in the research work. The mechanical structure of the robot is made of aluminum, with two independent wheels of 16cm diameters actuated by two DC motors. The distance between two wheels is 56.4cm. A third spherical omni-directional wheel is used to guarantee the system stability. The maximum continuous torque of each motor is 131mNm. The proportion of gear reduction for each motor is 86:1 and thus the total force actuating on the robot is near 141N. Shaft encoders with 500 counts/rev are placed at the motor axes, which provide 43000 counts for each turn of the wheel. A set of PLD (programmable logic device) boards is connected to the digital outputs of the shaft encoders. The printed circuits boards (PCB) are used to measure the speed of each motor at every 25ms.

An absolute counter provides the counts in order to measure the robot position by the odometer system. Another objective of these boards is to generate a signal of 23khz PWM for each motor.

The communication between the central digital can computer and the boards is made through the thus it parallel port. The speed is commanded by a byte and generate from 0 to 127 advancing or reversing speed commands. The maximal speed is near 0.5m/s. A set of microcontroller boards (MCS-51) is used to read the information available from different connected sensors. The rate of communication with these boards is 9600 b/s. Figure 2 shows the electronic and sensorial system blocks. The data gathering and the control by digital computer is set to 100ms.

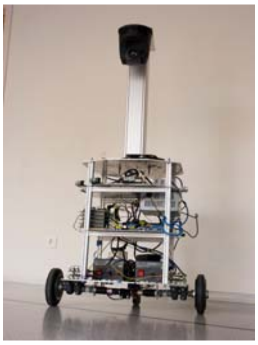


Figure 1: The robot PRIM used in this work.

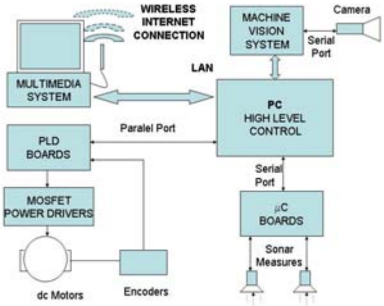


Figure 2: The sensorial and electronic system blocs.

The system flexibility is increased with the possibility of connecting with other computer systems through a local LAN. In this research, it is connected to a machine vision system that controls a colour camera EVI-D70P-PAL through the VISCA RS232-C control protocol. For instance, the camera configuration used in this work is of a horizontal field of view of 48º, and a vertical field of 37º. The focus, pan and tilt remain fixed under present configuration. Hence, the camera pose is set to 109cm from the floor with a tilt angle of 32º. The local desired coordinates, obtained by the visual perception information, are transmitted to the control unit connecting the USB port to the LAN.

2.2 Experimental Model

The parametric identification process is based on black box models (Lju, 1989), (Norton, 1986) and (Van Overschee, Moor, 1996). Thus, the transfer functions are related to a set of polynomials that allow the use of analytic methods in order to deal with the problem of controller design. The nonholonomic system dealt with in this work is considered initially as a MIMO (multiple input multiple output) system, which is composed of a set of SISO subsystems with coupled dynamic influence between two DC motors. The approach of multiple transfer functions consists in making the experiments with different speeds. In order to find a reduced-order model, several studies and experiments have been done through the system identification and model simplification.

2.2.1 System Identification

The parameter estimation is done by using a PRBS (Pseudo Random Binary Signal) as excitation input signal. It guarantees the correct excitation of all dynamic sensible modes of the system along the spectral range and thus results in an accurate precision of parameter estimation. The experiments to be realized consist in exciting two DC motors in different (low, medium and high) ranges of speed.

The ARX (auto-regressive with external input) structure has been used to identify the parameters of the robot system. The problem consists in finding a model that minimizes the error between the real and estimated data. By expressing the ARX equation as a lineal regression, the estimated output can be written as:

$$\hat{y} = \theta\varphi \tag{1}$$

with $\hat{y}$ being the estimated output vector, θ the vector of estimated parameters and φ the vector of measured input and output variables. By using the coupled system structure, the transfer function of the robot can be expressed as follows:

$$\begin{pmatrix} Y_R \\ Y_L \end{pmatrix} = \begin{pmatrix} G_{RR} & G_{LR} \\ G_{RL} & G_{LL} \end{pmatrix} \begin{pmatrix} U_R \\ U_L \end{pmatrix} \tag{2}$$

where Y_R and Y_L represent the speeds of right and left wheels, and U_R and U_L the corresponding speed commands, respectively. In order to know the dynamics of robot system, the matrix of transfer function should be identified. Figure 3 shows the speed response of the left wheel corresponding to a left PBRS input signal.

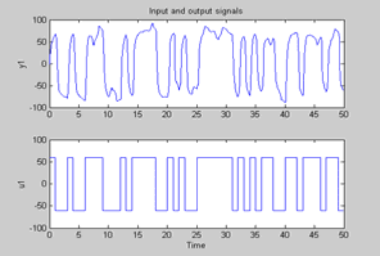


Figure 3: Left speed output for a left PRBS input signal.

The treatment of experimental data is done before the parameter estimation. In concrete, it includes the data filtering, using the average value of five different experiments with the same input signal, the frequency filtering and the tendency suppression. The system is identified by using the identification toolbox "ident" of Matlab for second order models. The following continuous transfer function matrix for medium speed is obtained:

$$\begin{pmatrix} Y_R \\ Y_L \end{pmatrix} = \begin{pmatrix} \dfrac{0.35s^2+4.82s+4.46}{s^2+5.84s+4.89} & \dfrac{0.02s^2+0.27s+0.32}{s^2+5.84s+4.89} \\ \dfrac{0.26s^2+3.41s+0.28}{s^2+5.84s+4.89} & \dfrac{0.11s^2+1.72s+5.12}{s^2+5.84s+4.89} \end{pmatrix} \begin{pmatrix} U_R \\ U_L \end{pmatrix} \tag{3}$$

It is shown by simulation results that the obtained model fits well with the experimental data.

2.2.2 Simplified Model of the System

This section studies the coupling effects and the way for obtaining a reduced-order dynamic model. It is seen from (3) that the dynamics of two DC motors are different and the steady gains of coupling terms are relatively small (less than 20% of the gains of main diagonal terms). Thus, it is reasonable to neglect the coupling dynamics so as to obtain a simplified model.

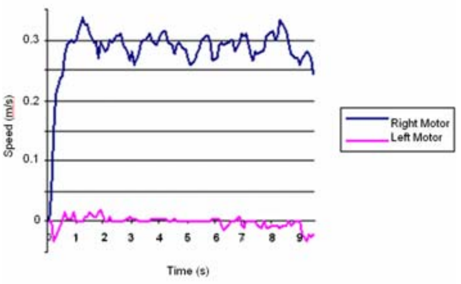


Figure 4: Coupling effects at the left wheel.

In order to verify it from real results, a set of experiments have been done by sending a zero speed command to one motor and other non-zero speed commands to the other motor. In Figure 4, it is shown a response obtained on the left wheel, when a medium speed command is sent to the right wheel. The experimental result confirms the above facts. The existence of different gains in steady state is also verified experimentally. Finally, the order reduction of system model is carried out trough the analysis of pole positions by using the method of root locus. Afterwards, the system models are validated through the experimental data by using the PBRS input signal. A two dimensional array with three different models for each wheel is obtained. Hence, each model has an interval of validity where the transfer function is considered as linear.

2.3 Odometer System Expression

Denote *(x, y, θ)* as the coordinates of position and orientation, respectively. The Figure 5 describes the positioning of robot as a function of the radius of left and right wheels *(R_e, R_d)*, and the angular incremental positioning *(θ_e, θ_d),* with *E* being the distance between two wheels and *dS* the incremental displacement of the robot. The position and angular incremental displacements are expressed as:

$$dS = \frac{R_d d\theta_d + R_e d\theta_e}{2} \qquad d\theta = \frac{R_d d\theta_d - R_e d\theta_e}{E} \tag{4}$$

The coordinates (x, y, θ) can be expressed as:

$$\begin{aligned} x_n &= x_{n-1} + dS\cos\left(\theta_{n-1} + d\theta\right) \\ y_n &= y_{n-1} + dS\sin\left(\theta_{n-1} + d\theta\right) \\ \theta_n &= \theta_{n-1} + d\theta \end{aligned} \tag{5}$$

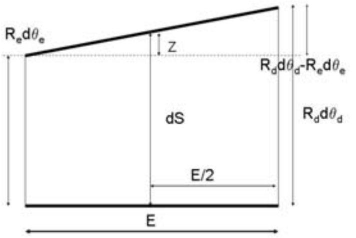


Figure 5: Positioning of the robot as functions of the angular movement of each wheel.

Thus, the incremental position of the robot can be obtained through the odometer system with the available encoder information obtained from (4) and (5).

2.4 Model Predictive Control

The model predictive control, MPC, has many interesting aspects for its application to mobile robot control. It is the most effective advanced control technique, as compared to the standard PID control, that has made a significant impact to the industrial process control (Maciejowski, 2002). Recently, real time mobile robot MPC implementations have been developed using global vision sensing (Gupta, Messom et al., 2005). In (Küne, Lages et al., 2005), it was studied the MPC based optimal control useful for the case when nonlinear mobile robots are used under several constraints, as well as the real time implementation possibilities when short prediction horizons are used. In general, the global trajectory planning becomes unfeasible since the sensorial system of some robots is just local. By using a MPC, the idea of the receding horizon can deal with the local sensor information. In this way, it is proposed a local model predictive control, LMPC, in order to use the available visual data in the navigation strategies for the goal achievement.

The MPC is based on minimizing a cost function, related to the objectives, through the selection of the optimal inputs. In this case, the cost function can be expressed as follows:

$$J(n,m) = \min_{\left\{U(k+i|k)_{j=0}^{j=m-1}\right\}} \left\{ \begin{array}{l} \left[X(k+n|k) - X_d\right]^T P\left[X(k+n|k) - X_d\right] \\ + \sum_{i=1}^{n-1} \left[X(k+i|k) - X_d\right]^T Q\left[X(k+i|k) - X_d\right] \\ + \sum_{j=0}^{m-1} U^T(k+i|k) R U(k+i|k) \end{array} \right\} \tag{6}$$

Denote $X_d=(x_d, y_d, \theta_d)$ as the desired coordinates. The first term of (6) is referred to the final desired coordinate achievement, the second term to the trajectory to be followed, and the last one to the input signals minimization. The parameters *P*, *Q* and *R* are weighting parameters. *X(k+n|k)* represents the terminal value of the predicted output after the horizon of prediction *n* and *X(k+i|k)* represents the predicted output values within the prediction horizon. The system constrains are also considered:

$$\left\{ \begin{array}{ll} \left|U(k+i|k)\right| \le G_1 & \alpha \in [0,1) \\ \left\|\left[x_{k+i}, y_{k+i}\right] - \left[x_o, y_o\right]\right\| \ge G_2 & \\ \left\|\left[x_{k+n}, y_{k+n}\right] - \left[x_d, y_d\right]\right\| \le \alpha \left\|\left[x_k, y_k\right] - \left[x_d, y_d\right]\right\| & \end{array} \right\} \tag{7}$$

The limitation of the input signal is taken into account in the first constraint. The second constraint is related to the obstacle points where the robot should avoid the collision. The last one is just a convergence criterion.

3 THE HORIZON OF LOCAL VISUAL PERCEPTION

The use of sensor information as a useful source to build 2D environment models consists of a free or occupied grid proposed by (Elfes, 1989). The knowledge of occupancy grids knowledge has been used for static indoor mapping with a 2D grid (Thrun, 2002). In other works of multidimensional grids, multi target tracking algorithms are employed by using obstacle state space with Bayesian filtering techniques (Coué et al., 2006). In this work it is proposed the use of the local visual information available from the camera as a local map that has enough information in order to achieve a global objective. The occupancy grid can be obtained in real time by using computer vision methods. The use of the optical flow has been proposed as a feasible obstacle avoidance method; as i.e., (Campbell et al., 2004), in which it was used a Canny edge detector algorithm that consists in Gaussian filtering and edge detection by using Sobel filters. Thus, optical flow was computed over the edges providing obstacle structure knowledge. The present work assumes that the occupancy grid is obtained by the machine vision system. It is proposed an algorithm that computes the local optimal desired coordinate as well as the local trajectory to be reached. The research developed assumes indoor environments as well as flat floor constraints. However, it can be also applied in outdoor environments.

This section presents firstly the local map relationships with the camera configuration and poses. Hence, the scene perception coordinates are computed. Then, the optimal control navigation strategy is presented, which uses the available visual data as a horizon of perception. From each frame, it is computed the optimal local coordinates that should be reached in order to achieve the desired objective. Finally, the algorithm dealing with the visual data process is explained. Some involved considerations are also made.

3.1 Scene Perception

The local visual data provided by the camera are used in order to plan a feasible trajectory and to avoid the obstacle collision. The scene available coordinates appear as an image, where each pixel coordinates correspond to a 3D scene coordinates. In the case attaining to this work, flat floor surface is assumed. Hence, scene coordinates can be computed using camera setup and pose knowledge, and assuming projective perspective. The Figure 6 shows the robot configuration studied in this work. The angles α, β and φ are related to the vertical and horizontal field of view, and the tilt camera pose, respectively. The vertical coordinate of the camera is represented by H.

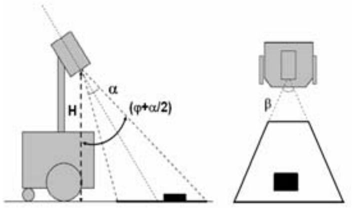


Figure 6: Fixed camera configuration including vertical and horizontal field of view, and vertical tilt angle.

Using trigonometric relationships, the scene coordinates can be computed:

$$y_j = H\tan(\varphi - \alpha/2 + \Delta\alpha) \qquad (8)$$

$$\Delta\alpha = K_j \frac{\alpha}{R} \qquad \left(0 \le \mathrm{K}_\mathrm{j} \le R\right)$$

$$x_{i,j} = \pm \frac{H}{\cos(\varphi - \alpha/2 + \Delta\alpha)} \tan(\Delta\beta) \qquad (9)$$

$$\Delta\beta = K_i \frac{\beta}{C} \qquad \left(0 \le \mathrm{K}_\mathrm{i} \le C/2\right)$$

The K_i and K_j are parameters used to cover the image pixel discrete space. Thus, R and C represent the image resolution through the total number of rows and columns. It should be noted that for each row position, which corresponds to scene coordinates y_j, there exist C column coordinates $x_{i,j}$. The above equations provide the available local map coordinates when no obstacle is detected. Thus, considering the experimental setup reported in Section 2, the local on-robot map depicted in Figure 7 is obtained.

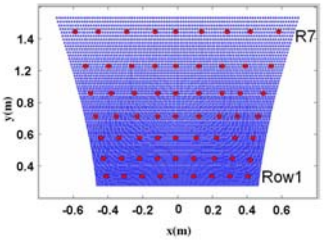


Figure 7: Local visual perception free of obstacles, under 96x72 or 9x7 low resolution grids.

3.2 Local Optimal Trajectory

The available information provided by the camera is considered as a local horizon where the trajectory is planned. Hence, a local map with free obstacle coordinates is provided. In this sense, the available local coordinates are shown in Figure 7. It is noted that low resolution scene grids are used in order to speed up the computing process.

The minimization of a cost function, which consists in the Euclidean distance between the desired coordinates and the available local scene coordinates, can be optimally solved by finding the local desired coordinates. Hence, the algorithm explores the image pixels, *IMAGE(i,j)*, considering just the free obstacle positions. Once the local desired point is obtained, a trajectory between the robot coordinates, at the instant when the frame was acquired, and the optimal scene coordinates is planned. Thus, the current robot coordinates are related to this trajectory, as well as to control methods.

3.3 Algorithms and Constraints

In this subsection, some constraints that arise from the experimental setup are considered. The narrow field of view and the fixed camera configuration make necessary that the robot stays oriented towards the desired coordinates. WMR movements are planned based on the local visual data, and always in advancing sense. Hence, the algorithms provide local desired coordinates to the control unit. If WMR orientation is not appropriate, the robot could turn around itself until a proper orientation is found. Another possibility is to change the orientation in advancing sense by the use of the trajectory/robot orientation difference as the cost function computed over the available visual data. This subsection proposes the local optimal suggested algorithms that have as special features an easy and fast computation. Some methods are presented in order to overcome the drawback of local minimal failures.

3.3.1 The Proposed Algorithms

The proposed algorithm, concerning to obtaining the local visual desired coordinates, consists of two simple steps:

- To obtain the column corresponding to best optimal coordinates that will be the local desired X_i coordinate.
- To obtain the closer obstacle row, which will be the local desired Y_j coordinate.

The proposed algorithm can be considered as a first order approach, using a gross motion planning over a low resolution grid. The obstacle coordinates are increased in size with the path width of the robot (Schilling, 1990). Consequently, the range of visually available orientations is reduced by the path width of WMR. Other important aspects as visual dead zone, dynamic reactive distance and safety stop distance should be considered. The dynamic reactive distance, which should be bigger than the visual dead zone and safety stop distance, is related to the robot dynamics and the processing time for each frame. Moreover, the trajectory situated in the visual map should be larger than a dynamic reactive distance. Thus, by using the models corresponding to the WMR PRIM, three different dynamic reactive distances are found. As i.e. considering a vision system that processes 4 frames each second, using a model of medium speed (0.3m/s) with safety stop distance of 0.25m and an environment where the velocity of mobile objects is less than 0.5m/s, a dynamic reactive distance of 0.45m is obtained. Hence, the allowed visual trajectory distance will set the speed that can be reached. The desired local coordinates are considered as final points, until not any new optimal local desired coordinates are provided. The image information is explored starting at the closer positions, from bottom to upside. It is suggested to speed up the computing process based on a previously calculated LUT, (look up table), with the scene floor coordinates corresponding to each pixel.

3.3.2 Local Minimal Failures

The local minimal failures will be produced when a convergence criterion, similar to that used in (7), is not satisfied. Thus, the local visual map cannot provide with closer optimal desired coordinates, because obstacles blocks the trajectory to the goal. In these situations, obstacle contour tracking is proposed. Hence, local objectives for contour tracking are used, instead of the goal coordinates, as the source for obtaining a path until the feasible goal trajectories are found. The Figure 8 shows an example with local minimal failures. It is seen that in A, the optimal trajectory is a straight line between A and E. However, an obstacle is met at B, and local minimal failure is produced at B. When this is produced, no trajectory can approach to the desired goal, *(Xd, Yd)*. Then, obstacle con-tour tracking is proposed between B and C. Once C is attained, local minimization along coordinates Y is found and the trajectory between C and D is planned. From D to E local minimums are reached until the final goal is achieved. It should be noted that once B is reached, the left or right obstacle contour are possible. However, the right direction will bring the robot to an increasing Y_j distance.

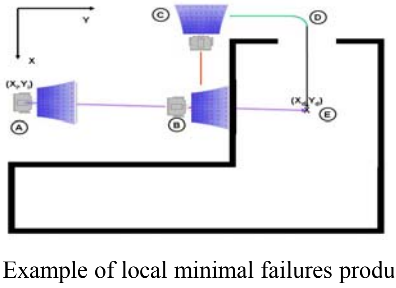


Figure 8: Example of local minimal failures produced at B with A being the starting point and E the desired goal.

The robot follows the desired goals except when the situation of obstacle contour tracking is produced, and then local objectives are just the contour following points. The local minimal failures can be considered as a drawback that should be overcome with more efforts. In this sense, the vision navigation strategies (Desouza, Kak, 2002) should be considered. Hence, it is proposed the use of feasible maps or landmarks in order to provide local objective coordinates that can be used for guiding the WMR to reach the final goal coordinates.

4 LMPC ALGORITHMS

This section gives the LMPC algorithms by using the basic ideas presented in the Section 2. The LMPC algorithm is run in the following steps:

- To read the actual position
- To minimize the cost function and to obtain a series of optimal input signals
- To choose the first obtained input signal as the command signal.
- To go back to the step 1 in the next sampling period

The minimization of the cost function is a nonlinear problem in which the following equation should be verified:

$$f(\alpha x+\beta y)\leq\alpha f(x)+\beta f(y) \qquad (10)$$

It is a convex optimization problem caused by the trigonometric functions used in (5), (Boyd, Vandenberghe, 2004). The use of interior point methods can solve the above problem (Nesterov, Nemirovskii, 1994). Among many algorithms that can solve the optimization, the descent methods are used, such as the gradient descent method among others, (Dennis, et al. 1996), (Ortega, et al. 2000). The gradient descent algorithm has been implemented in this work. In order to obtain the optimal solution, some constraints over the inputs are taken into account:

- The signal increment is kept fixed within the prediction horizon.
- The input signals remain constant during the remaining interval of time.

The input constraints present advantages such like the reduction in the computation time and the smooth behavior of the robot during the prediction horizon. Thus, the set of available input is reduced to one value. In order to reduce the optimal signal value search, the possible input sets are considered as a bidimensional array, as shown in Figure 9.

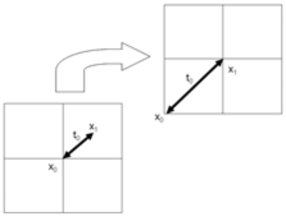


Figure 9: Optimal interval search.

Then, the array is decomposed into four zones, and the search is just located to analyze the center points of each zone. It is considered the region that offers better optimization, where the algorithm is repeated for each sub-zone, until no sub-interval can be found. Once the algorithm is proposed, several simulations have been carried out in order to verify the effectiveness, and then to make the improvements. Thus, when only the desired coordinates are considered, the robot could not arrive in the final point. Figure 10 shows that the inputs can minimize the cost function by shifting the robot position to the left.

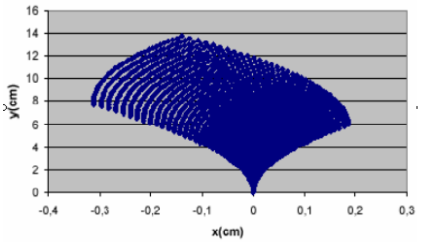


Figure 10: The left deviation is due to the greater left gain of the robot.

The reason can be found in (3), where the left motor has more gain than the right. This problem can be easily solved by considering a straight-line trajectory from the actual point of the robot to the final desired point. Thus, the trajectory should be included into the LMPC cost function. The Figure 11 shows a simulated result of LMPC for WMR obtained by using firstly the orientation error as cost function and then the local trajectory distance and the final desired point for the optimization. The prediction horizons between 0.5s and 1s were proposed and the computation time for each LMPC step was set to less than 100ms, running in an embedded PC of 700MHz. In the present research, the available horizon is provided by using the information of local visual data. Thus, the desired local points as well as the optimal local trajectory are computed using machine vision information.

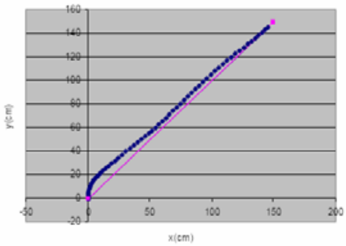


Figure 11: LMPC simulated results with a 45º trajectory.

5 CONCLUSIONS

This paper has integrated the control science and the robot vision knowledge into a computer science environment. Hence, global path planning by using local information is reported. One of the important aspects of the paper has been the simplicity, as well as the easy and direct applicability of the approaches. The proposed methodology has been attained by using the on-robot local visual information, acquired by a camera, and the techniques of LMPC. The use of sensor fusion, specially the odometer system information, is of a great importance. The odometer system uses are not just constrained to the control of the velocity of each wheel. Thus, the absolute robot coordinates have been used for planning a trajectory to the desired global or local objectives. The local trajectory planning has been done using the relative robot coordinates, corresponding to the instant when the frame was acquired. The available local visual data provides a local map, where the feasible local minimal goal is selected, considering obstacle avoidance politics.

Nowadays, the research is focused to implement the presented methods through developing flexible software tools that should allow to test the vision methods and to create locally readable virtual obstacle maps. The use of virtual visual information can be useful for testing the robot under synthetic environments and simulating different camera configurations. The MPC studies analyzing the models derived from experiments as well as the relative performance with respect to other control laws should also be developed.

ACKNOWLEDGEMENTS

This work has been partially funded the Commission of Science and Technology of Spain (CICYT) through the coordinated research projects DPI-2005-08668-C03, CTM-2004-04205-MAR and by the government of Catalonia through SGR00296.

REFERENCES

Boyd, S., Vandenberghe, L., 2004. *Convex Optimization*, Cambridge University Press.

Campbell, J., Sukthankar, R., Nourbakhsh, I., 2004. *Techniques for Evaluating Optical Flow in Extreme Terrain,* IROS.

Coué, C., Pradalier, C., Laugier, C., Fraichard, T., Bessière, P., 2006. *Bayesian Occupancy Filtering for Multitarget Tracking: An Automotive Application*, International Journal of Robotics Research.

Dennis, J.E., Shnabel, R.S., 1996. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Society for Industrial and Applied Mathematics.

DeSouza, G.N., Kak, A.C., 2002. *Vision for Mobile Robot Navigation: a survey*, PAMI, 24, 237-267.

Elfes, A., 1989. *Using occupancy grids for mobile robot perception and navigation,* IEEE Computer, 22, 46-57.

Fox, D., Burgard, W., and Thun, S., 1997. *The dynamic window approach to collision avoidance,* IEEE Robot. Autom. Mag. 4, 23-33.

Gupta, G.S., Messom, C.H., Demidenko, S., 2005. *Real-time identification and predictive control of fast mobile robots using global vision sensor*, IEEE Trans. On Instr. and Measurement, 54, 1.

Horn, B. K. P., 1998. *Robot Vision*, Ed. McGraw –Hill.

Küne, F., Lages, W., Da Silva, J., 2005. Point stabilization of mobile robots with nonlinear model predictive control, Proc. IEEE Int. Conf. On Mech. and Aut., 1163-1168.

Lju, L., 1989. *System Identification: Theory for the User*, ed., Prentice Hall.

Maciejowski, J.M., 2002. *Predictive Control with Constraints*, Ed. Prentice Hall.

Nesterov, Y., Nemirovskii, A., 1994. *Interior_Point Polynomial Methods in Convex Programming*, SIAM Publications.

Norton, J. P., 1986. *An Introduction to Identification*, ed., Academic Press, New York.

Nourbakhsh, I. R., Andre, D., Tomasi, C., Genesereth, M. R., 1997. *Mobile Robot Obstacle Avoidance Via Depth From Focus,* Robotics and Aut. Systems, 22, 151-58.

Ögren, P., Leonard, N., 2005. *A convergent dynamic window approach to obstacle avoidance,* IEEE T Robotics, 21, 2.

Ortega, J. M., Rheinboldt, W.C., 2000. *Iterative Solution of Nonlinear Equations in Several Variables,* Society for Industrial and Applied Mathematics.

Rimon, E., and Koditschek, D., 1992. *Exact robot navigation using artificial potential functions,* IEEE Trans. Robot Autom., 8, 5, 501-518.

Schilling, R.J., 1990. *Fundamental of Robotics*, Prentice-Hall.

Thrun, S., 2002. *Robotic mapping: a survey*, Exploring Artificial Intelligence in the New Millennium, Morgan Kaufmann, San Mateo, CA.

INTERNET-BASED TELEOPERATION: A CASE STUDY
Toward Delay Approximation and Speed Limit Module

Shengtong Zhong, Philippe Le Parc and Jean Vareille
Laboratoire Informatique des Systèmes Complexes (LISyC), Université de Bretagne Occidentale
20, avenue Le Gorgeu, 29285 Brest Cedex, France
{shengtong.zhong, philippe.le-parc, jean.vareille}@univ-brest.fr

Keywords: Remote control, teleoperation, mobile robot, Internet delay, path error, delay approximator, speed limit.

Abstract: This paper presents the internet-based remote control of mobile robot. To face unpredictable Internet delays and possible connection rupture, a direct teleoperation architecture with “Speed Limit Module” (SLM) and “Delay Approximator” (DA) is proposed. This direct control architecture guarantees the path error of the robot motion is restricted within the path error tolerance of the application. Experimental results show the effectiveness and applicability of this direct internet control architecture in the real internet environment.

1 INTRODUCTION

Internet is not only an information highway, but also a mean to remotely control mechanical systems, such as robotic devices. But Internet doesn’t provide a guaranteed Quality of Service (QoS); it entitles a number of limitation and difficulties, such as bandwidth constraint, transmission delays, packet lost, connection rupture etc. The situation above influence the performance of Internet based telerobotics systems, which is a new field in the recent decade.

The Mercury project (Goldberg et al., 1994) is one of the earliest telerobotics implementation over the internet, then coming with the Telerobot in Australia (Taylor and Dalton, 1997), the painting PumaPaint Robot (Stein, 1998), and Khepera robot (Saucy and Mondada, 2000) etc. During the past ten years, lots of such systems have been introduced by different researchers all over the world.

Most of these researches use a supervisory control scheme which enables operator to issue high level commands. As the internet time delay is unpredictable, the design of direct control scheme which enables user to control the motion of robot continuously may not be easy. Such control schemes have been proposed, but are not adequate to alleviate the influence of Internet time delay.

This paper presents direct teleoperation architecture of a continuous robot motion control which meets path error tolerance under the unpredictable Internet time delay. Here, the path error is guaranteed only if the path error at every turning/stop point is restricted within a path error tolerance which depends on application itself.

Two kinds of control strategy are introduced in the Section 2, followed by the detail of “Speed Limit Module” (SLM) with the quality level idea of GEMMA-Q (Ogor, 2001) and how it works together with the “Delay Approximator” (DA) to meet the application requirements. During the Section 3, the software implementation is presented with an application over Miabot (two-wheeled robot widely used in soccer competition organized by FIRA).

2 TELEOPERATION ARCHITECTURE

2.1 Generic Architecture

The teleoperation over Internet lies on a generic teleoperation architecture (Figure 1), through which commands are exchanged between remote system and operator (user). The user, through his Internet navigator, addresses a request to a Web Server (step 1) and downloads an application, such as a Java applet (step 2), on his own workstation. A connection is then established towards the Application Server in charge of robots and client management (step 3). In the same time, another connection with Multimedia Server is also established in the form of exchanging media signals. The user is now able to control the remote robot upon request.

The generic architecture is the same in most applications. The key problem is to alleviate the influence of Internet time delay and towards a continuous control within the path error requirement.

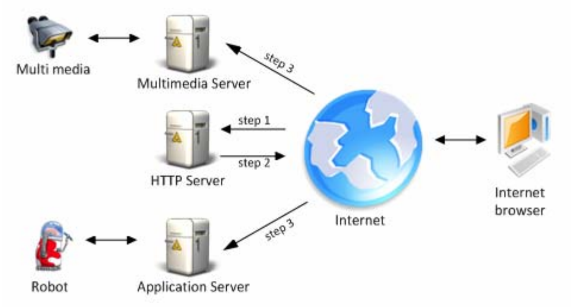


Figure 1: Generic architecture.

2.2 Control Strategies

2.2.1 “Move and Wait” Strategy

The “ Move and Wait” strategy (Sheridan, 1992), which is typical for space robots with long distance communication has been applied first. In the Miabot case, five commands: “move forward” (MF), “move backward” (MB), “turn left” (TL), “turn right” (TR) and “stop” (ST) have been defined. These commands are enough to perform any complex task, but the user has to send lots of commands and change the move or turn parameters from time to time to meet the requirement.

2.2.2 “Speed Control” Strategy

To reduce user’s workload and towards a fluent process during the complex task, a “Speed control” strategy, similar to real driving, is introduced. “No-stop move forward” (NSMF), “no-stop move backward” (NSMB), “turn left” (TL), “turn right” (TR) and “stop” (ST) commands are defined in this strategy. When the robot executes a turning command, it stops running first, perform the entire turning next, and then runs in previous speed again. The user may choose different running speed as well as turning degree.

The user’s operation reduce is significant. The robot is running with speed v during the whole task even though there is a turning/stop point. When the user sends a turning/stop command to robot, the command reaches the robot with a discrete time delay Δt due to Internet time delay. The robot may run an extra distance Δd before it performs the turning/stop.

$$\Delta d = v \times \Delta t$$

Where

v indicates the current speed of the robot, Δt is the current Internet time delay between sending the command from the user and executing the command by robot.

As Internet doesn’t provide a guaranteed Quality of Service (QoS), the time delay is unpredictable. Figure 2 is an execution result of the task A → E with “Speed control” strategy. The path error is accumulative, and path deviation is significant which may result in the failure of the task.

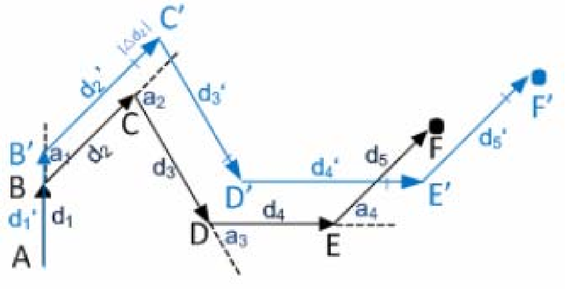


Figure 2: A path result with “Speed Control”.

2.3 Speed Limit Module and Delay Approximator

2.3.1 Speed Limit Module (SLM)

The GEMMA-Q (Ogar, 2001) methodology derived from GEMMA (ADEPA, 1981) has been proposed. The basic idea is to define the quality of communication as a discrete value. Following the value, the automated system has to follow different rules to ensure the security (path error). According to the type of quality, these rules might be different. If the user has a slightly deteriorated communication quality, the user might keep the control and the automated system has to work in deteriorated mode.

The GEMMA-Q has been used in the application over robot arm and prototype machine (Le Parc et al., 2001). In this paper, which aims to control a simple robot, use only the quality level idea of GEMMA-Q to design SLM. n communication quality levels are introduced which corresponds Internet time delay zone (Time D.). Each time delay zone has its own sub path error tolerance $\Delta\varepsilon_j$ (Sub T.) and Max Speed v_j as shown in the Table 1.

Table 1: Quality levels in SLM.

Qual.	Time D.	Sub T.	Max Speed
Q_0	$0 \sim t_0$	$\Delta\delta_0$	$v_0 = \Delta\delta_0 / t_0$
Q_1	$t_0 \sim t_1$	$\Delta\delta_1$	$v_1 = \Delta\delta_1 /(t_1 - t_0)$
Q_2	$t_1 \sim t_2$	$\Delta\delta_2$	$v_2 = \Delta\delta_2 /(t_2 - t_1)$
Q_3	$t_2 \sim t_3$	$\Delta\delta_3$	$v_3 = \Delta\delta_3 /(t_3 - t_2)$
…	…	…	…
Q_{n-1}	$t_{n-1} \sim t_n$	$\Delta\delta_{n-1}$	$v_{n-1} = \Delta\delta_{n-1} /(t_n - t_{n-1})$
Q_n	$\geq t_n$	0	$v_n = 0$

When the quality level changes, the Application Server evaluates the Current Robot Speed (CRS) and the Max Speed of this quality level (MSoT). If CRS $\leq$ MSoT, no command is sent to robot; else the Application Server sends MSoT command to robot (change CRS to MSoT).

MSoT is calculated by the time delay zone and sub tolerance which are defined by user according to application. In order to meet the path error tolerance δ at the turning/stop point, the following constraints are used when designing SLM:

$$\begin{cases} \sum_{j=0}^{n} \Delta\delta_j \leq \delta \\ v_{j+1} \leq v_j \quad j \in (0,1,\cdots n-1) \end{cases}$$

The above constraint guarantees the path error at the turning/stop point is restricted within δ in any Internet time delay situation. Q_0 is the best quality level, its Max Speed v_0 is the fastest running speed of the robot. Q_n is the disconnection situation, and the robot stops running immediately $(v_n = 0)$. The proof of how the constraint works are shown below.

There are two kinds of situation when the robot runs under the speed limit rules.

1. Stable network delay

It means the robot is running in the same time delay quality level Q_i between two continuous actual Internet time delay (from the real clock time of receiving previous actual Internet time delay to the real clock time of receiving next actual Internet time delay). Then

$$\begin{aligned} \Delta d &= v_i \times t_i \\ &\leq v_0 \times t_0 + v_1 \times (t_1 - t_0) + \cdots + v_i (t_i - t_{i-1}) \\ &= \Delta\delta_0 + \Delta\delta_1 + \cdots + \Delta\delta_i \\ &= \sum_{j=0}^{i} \Delta\delta_j \\ &\leq \delta \end{aligned}$$

2. Unstable network delay

The robot is running in the different time delay quality levels between two continuous actual Internet time delays. The worst case is from Q_0 to Q_n:

$$\begin{aligned} \Delta d &= \Delta\delta_0 + \Delta\delta_1 + \cdots + \Delta\delta_n \\ &= \sum_{j=0}^{n} \Delta\delta_j \\ &\leq \delta \end{aligned}$$

The path error Δd is within the restriction of path error tolerance δ in both situations. Then the path error at every turning/stop point in the task is guaranteed and the successfully continuous control of the robot is achieved.

With the constraints, the different quality levels with its respective time delay zone value, sub tolerance and max speed are defined according to different application. e.g.: the normal quality level of Internet time delay is Q_j in the application, its sub tolerance $\Delta\delta_j$ should take a larger percentage of δ. It means the robot is preferred to have larger sub tolerance in the normal quality level; in the same time, speed limit rules guarantees $\Delta d \leq \delta$ in any situation.

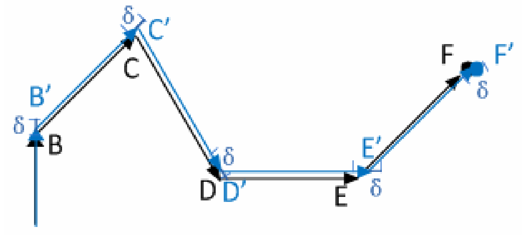


Figure 3: A path result using “Speed Control” with SLM.

In Figure 3, the path deviation at every turning/stop point is restricted within δ. Then a successful task is achieved using “Speed control” with SLM. The above describes the detail SLM with the quality level idea, next part emphasize on how the gets the quality level information of current time delay from DA.

2.3.2 Delay Approximator (DA) Principals

In the generic architecture, Pinger-Ponger is the mechanism to provide the Internet time delay information to DA. The next actual Internet time delay Δt_n is calculated as follows:

$$\Delta t_n = T_n - T_p$$

Where

T_p is the real clock time of receiving previous actual Internet time delay from Pinger-Ponger; T_n is the real clock time when Pinger-Ponger sends the next Internet time delay to DA.With above feature, the current (up-to-now) time delay Δt_c from T_p is formed as:

$$\begin{cases} \Delta t_c = t_j & j = 0 \\ \Delta t_c = t_j - t_{j-1} & j \in (1,2,\cdots n) \end{cases}$$

t_j is the time delay zone in SLM. Δt_c is the watchdog and j is the automatic counter initialized with 0. Δt_c is only activated when there is quality level change, and this quality level change is used as current time delay change information.

Pinger-Ponger only provides the Internet time delay when it gets one. The Internet time delay is unpredictable, so there is no idea when Pinger-Ponger gets new information.

There is no idea about the future, but the current situation is supervised by DA as following principals:

1. DA receives information from Pinger-Ponger.

Pinger-Ponger informs DA of receiving the next actual Internet time delay, and then DA forwards the quality level of calculated Δt_n to SLM. Meanwhile the previous parameters are set to be the current ones: the value of T_p is set to T_n , the quality level of previous actual Internet time is changed to the quality level of Δt_n, reset the watchdog (Δt_c).

2. No information from Pinger-Ponger and Δt_c is activated to a quality level change.

Δt_p is the previous actual Internet time delay. There are two kinds of situations:

- $\Delta t_c \le \Delta t_p$

  The quality level of current time delay is no worse than that of previous actual time delay; there is no action and DA keeps supervising.

- $\Delta t_c > \Delta t_p$

  The quality level of current time delay situation is worse than that of previous actual Internet time delay. When Δt_c is activated, it indicates a change of quality level and “DA” forwards the new quality level to SLM. DA keeps supervising.

DA keeps supervising the Internet delay situation all the time. It provides the real Internet time delay or current time delay to SLM.

3 EXPERIMENT

The remote control system server has been built in UBO, France. Some users (mainly from France, Sweden, and China) are invited to test the feasibility of effectiveness of the system. The average Internet time delay varies a lot between European users and Asian users. It’s not easy to find a uniform SLM for all the users, but certain SLM works fine for some users. The test is still on-going currently. The average Internet time delay of the user is able to observe during the test, so next step consideration is to build SLM for respective user group (distinguished by average Internet time delay. e.g.: Asia, Europe, etc.) and the system can choose different SLM automatically due to the user group.

4 CONCLUSION

In this paper, the SLM and DA based architecture is proposed to face the unpredictable Internet delay in Internet-based robot control. This approach guarantees the path error of the continuous robot motion. Here, the path error is guaranteed only if the path error at every turning/stop point is within a path error tolerance δ which depends on application itself.

In this architecture, the current time delay is supervised by DA; and SLM applies different speed limit rules according to the current time delay situation. Then the robot is always running in a proper speed which meets the path error restriction. Finally, a continuous control of the internet-based robot is achieved successfully.

REFERENCES

Andreu, D., Fraisse, P., Roqueta, V., Zapata, R., 2003. Internet enhanced teleoperation toward a remote supervised delay regulator. ICIT'03, *IEEE Int. Conf. on Industrial Technology, Maribor.*

Ogor, P., 2001. Une architecture générique pour la supervision sure à distance de machines de production avec Internet. *Ph.D. thesis, Université de Bretagne Occidentale.*

ADEPA, 1981. GEMMA: Guide d’études des modes de marche et d’arrêt.

Meng Wang, and James N.K. Liu, 2005. Interactive control for Internet-based mobile robot teleoperation. *Robotics and Autonomous Systems, Volume 52, Issues 2-3, 31 August 2005, Pages 160-179.*

K.-H. Han, J.-H. Kim, 2002. Direct Internet Control Architecture for Personal Robot. *Proc. of the 2002 FIRA Robot World Congress, pp. 264-268, Seoul.*

P. Ogor, P. Le Parc, J. Vareille et L. Marcé, 2001. Control a Robot on Internet. *6th IFAC Symposium on Cost Oriented Automation, Berlin.*

J. Vareille, P. Le Parc et L. Marcé, 2004. Web remote control of mechanical systems: delay problems and experimental measurements of Round Trip Time. *2nd Workshop CNRS-NSF Applications of Time-Delay systems, Nantes.*

OPTICAL NAVIGATION SENSOR

Incorporating Vehicle Dynamics Information in Mapmaking

Tibor Takács and Viktor Kálmán
Department of Control Engineering and Information Technology, Budapest University of Technology and Economics
Magyar Tudósok körútja 2.,Budapest, Hungary
tt488@hszk.bme.hu, kalman@iit.bme.hu

Keywords: Mobile robot navigation sensor, vehicle dynamics, map making.

Abstract: Accurate odometry and navigation may well be the most important tasks of a mobile robot's control system. To solve this task it is necessary to utilize proper sensors which provide reliable information about the motion. This paper presents the prototype of an optical navigation sensor which can be an alternative choice for the dead reckoning navigation system of a vehicle or mobile robot. The last part of the paper presents another application in an inertial navigation system that enables a new approach to map making which incorporates vehicle dynamics into the world map.

1 INTRODUCTION

Autonomous mobile robots gain more and more importance in automotive, domestic and military applications. Sensing and interpreting their surroundings correctly are crucial in all these areas. The sensor presented in this paper aims to improve existing odometry methods, and to aid the creation of world maps that incorporate the state of the vehicle in the world map and classifies terrain accordingly. In the first half of this document an overview of the principle and the prototype of the optical navigation sensor are presented, the second section of the paper proposes a method for world model building for mobile robots and shows some other application where this sensor can be used.

2 NAVIGATION TECHNIQUES

Accurate self localization is one of the most important topics in autonomous vehicle technology. During centuries researchers have developed a lot of techniques to measure the precise position of land or aerial vehicles and mobile robots.

Global methods (e.g. GPS) determine directly the absolute position, but unfortunately in several cases these are not useable, due to reception problems.

Dead reckoning methods estimate the present location of a navigating agent by advancing some previous position using known course, velocity and time information (Borenstein, 1996).

The odometry (the most simplistic and prevalent implementation of dead reckoning) estimates the course and distance of the moving agent's path to calculate the global location by measuring the wheel rotation and/or the steering orientation (Dixon, 1997). This position estimating can be strongly inaccurate under real conditions through the mobility configuration (e.g. tracked vehicles) or through the wheel-slippage, overacceleration or driving over uneven floor.

3 OPTICAL NAVIGATION SENSOR

3.1 Optical Flow

Through of the inaccuracy of the odometry it is necessary to develop a cheap sensor-system that provides well-authenticated dislocation-data for dead reckoning navigation in real world conditions. The optical navigation sensor is a possible solution to fulfil these conditions.

The working principle of the sensor is optical flow, namely the motion information is generated from visual information (from an image-sequence provided by a camera facing the ground). Typically the motion is represented as vectors originating or terminating at pixels in a digital image sequence (CVRG, 2000). An example is shown in figure 1.

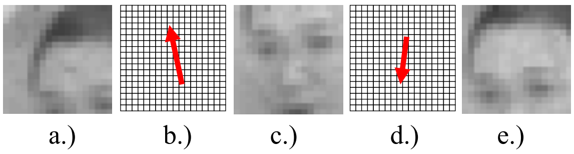


Figure 1: The principle of the optical flow.

3.2 The Prototype of the Sensor

As a basis for further research, we have built the prototype of the optical navigation sensor (see figure 2 and figure 3).

As a camera (what provides the image-sequence for calculating the dislocation) we have chosen the sensing and processing chip of a low-cost and wide-spread optical mouse. (Hereinafter this chip is referred to as "mouse-chip".) The chip includes a low-resolution camera and a DSP for calculating the optical flow by hardware at a very high rate. Although its resolution is relatively low, for motion detection and measurement it is sufficient because only the texture is important. It measures the relative dislocations in the x and y direction.

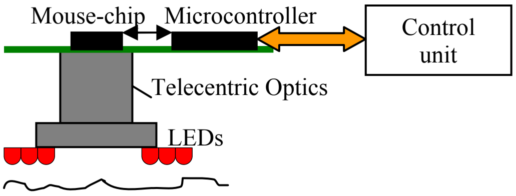


Figure 2: The model of the sensor.

Because of the uneven floor the distance between the sensor and the floor is continuously changing. To compensate this effect the sensor has telecentric optics, which has a constant magnification rate therefore in a certain range the measurements are independent from the height of the sensor relative to the ground.

Naturally, it was necessary to design a microcontroller based architecture round the mouse-chip which offers a useable interface for a dead reckoning navigation system. The microcontroller reads the motion or the image information of the mouse-chip and sends them to the processing unit, for example to a PC in the development stage or to a navigation or control unit at real conditions (Takács, 2007).

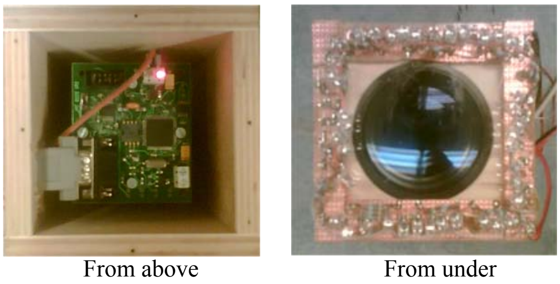


Figure 3: The prototype of the sensor.

3.3 Testing, Error Measurement

To measure the errors of the sensor we mounted it on a test platform (figure 3/a) and performed some experiments.

The first group of tests analyzes the dependence from the texture of the floor and from the height over ground. The experience showed that the sensor is really insensitive to the height (thanks to the telecentric optics) but unfortunately it sometimes misses on shining or totally homogeneous surfaces.

This property is inherited from the mouse-chip of the cheap optical mice. Laser equipped optical mice might be better from this aspect. Therefore using the mouse-chip of a laser mouse presumably solves this problem.

With the second type of the tests we measured the accuracy of the sensor. First we determined the model of the platform then we executed the so called unidirectional square-path test (Borenstein, 1996). The model of the navigating agent and the connection between the motion and the measured sensor values is shown on figure 3/b. The y component measures orientation change, the linear dislocation appears in the value of the x component.

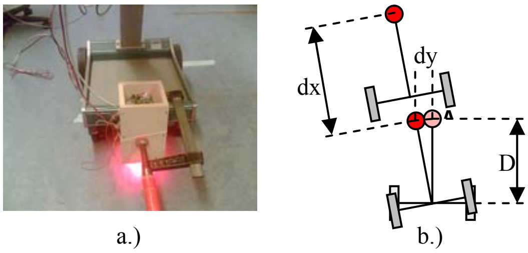


Figure 3: The test platform and its model.

By normal conditions the measured error was relatively small in the course of movement around the square. 4,5 meters of travel yielded 0,01-0,02 meters of the inaccuracy (meaning 0,3% overall error).

It is important to add that slip free wheel contact was supposed. In case of slippage the error grew dramatically, the inaccuracy could even reach the 0,3-0,4 meters (7% error). It was caused by the appearance of linear dislocation in the value of the y component, and the ratio between the linear dislocation and the orientation change can not been determined.

It seems to be a paradox situation since this optical navigation sensor was developed for replacing the wheel encoder based odometry, however we faced the same problem (the inaccuracy from the slippage of the wheels). But as opposed to the odometry, out experiences showed, that this method has a simple and effective solution. By using two optical sensors, a platform independent an accurate dead reckoning navigating sensor-system can be developed that works well under real conditions (wheel-slippage, travel over uneven floor etc.) too.

4 ALTERNATIVE APPLICATIONS

4.1 Electronic Stability Control of Vehicles

One of the most critical areas of stability systems (ESP, ABS etc.) is the accurate measurement of the dynamic state of the vehicle. Probably the most important state variable is the slip angle of the moving vehicle. It is needed for the calculation of the force between the wheels and the ground and also for the exact description of the motion-state.

Presently used measurement procedures of the slip angle are in general very expensive therefore it they are not really usable in mass production (Kálmán, 2005). Hence it is necessary to use estimation methods to provide the value of the slip angle. Unfortunately these procedures can not provide accurate results (Bári, 2006).

This optical sensor provides an effective and cheap solution to solve the problem of the slip angle because it can measure (instead of estimation) directly the slip angle, the ground velocity in 2D and other important motion information. Therefore it can be an additional sensor of any vehicle handling enhancement system, for example ABS or ESP. Naturally for utilization in the automotive industry it is necessary to fulfil numerous standards and reliability requirements, which will not be discussed in this paper.

4.2 Linking Vehicle Dynamic Properties into World Map Building

Another interesting area of application is a multi sensor navigation unit of a mobile robot or automated vehicle. By fusing information from the optical correlation sensor and other sensors measuring the vehicles inner state - such as acceleration, tilt, wheel speed, GPS sensors - very precise dynamical information can be obtained. This information can be used to build a world map that contains information not only about the environment but also its effects on the dynamical behaviour of the robotic vehicle. The optical sensor plays an important role in this unit because it is able to measure true ground velocity relative to our vehicle, making us able to determine wheel slippage and slip angle.

Intelligent mobile robots navigate around in their environment by gathering information about their surroundings. The most common approach is to use ranging sensors mounted on the robot to form occupancy grids or equivalent. Other approaches avoid this metric division of space and favour topological mapping. By combining these mapping techniques it is possible to form a hierarchical map that has the advantages of both methods while some of the disadvantages can be avoided (Thrun, 1998).

Occupancy grids classify the individual cells based on range data and possibly other features such as colour or surface texture or variation. This becomes very important in outdoor mobile robotics when the robot needs to distinguish between real obstacles and traversable terrain. An extreme case is given by navigation in a field of tall grass. The elevation map will represent the scene as a basically horizontal surface above the ground level; that is, as a big obstacle in front of the vehicle. It is apparent that only by integrating the geometry description with terrain cover characterization will a robot be able to navigate in such critical conditions (Belluta, 2000).

Topological maps describe the world in terms of connections between regions. This is usually enough indoors, or in well structured environments, but when travelling through more complex terrain a different representation might be necessary. For example a sloping gravel road or sand dune may only be traversable at a certain speed or only one way, up or downwards. By applying information from the inertial navigational unit, such as slope angle, wheel slippage, actual movement versus desired movement, these characteristics can be learned (or used from apriori information) and the connections of the topological graph can be updated accordingly.

Terrain characteristics (and those of our vehicle) determine the maximum safe speed, braking distance curve radius at a given speed, climbing manoeuvres etc. It is obvious that the more information we have about a certain region we are planning to travel through, the more driving efficiency we can achieve, as it is generally unsafe to drive at high speed through bumpy terrain or make fast turns on a slippery surface. By incorporating the data from the navigational unit into the world map, we can associate driving guidelines to a given map segment. Also on the higher, topological level - using apriori information - we can identify the type of the terrain for a given point of our topological graph, as office environment, forest, urban area, desert etc. By doing so, we narrow down our choices when making decisions about terrain coverage. For example it is unlikely to encounter sand, water or foliage in an office environment. If we know the type of terrain ahead we can make a more accurate estimate of the driveability of the area thus increasing driving efficiency.

In this section a hierarchical map making method was proposed which uses data from a multi-sensor navigation unit that supplies information about vehicle dynamics. This unit heavily relies on the optical correlation sensor described in the preceding sections. By measuring wheel slip and vehicle slip angle we are able to associate drivability guidelines such as safe speed, friction coefficient, minimal driving speed etc. to a given map segment or type of terrain. A higher level of environment recognition was also proposed: based on apriori information, or sensor data the vehicles control system decides the type of environment (e.g. office, forest, desert) the robot traverses at the time, and changes the probability of terrain types, characteristic of the type of environment, thus simplifying terrain classification.

5 SUMMARY

In the previous sections we presented an optical navigation sensor which measures motion information (velocity, dislocation, slip angle) without ground contact.

An alternative dead reckoning technique is proposed in section 3, that yields superior accuracy compared to wheel encoder based methods. In the first part a short overview of principle (e.g. optical flow) is given followed by the description of our experimental setup. Experimental results are given in the last part. Conclusions: the current system should be made platform independent by using two rigidly linked sensors, use of laser mouse chips is recommended to overcome the problem of texture dependency.

Finally examples of alternative application areas were presented: slip angle measurement for the safety systems of vehicles and hierarchical map building with additional driveability information.

As a conclusion we can say that the device and dependent methods presented here can serve as cheap and accurate alternative solutions to numerous problems of the robot and automotive industry.

REFERENCES

Borenstein, J., Everett, H. R., Feng, L., 1996. “Where am I?” Sensors and Methods for Mobile Robot Positioning, 13-29., 130-150.

Takács, T., 2007. Alternative Path-Tracking Sensor-System for Mobile Robots, microCAD – International Scientific Conference.

Bári, G., Kálmán V., Szabó, B., 2006. Járműdinamikai állapotbecslő algoritmus fejlesztése. A jövő járműve – Járműipari innováció, BME-EJJT, in Hungarian.

Kálmán, V., 2005. Elemzés az "optical flow" (~képfolyam) elven működő jármű és mobil robot-technikai érzékelők kínálatáról, in Hungarian.

Dixon, J., Henlich, O., 1997. Mobile Robot Navigation, SURPISE. Retrieved from http://www.doc.ic.ac.uk/~nd/surprise_97/journal/vol4/jmd/

Horn, B.K.P., Schunck, B.G., 1980. Determining optical flow. AI Memo 572. Massachusetts Institue of Technology.

Thrun, S., 1998. Learning maps for indoor mobile robot navigation. Artificial Intelligence, 99(1): 21–71..

Belluta, P., Manduchi, R., Matthies, L., Owens, K., Rankin, A., 2000. Terrain Perception for Demo III. Proceedings of the Intelligent Vehicles Symposium. Dearborn, Michigan.

VISION-BASED OBSTACLE AVOIDANCE FOR A SMALL, LOW-COST ROBOT

Chau Nguyen Viet and Ian Marshall
Computer Science Department, University of Kent, Canterbury, United Kingdom
cn41@kent.ac.uk, ian.marshall@physics.org

Keywords: Obstacle-avoidance, robot vision.

Abstract: This paper presents a vision-based obstacle avoidance algorithm for a small indoor mobile robot built from low-cost, and off-the-shelf electronics. The obstacle avoidance problem in robotics has been researched extensively and there are many well established algorithms for this problem. However, most of these algorithms are developed for large robots with expensive, specialised sensors, and powerful computing platforms. We have developed an algorithm that can be implemented on very small robots with low-cost electronics and small computing platforms. Our vision-based obstacle detection algorithm is fast and works with very low resolution images. The control mechanism utilises both visual information and sonar sensor's measurement without having to fuse the data into a model or common representation. The robot platform was tested in an unstructured office environment and demonstrated a reliable obstacle avoidance behaviour.

1 INTRODUCTION

Obstacle avoidance is one of the most fundamental problems in the field of mobile robotics. Despite the problem being studied extensively, a reliable obstacle avoidance behaviour in a dynamics and unstructured environment, i.e. an environment that is not modified specifically to suit the robot, is still very hard to achieve especially for small robots. Vision can be used to detect obstacles and one of the developed class of algorithms is based on colour-based terrain segmentation (Lorigo et al., 1997; Lenser and Veloso, 2003; Ulrich and Nourbakhsh, 2000). If we can assume that a robot with a digital camera pointing forward is operating on a flat surface and all objects have their bases on the ground, then the distance from the robot to an object is linear to the y-axis coordinate of the object's appearance in the perceived image. We have developed a similar algorithm that utilises a low resolution digital camera and a low powered micro processor. What makes our algorithm different from existing algorithms is the use of a lookup map for colour classification and a reduced colour space. Lookup map is a very fast classification method. On a Gumstix computer clocks at 200 MHz, our algorithm can process more than 500 frames of $87 * 44$ pixels per second. The vision algorithm presented in (Lenser and Veloso, 2003) uses 3 array access operations and an AND bitwise operations for each pixel. Our algorithm uses only one array access operation. Lugino and her group developed an algorithm that can work with low resolution image 64*64 pixels frame in (Lorigo et al., 1997). Our algorithm works with even lower resolution of 22*30 pixels frame. This reduces the computing cycle required for the vision algorithm and enables our algorithm to run on embedded computing devices. Our robot is small, less than a kilo, and energy efficient; it is powered by AA batteries. We present both the vision algorithm and the robot design.

Due to the camera's narrow field of view (FOV), two sonar sensors were added to expand the robot's FOV. The control mechanism is reactive, it has no memory and acts upon the most current sensor readings only. This allows the robot to respond quickly to changes in the environment. The approach we used is inspired by the subsumption architecture (Brooks, 1985) and Braintenberg vehicles (Braitenberg, 1984). The obstacle avoidance algorithm might be used as a module in a more complex system e.g. the first level of competence in a subsumption architecture. It can be used on it own in applications such as exploration, surveillance. Because only a small fraction of the CPU is required for obstacle avoidance, more spaces are available for complex behaviours.

This paper is organised as follows. In section II, we present the vision algorithm and control mechanism. Section III describes the hardware configuration and software implementation. The experiments and results are reported in section IV.

2 VISION AND CONTROL ALGORITHM

2.1 Ground and Obstacles Segmentation

In our classification algorithm, pixels are classified according to their colour appearance only. The colour space we use is the RGB colour space. Each colour in the colour space is set to be a ground or obstacle colour. This classification information is stored in a binary lookup map. The map is implemented as a three dimensions vector of integers. To classify a pixel, its RGB components are used as indices to access the class type of the pixel. The classification process is very fast since for each pixel only one array lookup operation is needed.

The lookup map is populated from example pictures of the ground. First, the algorithm counts the number of pixels of each colour in the example pictures. Then if the number of pixels of a colour is more than 5% of the total number of pixels in those pictures, that colour is set to be a ground colour. The 5% threshold is used to eliminate noises in the images. Procedure 1 describes this calibration process. A lookup map is also very efficient for modification. At the moment, the calibration process is done once before the robot starts moving and the lookup map remains unchanged. We anticipate that the colour appearance of the ground and the lightning condition are likely to change if the robot operates for a long period or moves into different environments therefore any classification technique is required to adapt to these changes. In the near future, we plan to implement an on-line auto-calibrating algorithm for the vision module. Procedure 2 describes how a pixel is classified during the robot's operation.

In a constrained platform the amount of memory needed to store the full 24 bits RGB space is not available. To overcome this problem, we reduce the original 24 bits RGB colour space to 12 bits and decrease the size of lookup table from 2^{24} elements to 2^{12} elements. Effectively, we make the classifier more general since each element of the reduced table represents a group of similar colours in the original space. We also use very low resolution images of $22 * 30$ pixles. Fig. 1 has two examples of the outputs from this segmentation procedure. At the top row is a picture taken from the camera mounted on our robot at the maximum resolution and the binary image produced by the segmentation procedure. At the bottom row is the down-sampling version of the top row picture and its corresponding binary image.

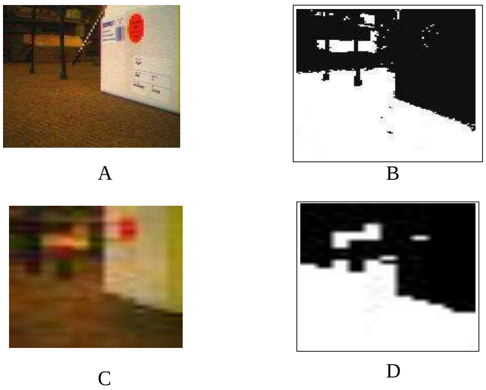


Figure 1: The images processed by the vision module. A is a full resolution colour image. B is the binary obstacle-ground image of A. C is a low resolution image from A and D is its corresponding binary image.

The output of the image segmentation is a binary image differentiating obstacles from the ground. Assuming all objects have their bases on the ground, the distance to an object is the distance to its base. This distance is linear to the y-coordinate of the edge between the object and the ground in the binary image. For obstacle avoidance, we only need the distance and width of obstacles but not their height and depth. Therefore a vector of distance measurements to the nearest obstacles is sufficient, we call this obstacle distance vector (ODV). We convert the binary image to the required vector by copying the lowest y-coordinate of a non-floor pixel in each column to the corresponding cell in the vector. Each element of the vector represents the distance to the nearest obstacle in a specific direction.

2.2 Control Algorithm

The control algorithm we adopted is reactive, decisions are made upon the most recent sensory readings. The inputs to the controller are the obstacle distance vector, produced by the visual module, and distance measurements from the two sonar sensors pointing at

Procedure 1 PopulateLookupMap (n: number of pixels , P : array of n pixels.

```
for $i = 0$ to $n - 1$ do
    $(R, G, B) =\Leftarrow rescale(P[i]_r, P[i]_g, P[i]_b)$
    $pixel\_counter[R][G][B] \Leftarrow pixel\_counter[R][G][B] + 1$
end for
for (R,G,B) = (0,0,0) to MAX(R,G,B) do
    $is\_ground\_map[R][G][B] \Leftarrow pixel\_counter[R][G][B] > n * 5\%$
end for
return $is\_ground\_map$
```

Procedure 2 is_ground(p : pixel).

```
$(R, G, B) \Leftarrow rescale(p_r, p_g, p_b)$
return is_ground_map[R][G][B];
```

the sides of the robot. The ODV gives a good resolution distance map of any obstacle in front of the robot. Each cell in the vector is the distance to the nearest obstacle in a direction of an angle of about 2.5°. The angular resolution of the two sonar sensors are much lower. So the robot has a good resolution view at the front and lower at the sides. The controlling mechanism consists of several reflexive rules.

- If there are no obstacles detected in the area monitored by the camera, run at maximum speed.
- If there are objects in front but further than a trigger distance, slow down.
- If there are objects within the trigger distance, start to turn to an open space.
- If a sonar sensor reports a very close object, within 5 *cm*, turn to the opposite direction.

The control algorithm does not calculate how far the robot should turn. It will keep turning until the area in front is clear. The robot looks for an open space by first looking in the opposite direction to the perceived obstacle, if the half image in that side is free of obstacles, the robot will turn to this direction. If there are obstacles in both left and right half of the image, the two measurements from sonar sensors are compared and the robot will turn to the direction of the sonar sensor that reports no existence of obstacles or a biger distance measurement. There is no attempt to incorporate or fuse data from the camera and sonar sensors together into a uniformed representation. The algorithm uses the sensor readings as they are.

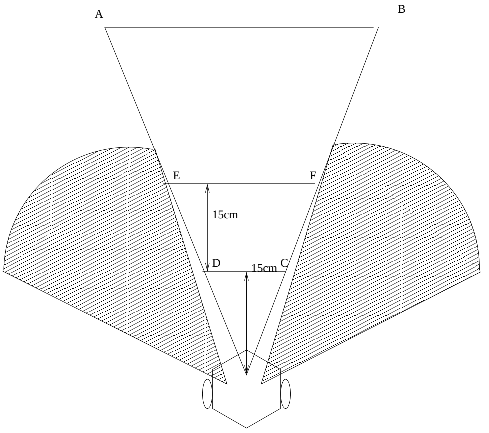


Figure 2: A visualisation of the monitored area. ABCD : the area captured by the camera. Shaded areas represent the sonar sensors views. Segment EF is the trigger distance line.

3 PLATFORM CONFIGURATION AND IMPLEMENTATION

The robot control software runs on a Gumstix (gum,), a small Linux computer that has an Intel 200 MHz ARM processor with 64 Mb of RAM. The vision sensor is a CMUcam2 module connected to the Gumstix via a RS232 link. A Brainstem micro-controller is used to control sonar sensors and servos. The robot is driven by two servos. These electronic devices are mounted on a small three wheeled robot chassis. The total cost of all the components is less than 300 US dollars. The robot can turn on the spot with a small radius of about 5 *cm*. Its maximum speed is $15cm/s$. The robot is powered by 12 AA batteries. A fully charged set of batteries can last for up to 4 hours. Fig. 2 shows the area in front of the robot that is monitored by the robot's sensors. The CMUcam is mounted on the robot pointing forward at horizontal level,15 cm above the ground, and captures an area of about $75cm^2$. Because the camera has a relatively narrow FOV of about 55°, the two sonar sensors on the side are needed. In total, the robot's angle of view is 120°. The robots dimensions are $20cm * 20cm * 15cm$. The hardware configuration was determined by the trial and error method. The parameters we presented here were the best that we found and were used in the experiments reported in section IV.

The maximum frame resolution of the CMUCam2 is $87 * 144$ pixels, we lower the resolution to only $22 * 30$ pixels. We only need the first bottom half of the picture so the final image has dimensions of $22 * 15$. In our implementation, the obstacle distance vector has 22 cells, each cell corresponds to an angle of 2.5° . The vector is then outputted to the controller. Since the control algorithm doesn't build a model, there is no need to convert the pixel's y-coordinates to an absolute measurement e.g. cm or inch. Because the resolution of the images is very low, the distance estimation is not very accurate. At the lowest row of the image, where the ratio between pixel and the projected real world area is highest, each pixel represents an area of $2*1.5cm^2$.

The distance that triggers the robot to turn is set to 30 *cm*. The robot needs to turn fast enough so that an obstacle will not be closer than 15 *cm* in front of it since the distance of any object in this area can not be calculated correctly. At maximum speed , the robot will have about two seconds to react and if the robot has already slowed down while approaching the object, it will have about three seconds. We have tried different combinations of trigger distances and turning speeds to achieve a desirable combination. The first criteria is that the robot must travel safely, this criteria sets the minimum turning speed and distance. The width of view of the camera at the distance of 30 *cm* from the robot or 35 *cm* from the camera is 30 *cm*. The width of our robot is 20 *cm*, so if the vision module does not find an obstacle inside the trigger range, the robot can safely move forward. The second criteria is the robot needs to be able to go to cluttered areas. This means it should not turn too early when approaching objects. Also when the robot is confronted by the wall or a large object, it should turn just enough to move along the wall/object and not bounce back. This criteria encourages the robot to explore the environment.

4 EXPERIMENTS

4.1 Experiment Setup and Results

We tested the robot in two environments, a $1.5*2.5m^2$ artificial arena surrounded by 30*cm* height walls and an office at the University of Kent Computing department, shown in Fig. 3. The surface of the artificial arena is a flat cartoon board with green wallpapers on top. We put different real objects such as boxes, shoes, books onto the arena. We first tested the robot in the arena with no objects (the only obstacless are walls) and then made the tests more difficult by adding objects. The office is covered with a carpet. The arena presents a more controlable environment where the surface is smooth and relatively colour-uniformed. The office environmnent is more challenging where even though the ground is flat its surface is much more coarse and not colour-uniformed.

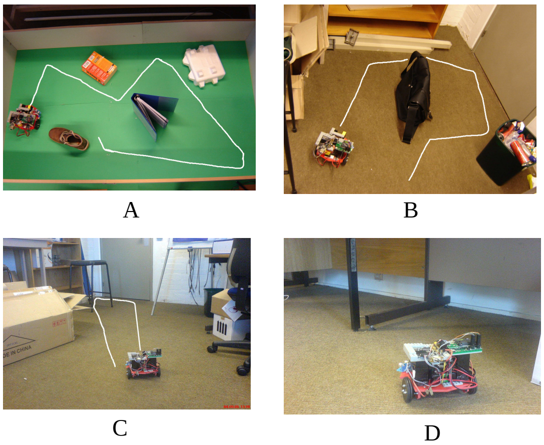


Figure 3: Snapshots of the robot in the test environments and its trajectories. A: the artificial arena with 4 objects. B: A small area near the office corner. C: A path that went through a chair's legs. D: An object with no base on the ground.

For each test, the robot run for 5 mins. We placed the robot in different places and put different objects into the test area. In general, the robot is quite competent; Table I summaries the result. The vision-based obstacle detection module correctly identified obstacle with almost 100% accuracy, that is if there was an obstacle in the camera view, the algorithm would register an non-ground area. Although the calculated distances of obstacles are not very accurate, they provide enough information for the controller to react. The simple mechanism of finding an open space worked surprisingly well. The robot was good at finding a way out in a small area such as the areas under tables and between chairs. The number of false positives are also low and only occured in the office environment. This is because the office's floor colours are more difficult to capture thoroughly. Further analysis revealed that false positives often occurred in the top part of the images. This is explained by the ratio of pixels/area in the upper part of the image being lower than the bottom part. At the top row of the image, each pixel corresponds to an area of $7*4cm$ while at the bottom row the area is $2*1.5cm$. Fortunately, the upper part also corresponds to the further area in real world. Therefore, most false positive cases resulted in unnecessary decreasing of speed but not changing direction. Because of the robot's reactive behaviour, it is capable of responding quickly to changes in the environments. During some of the tests, we removed and put obstacles in front of the robot. The robot could react to the changes and altered it's running direction accordingly.

Fig. 3 shows 4 snapshots of the robot during operation and its trajectory. In picture A, the robot ran in the arena with 4 obstacles, it successfully avoided

Table 1: Performance summary.

Environment	No of Obstacles	Duration	Average speed	No. of collisions	False positive
Arena	0	60 min	13 cm/c	0	0%
Arena	1	60 min	10 cm/s	0	0%
Arena	4	60 min	6 cm/s	2	0%
Office	> 10	180 min	9 cm/s	7	3%

all the objects. On picture B, the robot went into a small area near a corner with a couple of obstacles and found a way out. On picture C, the robot successfully navigated through a chair's legs which presented a difficult situation. Picture D was a case where the robot failed to avoid an obstacle. Because the table leg cross-bar is off the floor, the robot underestimated the distance to the bar.

4.2 Discussion

We found that adding more obstacles onto the arena did not increase the number of collisions significantly. However the average speed of the robot dropped as the arena became more crowded. The speed loss is admitedly due to the robot's reactive behaviour. The robot does not do path planning therefore it can not always select the best path. In some cluttered areas, the robot spent a lot of time spinning around before it could find a viable path. We can improve the robot's behaviour in these situations by having a mechanism to detect cluttered and closed areas so the robot can avoid them.

Each control cycle takes about 150*ms* or 7Hz. Table II shows the time spent on each task in the cycle. The algorithm used only 15% of the CPU during operation. This leaves plenty of resources for higher behaviours if needed. It is possible to implement this algorithm with a less powerful CPU. Since only 10% of the CPU time is spent on processing data, a CPU running at 20 MHz would be sufficient. So instead of the Gumstix computer we can use a micro-controller such as a Brainstem or a BASIC STAMP for both image processing and motion control without any loss in performance. The memory usage is nearly one Mb which is rather big. We did not try to optimise memory usage during implementation so improvements could be made. We plan to implement this control algorithm on a micro-controller instead of the Gumstix. This change will reduce the cost and power usage of the robot by a large amount. To the best of our knowledge, there has not been a mobile robot that can perform reliable obstacle avoidance in unconstrained environments using such low resolution vision and slow microprocessor.

Table 2: Speed performance.

Task	Time
Image acquiring	95 ms
Sonar sensor reading	125 ms
Image processing	5 ms
Controller	1 ms
Servos updating	$< 1ms$
Logging	3 ms
Total	150 ms

REFERENCES

http://www.gumstix.org.

Braitenberg, V. (1984). *Vehicles: Experiments in Synthetic Psychology*. MIT Press/Bradford books.

Brooks, R. A. (1985). A robust layered control system for a mobile robot. Technical report, MIT, Cambridge, MA, USA.

Lenser, S. and Veloso, M. (2003). Visual sonar: fast obstacle avoidance using monocular vision. In *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 1, pages 886–891.

Lorigo, L., Brooks, R., and Grimsou, W. (1997). Visually-guided obstacle avoidance in unstructured environments. In *Intelligent Robots and Systems, 1997. IROS '97., Proceedings of the 1997 IEEE/RSJ International Conference on*, volume 1, pages 373–379, Grenoble, France.

Ulrich, I. and Nourbakhsh, I. R. (2000). Appearance-based obstacle detection with monocular color vision. In *AAAI/IAAI*, pages 866–871.

A DISTRIBUTED MULTI-ROBOT SENSING SYSTEM USING AN INFRARED LOCATION SYSTEM

Anssi Kemppainen, Janne Haverinen and Juha Röning
Department of Electrical and Information Engineering, University of Oulu, Finland
{*pikinen, johannes, jjr*}*@ee.oulu.fi*

Keywords: Distributed sensing, relative pose estimation, multi-robot system, formation control.

Abstract: Distributed sensing refers to measuring systems where, instead of one sensor, multiple sensors are spatially distributed to improve the robustness of the system, increase the relevancy of the measurements and cut costs, since smaller and less precise sensors are used. Spatially distributed sensors fuse their measurements into the same co-ordinates, which requires the relative positions of the sensors. In this paper we present a distributed multi-robot sensing system in which the relative poses (positions and orientations) of the robots are estimated using an infrared location system. The relative positions are estimated using intensity and bearing measurements of received infrared signals. The relative orientations are obtained by fusing the position estimates of the robots. The location system enables a group of robots to perform distributed and co-operative environment sensing by maintaining a given formation while the group measures distributions of light and a magnetic field, for example. In the experiments, a group of three robots moved and collected spatial information (i.e. illuminance and compass headings) from the given environment. The information was stored on grid maps that present illuminance and compass headings. The experiments demonstrated the feasibility of using the distributed multi-robot sensing system in mobile sensing applications.

1 INTRODUCTION

Distributed sensing (Brooks and Iyengar, 1998), (Catterall et al., 2003) refers to measuring systems where, instead of one sensor, multiple sensors are spatially distributed to improve the robustness of the system, increase the relevancy of the measurements and cut costs, since smaller and less precise sensors are used. In the robotic domain, distributed sensing enables multi-robot systems (MRS) to perform mapping and exploration (Sujan et al., 2004), allocate tasks among robots (Pagello et al., 2006), and plan paths and navigate in an unknown or partially unknown environment (Cai et al., 1996), for example.

Distributed sensing in an unknown enviroment benefits from knowing the relative locations of sensors. The relative locations enable the sensors to place their measurements into the same sensor-centric map. Autonomous sensing systems are able to use one sensor as an origin of the co-ordinates, while the other sensors measure the surrounding area. Such systems do not require an external infrastructure for positioning (GPS, WLAN, etc.). However, the measurement range of the relative location system must be sufficient to encompass the sensing area.

In this paper we present a distributed multi-robot sensing that uses an infrared location system (Kemppainen et al., 2006). The location system estimates the relative poses (positions and orientations) of the robots. Related systems have been presented that exploit several techniques, including laser range finders (Schneider and Wildermuth, 2004), (Montesano et al., 2004), (Howard et al., 2003),(Moors et al., 2003), ultrasonic TOF measurement (Shoval and Borenstein, 2001), (Grabowski et al., 2000) and vision (Montesano et al., 2005), (Spletzer et al., 2001), to locate and recognise other robots. A comparison of the infrared location system and related systems was discussed in (Kemppainen et al., 2006). We selected infrared sensors since they are small and capable of relative angle measurements between an emitter and receiver. In addition, infrared radiation does not reflect from walls and object surfaces as much as ultrasound. In relative angle measurements, the effects of multipath reflection would be crucial because of ambiguous angle estimates.

To demonstrate the distributed sensing system in mobile sensing applications, we conducted experiments where a group of three robots measured the distributions of illuminance and a magnetic field while maintaining a triangle formation.

2 INFRARED LOCATION SYSTEM

The infrared location system, originally presented in (Kemppainen et al., 2006), is a vital part of the multi-robot system (Haverinen et al., 2005) that enables the robots to maintain a given formation while sensing the environment. The key idea of the location system is to estimate the relative positions without data transmission between the robots. However, radio transmission is used to share estimates among the group in order to enable the robots to estimate their relative orientations.

The location system uses intensity and bearing measurements of received signals to estimate the locations of other robots in polar co-ordinates. By sharing these estimates among the group of robots, it is possible to also estimate their relative orientations. In addition, each robot is identified through different frequencies in the received signal.

Figure 1 presents the main components of the location system. A conical mirror reflects an emitted signal sideways into a unified zone, whereas beam collector collects signals from other robots. A servo system with a DC motor, Hall-effect sensors and discrete PID controller is used to rotate the beam collector at a constant angular velocity. The measurement range of the system is approximately five metres, giving the most accurate estimates for radial co-ordinates when the distance between two robots is in the range of [0.5; 2.5 m].

3 EXPERIMENTS

Experiments were conducted to demonstrate the feasibility of using the distributed multi-robot sensing system in mobile sensing applications. For example, co-operative mapping in an unknown environment requires moving platforms that are capable of measuring spatial information and estimating the relative positions of the robots. For our experiments, we implemented a distributed sensing system in which a group of three robots measured spatial information in a given environment.

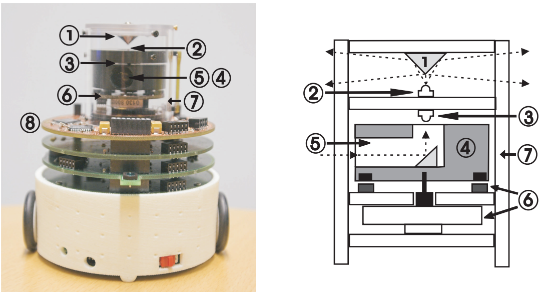


Figure 1: The actual system and the illustration of mechanics: 1) mirror, 2) emitter, 3) receiver, 4) beam collector, 5) aperture, 6) DC motor and Hall-effect-sensors, 7) see-through body, 8) control electronics.

3.1 Formation Control

Formation control enables a multi-robot system to measure spatial distributions while moving across the measurement area. The measurement range of the infrared location system is restricted to five metres, and to be able to fuse spatial measurements onto the same map, the multi-robot system is required to maintain limited relative distances. Formation control is required not only to limit the distance between the robots, but also to enable co-ordinated sensing in order to reduce mapping time.

In our experiments one of the robots acted as a leader and the other two followed the leader. Together they constituted a right-angle triangular formation where the distance between the leader and the followers was 1.5 metres. For each following robot, formation control consisted of two P controllers; one for translational and the other for rotational speed control. The infrared location system updated the relative pose measurements approximately every three seconds, giving the relative poses of the following robots in the leader's co-ordinates. Fig. 2 presents how errors in the relative orientation and position were used in the rotational and translational speed controllers (respectively).

The control cycle length in each controller was 200 ms, while the location system updated poses only once every three seconds. In order to update pose errors in each control cycle, each robot estimated their relative movements using odometer readings. In the experiments, the odometer readings enabled the formation to be driven at a 10 cm/s translational velocity.

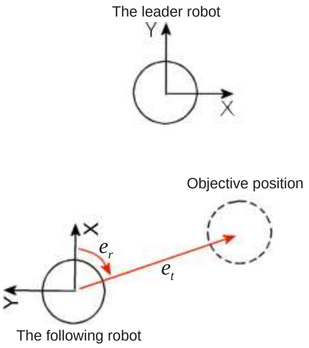


Figure 2: The pose of the following robot relative to the leader is estimated and used to control the robot to the objective position. P controllers use rotation e_r and translation e_t errors to control rotational and translational speeds of the robot.

3.2 Spatial Measurements

In the experiments, three robots measured distributions of light and a magnetic field over a given environment, producing maps of illuminance and compass headings. Fig. 3 presents the trajectories of the robots while moving and measuring the environment. The leader was driven from position (50 cm, -250 cm) to position (20 cm, 450 cm), while the group maintained a triangular formation. Spatial measurements were stored on grid maps presented in the leader's coordinates.

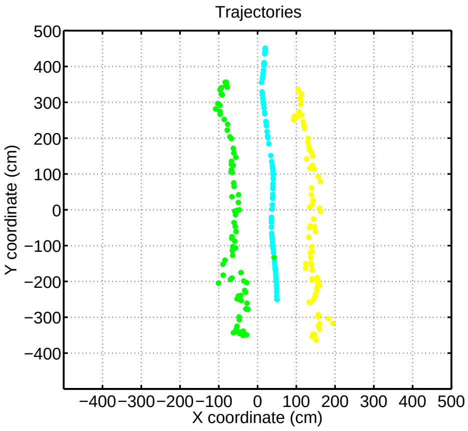


Figure 3: Trajectories in global coordinates.

Fig. 4 presents a grid map of illuminance, where the highest intensity is depicted with white colour and the lowest intensity, for the cells not visited, with black colour. This gives spatial information about the distribution of light in the given environment. The cells with the highest intensities are close to lights and the cells with the smallest intensities are shadowed areas close to chairs, plants and walls.

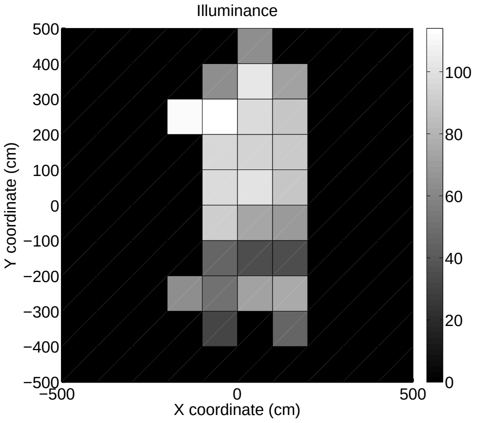


Figure 4: Illumincance in global coordinates.

Fig. 5 presents a grid map of compass headings with a bidirectional arrow, where red points to the north and white to the south. Since compass headings are disturbed indoors by electric cables and metal structures, the values of the compass headings give us spatial information about the magnetic field in the environment. However, in the experiments, the magnetic field of the measured environment was parallel, which gave us small spatial variations in compass heading.

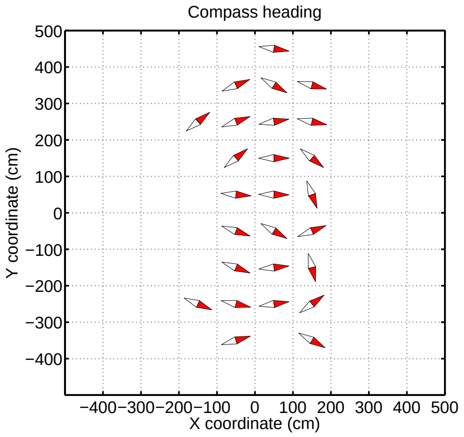


Figure 5: Compass heading in global coordinates.

The experiments demonstrated distributed sensing in a group of robots to produce distributions of illuminance and a magnetic field. All the measurements were tied to the co-ordinates of the leading robot. Since the ground truth positions were missing, positioning errors in the global co-ordinates of the sensing robots were affected by the odometer error of the leading robot and the errors of the infrared location system. In addition, the grid size in the experiments was 1x1 metres, which gave only a coarse picture of the true distributions. However, these preliminary experiments demonstrated the feasibility of using the infrared location system in distributed autonomous sensing systems.

4 CONCLUSION AND FUTURE WORKS

In this paper we presented a distributed multi-robot sensing system that uses an infrared location system. The infrared location system estimates poses in a multi-robot system, enabling the robots to maintain a given formation while sensing the environment. In addition, knowing their poses enables the robots to place their measurements on the same map.

We conducted an experiment where a group of three robots moved and measured spatial information in a right-angle triangular formation. Leader-follower formation control used pose estimates and P controllers to control the rotational and translational speeds of the following robots. In the experiments we measured spatial distributions of illuminance and a magnetic field, which gave us information about shadowing objects, metal structures and electric cables. In addition, since the information is spatially distributed, it can be used in mapping and localization applications.

The main contribution of the research was the construction and validation of a distributed multi-robot sensing system for mobile sensing applications. Future research will focus on developing methods for multi-robot exploration utilising spatial information.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the support of the Academy of Finland.

REFERENCES

Brooks, R. R. and Iyengar, S. S. (1998). *Multi-sensor fusion: fundamentals and applications with software*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Cai, A., Fukuda, T., Arai, F., and Ishihara, H. (1996). Cooperative path planning and navigation based on distributed sensing. In *IEEE International Conference on Robotics and Automation (ICRA-96)*, pages 2079 – 2084, Minneapolis.

Catterall, E., Laerhoven, K. V., and Strohbach, M. (2003). Self-organization in ad hoc sensor networks: an empirical study. In *ICAL 2003: Proceedings of the eighth international conference on Artificial life*, pages 260–263, Cambridge, MA, USA. MIT Press.

Grabowski, R., Navarro-Serment, L., Paredis, C., and Khosla, P. (2000). Heterogeneous teams of modular robots for mapping and exploration. *Autonomous Robots*, 8(3):293–308.

Haverinen, J., Parpala, M., and Röning, J. (2005). A miniature mobile robot with a color stereo camera system for swarm robotics research. In *IEEE International Conference on Robotics and Automation (ICRA2005)*, pages 2494–2497, Barcelona, Spain.

Howard, A., Mataric, M., and Sukhatme, G. (2003). Putting the 'i' in team: an ego-centric approach to cooperative localization. In *International Conference on Robotics and Automation*.

Kemppainen, A., Haverinen, J., and Röning, J. (2006). An infrared location system for relative pose estimation of robots. In *16-th CISM-IFToMM Syposium of Robot Design, Dynamics, and Control (ROMANSY 2006)*, pages 379–386, Warsaw, Poland.

Montesano, L., Gaspar, J., Santos-Victor, J., and Montano, L. (2005). Fusing vision-based bearing measurements and motion to localize pairs of robots. In *International Conference on Robotics and Automation*.

Montesano, L., Montano, L., and Burgard, W. (2004). Relative localization for pairs of robots based on unidentifiable moving features. In *International Conference on Intelligent Robots and Systems*.

Moors, M., Schneider, F. E., and Wildermuth, D. (2003). Relative position estimation in a group of robot. In *International Conference on Climbing and Walking Robots*.

Pagello, E., D'Angelo, A., and Menegatti, E. (2006). Cooperation issues and distributed sensing for multirobot systems. *Proceedings of the IEEE*, 94(7):1370–1383.

Schneider, F. and Wildermuth, D. (2004). Using an extended kalman filter for relative localisation in a moving robot formation. In *International Workshop on Robot Motion and Control*.

Shoval, S. and Borenstein, J. (2001). Measuring the relative position and orientation between two mobile robots with binaural sonar. In *International Topical Meeting on Robotics and Remote Systems*.

Spletzer, J., Das, A., Fierro, R., Taylor, C., Humar, V., and Ostrowski, J. (2001). Cooperative localization and control for multi-robot manipulation. In *Conference on Intelligent Robots and Systems*.

Sujan, V. A., Dubowsky, S., Huntsberger, T. L., Aghazarian, H., Cheng, Y., and Schenker, P. S. (2004). An architecture for distributed environment sensing with application to robotic cliff exploration. *Auton. Robots*, 16(3):287–311.

AUTOMATIC VISION-BASED MONITORING OF THE SPACECRAFT ATV RENDEZVOUS / SEPARATIONS WITH THE INTERNATIONAL SPACE STATION

A. A. Boguslavsky, V. V. Sazonov, S. M. Sokolov
Keldysh Institute for Applied Mathematics, Miusskaya Sq., 4, Moscow, Russia
{boguslav, sazonov, sokolsm}@keldysh.ru

A. I. Smirnov, K. U. Saigiraev
RSC ENERGIA, Korolev, Russia
{smirnov, cxy}@scsc.ru

Keywords: Algorithms of determining the motion parameters, space rendezvous / separations processes, accuracy, real time vision system.

Abstract: The system which allows automating the visual monitoring of the spacecraft ATV rendezvous / separations with the international space station is being considered. The initial data for this complex is the video signal received from the TV-camera, mounted on the station board. The offered algorithms of this video signal processing in real time allow restoring the basic characteristics of the spacecraft motion with respect to the international space station. The results of the experiments with the described software and real and virtual video data about the docking approach of the spacecraft with the International Space Station are being presented. The accuracy of the estimation of the motion characteristics and perspectives of the use of the package are being discussed.

1 INTRODUCTION

One of the most difficult and crucial stages in managing the flights of space vehicles is the process of their docking approach. The price of a failure at performing of this process is extremely high. The safety of crew, station and space vehicles also in many respects depends on a success of its performance.

The radio engineering means of the docking approach, which within many years have been used at docking of the Russian space vehicles, are very expensive and do not allow to supply docking to not cooperated station.

As reserve methods of docking approach monitoring the manual methods are applied, for which in quality viewfinders the optical and television means are used. For docking approach of the pilotless cargo transport vehicle “Progress” to the orbital station “Mir” a teleoperation mode of manual control (TORU) was repeatedly used, at which the realization the crew of the station, having received the TV image of the station target from a spacecraft, carried out the manual docking approach.

At the center of the flight management the control of objects relative motion parameters (range, speed, angular deviations) should also be carried out. The semi-automatic TV methods of the monitoring which are being used till now, do not satisfy the modern requirements anymore. Recently appeared means of both the methods of the visual data acquisition and processing provide an opportunity of the successful task decision of a complete automatic determination and control of space vehicles relative motion parameters.

Flights of the ATV have not begun yet. Therefore images of the ATV in ground-based conditions and simulated space images are available now. But already there are some papers discussing possible approaches to the decision of similar images analysis problems (Chi-Chang, H.J. and McClamroch, N.H., 1993; Casonato, G. and Palmerini, G.B., 2004). The described approach is based on experiments with similar system for the “Progress” spaceship (Boguslavsky et al., 2004).

The system for ATV has a number of differences from the system used for "Progress". First of all, this various arrangement of the TV-camera and special target for the monitoring of rendezvous / separations processes. In the ATV case the TV-camera settles down on ISS, and by the transport vehicle the special targets (basic and reserve) are established. Besides that is very important for visual system, the images of objects for tracking in a field of view have changed.

In view of operation experience on system Progress - ISS, in system ATV - ISS essential changes to the interface of the control panel (see Part 4) were made.

The main stages of the visual data acquisition and processing in the complex are realized mostly in the same way as the actions performed by an operator.

In the presented field of vision:

1. An object of observation is marked out (depending on the range of observation, it can be the vehicle as a whole or a docking assembly or a special target);

2. Specific features of an object under observation defined the patterns of which are well recognizable in the picture (measuring subsystem).

3. Based on the data received:

The position of these elements relative to the image coordinate system is defined;

Parameters characterizing the vehicle and station relative position and motion are calculated (calculation subsystem).

In addition to operator actions, the complex calculates and displays the parameters characterizing the approach/departure process in a suitable for analysis form.

This research work was partially financed by the grants of the RFBR ## 05-01-00885, 06-08-01151.

2 MEASURING SUBSYSTEM

The purpose of this subsystem is the extraction of the objects of interest from the images and performance of measurements of the points' coordinates and sizes of these objects. To achieve this purpose it is necessary to solve four tasks:

1) Extraction of the region of interest (ROI) position on the current image.

2) Preprocessing of the visual data inside the ROI.

3) Extraction (recognition) of the objects of interest.

4) Performing the measurements of the sizes and coordinates of the recognized objects.

All the listed above operations should be performed in real time. The real time scale is determined by the television signal frame rate. The other significant requirement is that in the considered software complex it is necessary to perform updating of the spacecraft motion parameters with a frequency of no less than 1 time per second.

For reliability growth of the objects of interest the extraction from the images of the following features are provided:

- Automatic adjustments of the brightness and contrast of the received images for the optimal objects of interest recognition.

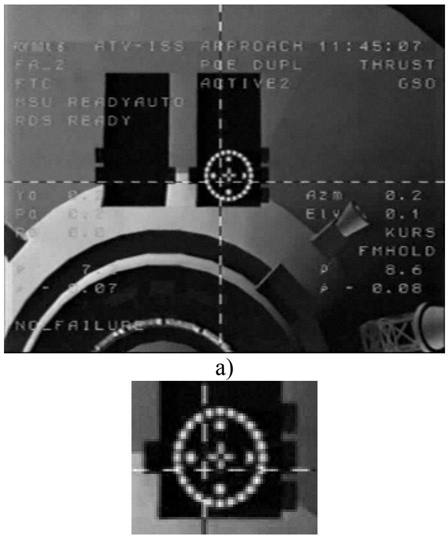
- Use of the objects of interest features of the several types. Such features duplication (or even triplication) raises reliability of the image processing when not all features are well visible (in the task 3).

- Self-checking of the image processing results on a basis of the a priori information about the observed scenes structure (in the tasks 1-4).

The ways of performing the calculation of the ROI position on the current image are (the task 1):

1a) Calculation of the ROI coordinates (figure 1) on the basis of the external information (for example, with taking into account a scene structure or by the operator selection).

1b) Calculation of the ROI coordinates and sizes on the basis of the information received from the previous images processing.



a)

b)

Figure 1: An example of identification of a region of interest in the TV camera field of view (1.a) way): a) – total field of view, b) – a region of interest.

The second (preprocessing) task is solved on the basis of the histogram analysis. This histogram describes a brightness distribution inside the ROI. The brightness histogram should allow the reliable transition to the binary image. In the considered task the brightness histogram is approximately bimodal. The desired histogram shape is provided by the automatic brightness and contrast control of the image source device.

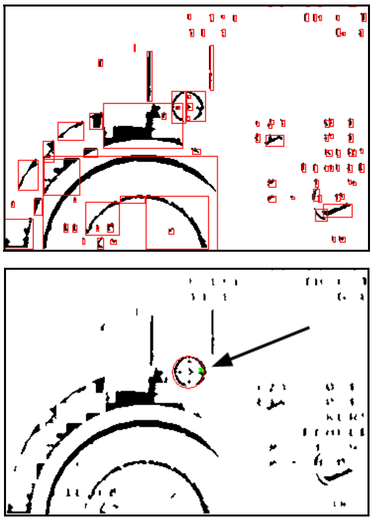
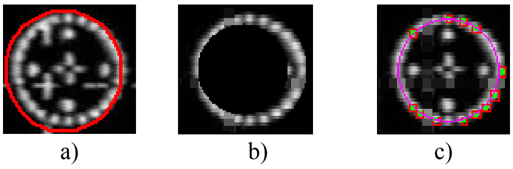


Figure 2: Coarse allocation of a target from all candidates on a basis of the a priory information: the relative sizes of labels and target plate and their relative location.

At the third processing stage (the task 3) the extraction of the objects of interest is performed. These objects are the spaceship contour, the docking unit disk, the target cross and the target disk with small labels. The main features are the spaceship contour, the cross and the target labels. These features are considered the main structured elements of the recognized scene and used for measurement. At features extraction both edge-based (Canny, 1986; Mikolajczyk et al., 2003) and region-based methods are used (Sonka and Hlavac, 1999).

With accordance to such elements set the processing of the current field image is being performed. For example, on distances less than 15 meters are made detection of a target plate, its labels and a cross. During such processing (figures 1, 2, 3) the results of the previous measurements can be used for estimation of the small area of interest position (figure 1, below). If the previous results are not present (for example, immediately after system start), as area of interest is processed the full field image. During processing among candidates for a target image of a target it selects the most probable candidate (figure 2). For its verification the detection of labels and of radius of a target (figure 3) is performed. After that the detection of a cross of a target is performed.



a) b) c)

Figure 3: An example of allocation of an image of a target on the basis of the coarse coordinates of a target plate and a priory of the information on an arrangement of labels of a target. a) - definition of area of interest on a basis a priory of the information and found coarse location of the plate; b) area of interest for search of labels of a target; c) - result of allocation of labels of a target.

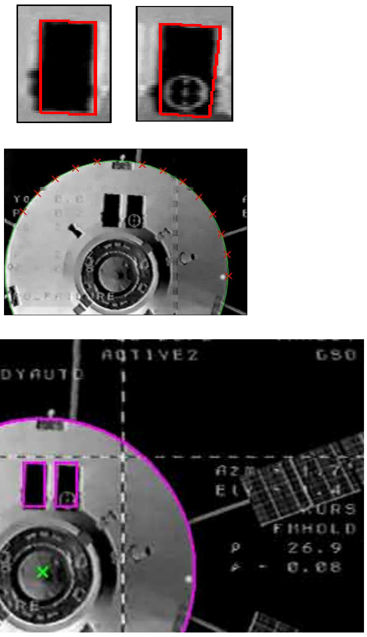


Figure 4: An example of allocation of edge of target boxes and external contour of spaceship (for distance >10 meters). Top-down: extracted edges of the left and the right boxes; contour of the spaceship; results of the frame processing (with the marked docking unit image centre).

On distances more than 15 m a label and a cross of a target reliably are not visible. Therefore the detection of an external contour of the ship and a target boxes (figure 4) is realized. Boxes of a target allow to confirm a hypothesis about the detection of the ship and to specify its roll position.

The fourth operation (performing of the measurement) is very straightforward. But for this operation it is necessary to recognize the reliable objects results from the previous processing stages.

3 CALCULATION PART

Preprocessing of a frame (more exactly, a half-frame) gives the following information:

T – reference time of a frame (in seconds); X_C, Y_C – real values coordinates of the center of the cross on the target; N_1 –number of couples of points on the horizontal crossbar of the cross; X_i, Y_i $(i = 1,2,\dots,N_1)$ – integer coordinates of points on the top of the horizontal crossbar of the cross; $X_i, Y_i' (i = 1,2,\dots,N_1)$ – coordinates of points on the bottom of the horizontal crossbar of the cross; N_2 – number of couples of points on the vertical crossbar of the cross; U_i, V_i $(i = 1,2,\dots,N_2)$ – coordinates of points on the left side of vertical crossbar of the cross; U_i', V_i $(i = 1,2,\dots,N_2)$ – coordinates of points on the right side of vertical crossbar of the cross; X_O, Y_O, R – coordinates of the center of the circle on the target and its radius; N_3, A_i, B_i $(i = 1,2,\dots,N_3)$ – number of points on the circle and their coordinates; X_S, Y_S, R_S – coordinates of the center of the circle, which is the station outline, and its radius (real numbers).

Here all coordinates are expressed in pixels. Successful preprocessing a frame always gives the values of X_C, Y_C and X_O, Y_O, R, but if there is an opportunity, when appropriate $N_k > 0$, those quantities are determined once again by original information. Bellow we describe the calculation of those quantities in the case when $N_1 > 0, N_2 > 0, N_3 > 0$ and $R_S = 0$. If R_S differs from zero, the data are used in other way (see bellow).

Determining the coordinates of the cross center. We change the data

$$\frac{Y_i + Y_i'}{2} \to Y_i, \quad \frac{U_i + U_i'}{2} \to U_i \quad (i = 1,2,\dots)$$

and get coordinates of two sequences of points, which lie in centerlines of horizontal and vertical crossbars of the cross. Those centerlines have equations $ax - y = c_1$ (horizontal) and $x + ay = c_2$ (vertical). Here a, c_1 and c_2 are coefficients. The form of the equations takes into account the orthogonality of these lines. The coefficients are determined by minimization of the quadratic form

$$\sum_{i=1}^{N_1} (aX_i - Y_i - c_1)^2 + \sum_{j=1}^{N_2} (U_j + aV_j + c_2)^2$$

on a, c_1, c_2, i.e. by solving the linear least squares problem. The coordinates of the cross center are

$$X_C^* = \frac{ac_1 + c_2}{1 + a^2}, \quad Y_C^* = \frac{c_1 - ac_2}{1 + a^2}.$$

Determining the radius and the center of the target circle is realized in two stages. In the first stage, we obtain preliminary estimations of these quantities based on elementary geometry. In the second stage, we solve the least squares problem minimizing the expression

$$\Phi_2 = \sum_{i=1}^{N_3} \left[\sqrt{(A_i - X_O)^2 + (B_i - Y_O)^2} - R \right]^2$$

on X_O, Y_O, R by Gauss-Newton method (Bard, 1974). Let its solution be X_O^*, Y_O^*, R^*. As a rule, $|X_O - X_O^*|$ and $|Y_O - Y_O^*|$ do not exceed 1.5 pixels. Below for simplicity of notations, we will not use an asterisk in designations of recomputed parameters.

3.1 Basic Geometrical Ratios

We use the right Cartesian coordinate system $Cy_1y_2y_3$, which is connected with the target. The point C is the center of the target circle, the axis Cy_3 is directed perpendicularly to the circle plane away from station, i.e. is parallel a longitudinal axis of the Service module, the axis Cy_2^+ intersects a longitudinal axis of the docking device on the Service Module. Also, we use right Cartesian coordinate system $Sx_1x_2x_3$ connected with the TV camera on the spacecraft. The plane Sx_1x_2 is an image plane of the camera, the axis Sx_3 is a camera optical axis and directed on movement of the spacecraft, the axis Sx_2^- intersects an axis of the docking device of the spacecraft. Let $\| a_{ij} \|_{i,j=1}^3$ be the transition matrix from the system $Sx_1x_2x_3$ to the system $Cy_1y_2y_3$. The transition formulas are

$$y_i = d_i + \sum_{j=1}^{3} a_{ij} x_j \ (i = 1,2,3),$$

$$x_j = \sum (y_i - d_i) a_{ij} \ (j = 1,2,3).$$

Here d_1, d_2, d_3 are coordinates of the point S in the system $Cy_1y_2y_3$.

The matrix characterizes ideal docking $\| a_{ij} \| = \operatorname{diag}(1, -1, -1)$. In actual docking the transition matrix is

$$\| a_{ij} \| = \begin{Vmatrix} 1 & -\varphi_3 & \varphi_2 \\ -\varphi_3 & -1 & \varphi_1 \\ \varphi_2 & -\varphi_1 & -1 \end{Vmatrix}$$

where $\varphi, \varphi_2, \varphi_3$ are components of the vector of an infinitesimal rotation of the system $Sx_1x_2x_3$ with respect to its attitude in ideal docking. We suppose deviations from ideal docking are small.

If any point has in the system $Sx_1x_2x_3$ the coordinates (x_1, x_2, x_3), its image has in the image plane the coordinates $\xi_1 = \frac{fx_1}{x_3}$, $\xi_2 = \frac{fx_2}{x_3}$.

Here f is focal length of the camera. The coordinates ξ_1 and ξ_2, expressed in pixels, were meant in the above description of processing a single video frame. Let coordinates of the same point in the system $Cy_1y_2y_3$ be (y_1, y_2, y_3). Then

$$\xi_i = f\frac{(y_1 - d_1)a_{1i} + (y_2 - d_2)a_{2i} + (y_3 - d_3)a_{3i}}{(y_1 - d_1)a_{13} + (y_2 - d_2)a_{23} + (y_3 - d_3)a_{33}}.$$

The coordinates of the center of the target circle the C in the system $Cy_1y_2y_3$ are $(0,0,0)$, therefore

$$X_O = f\frac{d_1 - d_2\varphi_3 + d_3\varphi_2}{d_1\varphi_2 + d_2\varphi_1 - d_3}, \quad Y_O = -f\frac{d_1\varphi_3 + d_2 + d_3\varphi_1}{d_1\varphi_2 + d_2\varphi_1 - d_3}.$$

In docking $|d_1| << d_3$, $|d_2| << d_3$, so it is possible to use the simplified expressions

$$X_O = -\frac{fd_1}{d_3} - f\varphi_2, \quad Y_O = \frac{fd_2}{d_3} + f\varphi_1.$$

The center of the cross in the system $Cy_1y_2y_3$ has the coordinates $(0,0,b)$. In this case under the similar simplification, we have

$$X_C = -\frac{fd_1}{d_3 - b} - f\varphi_2, \quad Y_C = \frac{fd_2}{d_3 - b} + f\varphi_1.$$

So $X_C - X_O = -\frac{fbd_1}{d_3(d_3 - b)}, Y_C - Y_O = \frac{fbd_2}{d_3(d_3 - b)}$.

The radius r of the target circle and radius R of its image in the image plane are connected by the ratio $R = fr / d_3$.

The last three ratios allow to express d_3, d_1 and d_2 through R, $X_C - X_O$ and $Y_C - Y_O$. Then it is possible to find φ_1 and φ_2, having solved concerning these angles the expressions for X_O, Y_O or X_C, Y_C. As to the angle φ_3, the approximate ratio $\varphi_3 = a$ takes place within the framework of our consideration.

The processing a frame is considered to be successful, if the quantities d_i, φ_i $(i = 1, 2, 3)$ were estimated. As a result of successful processing a sequence of frames, it is possible to determine spacecraft motion with respect to the station. The successfully processed frames are used only for motion determination.

3.2 Algorithm for Determination of the Spacecraft Motion

The spacecraft motion is determined in real time as a result of step-by-step processing of a sequence of TV images of the target. The data are processed by separate portions.

In each portion is processed in two stages. The first stage consists in determining the motion of the spacecraft center of mass; the second stage consists in determining the spacecraft attitude motion. Mathematical model of motion is expressed by formulas

$$d_1 = z_1 + z_2t, \quad d_2 = z_3 + z_4t, \quad d_3 = z_5 + z_6t,$$
$$\varphi_1 = v_1 + v_2t, \quad \varphi_2 = v_3 + v_4t, \quad \varphi_3 = v_5 + v_6t.$$

Here t is time counted off the beginning of processing the frame sequence, z_i and v_j are constant coefficients. The ratios written out have the obvious kinematical sense. We denote the values of the model coefficients, obtained by processing the portion of the data with number n, by $z_i^{(n)}, v_j^{(n)}$ and the functions $d_i(t)$, $\varphi_i(t)$, corresponding to those coefficients, by $D_i^{(n)}(t)$, $\Phi_i^{(n)}(t)$.

Determining the motion consists in follows. Let there be a sequence of successfully processed frames, which correspond to the instants $t_1 < t_2 < t_3 < \ldots$. The frame with number k corresponds to the instant t_k. Values of the quantities X_C, Y_C, a, X_O, Y_O, R, which were found by processing this frame, are $X_C^{(k)}$, $Y_C^{(k)}$, etc. These values with indexes $k = 1, 2, \ldots, K_1$ form the first data portion, the value with indexes $k = K_1 + 1, K_1 + 2, \ldots, K_2$ – the second one, with indexes $k = K_{n-1} + 1, K_{n-1} + 2, \ldots, K_n$ – the n-th portion.

The first data portion is processed by a usual method of the least squares. The first stage consists in minimization of the functional

$$\Psi_1(z) = \sum_{k=1}^{K_1} A_k,$$

$$A_k = w_1\left[X_C^{(k)} - X_O^{(k)} + \frac{fbd_1^{(k)}}{d_3^{(k)}[d_3^{(k)} - b]}\right]^2 +$$

$$w_2\left[Y_C^{(k)} - Y_O^{(k)} - \frac{fbd_2^{(k)}}{d_3^{(k)}[d_3^{(k)} - b]}\right]^2 + w_3\left[R^{(k)} - \frac{fr}{d_3^{(k)}}\right]^2$$

Here $d_i^{(k)} = d_i(t_k)$, $z = (z_1, z_2, \ldots, z_6)^T$ is a vector of the coefficients, which specify the functions $d_i(t)$, w_i, is positive numbers (weights). The minimization is carried out by Gauss -Newton method (Bard, 1974). The estimation $z^{(1)}$ of z and the covariance matrix P_1 of this estimation are defined by the formulas

$$z^{(1)} = \left[z_1^{(1)}, z_2^{(1)}, \ldots, z_6^{(1)}\right]^T = \arg\min \Psi_1(z),$$

$$P_1 = \sigma^2 B_1^{-1}, \qquad \sigma^2 = \frac{\Psi_1\left[z^{(1)}\right]}{3K_1 - 6}.$$

Here B_1 is the matrix of the system of the normal equations arising at minimization of Ψ_1. The matrix is calculated at the point $z^{(1)}$.

At the second stage, the quantities

$$\alpha_1^{(k)} = \frac{1}{f}\left[Y_O^{(k)} - \frac{fD_2^{(1)}(t_k)}{D_3^{(1)}(t_k)}\right], \alpha_2^{(k)} = -\frac{1}{f}\left[X_O^{(k)} + \frac{fD_1^{(1)}(t_k)}{D_3^{(1)}(t_k)}\right]$$

are calculated and three similar linear regression problems

$$\alpha_1^{(k)} \approx v_1 + v_2 t_k, \qquad \alpha_2^{(k)} \approx v_3 + v_4 t_k,$$

$$a^{(k)} \approx v_5 + v_6 t_k \qquad (k = 1, 2, \ldots, K_1)$$

are solved using the standard least squares method (Seber, 1977). We content ourselves with description of estimating the couple of parameters v_1, v_2. We unite them in the vector $v = (v_1, v_2)^T$. The estimations $v_1^{(1)}$ and $v_2^{(1)}$ provide the minimum to the quadratic form

$$F_1(v) = \sum_{k=1}^{K_1}\left[\alpha_1^{(k)} - v_1 - v_2 t_k\right]^2.$$

Let Q_1 be the matrix of this form. Then the covariance matrix of the vector $v^{(1)} = [v_1^{(1)}, v_2^{(1)}]^T$ is $Q_1^{-1} F_1[v^{(1)}]/(K_1 - 2)$.

The second data portion is carried out as follows. At the first stage, the functional

$$\Psi_2(z) = \left[z - z^{(1)}\right]^T C_2 \left[z - z^{(1)}\right] + \sum_{k=K_1+1}^{K_2} A_k$$

is minimized. Here $C_2 = qB_1$, q is a parameter, $0 \le q \le 1$. The estimation of z and its covariance matrix have the form

$$z^{(2)} = \arg\min \Psi_2(z), \qquad P_2 = \sigma^2 B_2^{-1},$$

$$\sigma^2 = \frac{\Psi_2\left[z^{(2)}\right]}{3(K_2 - K_1) - 6},$$

where B_2 is the matrix of the system of the normal equations, which arise at minimization of Ψ_2, calculated at the point $z^{(2)}$.

At the second stage, the quantities $\alpha_1^{(k)}$ and $\alpha_2^{(k)}$ (see above) are calculated and the estimation of the coefficients $v_j^{(2)}$ are found. The estimation $v^{(2)}$ provides the minimum to the quadratic form

$$F_2(v) = q'\left[v - v^{(1)}\right]^T Q_1 \left[v - v^{(1)}\right] + \sum_{k=K_1+1}^{K_2}\left[\alpha_1^{(k)} - v_1 - v_2 t_k\right]^2.$$

Here q' is a parameter, $0 \le q' \le 1$. Let Q_2 be the matrix of this form. The covariance matrix of the estimation $v^{(2)}$ is $Q_2^{-1} F_2[v^{(2)}]/(K_2 - K_1 - 2)$.

The third and subsequent data portions are processed analogously to the second one. The formulas for processing the portion with number n are obtained from the formulas for processing the second portion by replacement of the indexes expressed the portion number: $1 \to n-1$, $2 \to n$.

Our choice of C_n and $\Psi_n(z)$ means that the covariance matrix of errors in $X_C - X_O$, $Y_C - Y_O$ and R is equal to $\mathrm{diag}(w_1^{-1}, w_2^{-1}, w_3^{-1})$.

It is easy to see that $C_n < B_{n-1}$, i.e. the matrix $B_{n-1} - C_n$ is positive definite. The introduction of the matrix G_n provides diminution of influence of the estimation $z^{(n-1)}$ on the estimation $z^{(n)}$. Unfortunately, the matrix G_n is unknown. In such situation, it is natural to take $C_n = qB_{n-1}$. One has $C_n < B_{n-1}$ if $q < 1$. The described choice of C_n means, that procession of the n-th data portion takes into account the data of the previous portions. The data of the n-th portion are taken in processing with the weight 1, the $(n-1)$-th portion is attributed the weight q, the $(n-2)$-th portion has the weight q^2, etc.

The results of processing the n-th data portion are represented by numbers $D_i^{(n)}(t_{K_n})$, $\Phi_i^{(n)}(t_{K_n})$ $(i = 1, 2, 3; n = 1, 2, \ldots)$. We calculate also the quantities $\rho = \sqrt{d_1^2 + d_2^2 + d_3^2}$, $u = \dfrac{d\rho}{dt}$,

$$\alpha = \arctan\frac{d_2}{\sqrt{d_1^2 + d_3^2}}, \qquad \beta = \arctan\frac{d_1}{d_3}.$$

The angle α is called a passive pitch angle, the angle β is a passive yaw angle. If docking is close to ideal (considered case), then $|d_1| << d_3$, $|d_2| << d_3$ and $\alpha = d_2 / d_3$, $\beta = d_1 / d_3$.

The angle φ_1 is called an active pitch angle, φ_2 is an active yaw angle, φ_3 is an active roll angle. We remind these angles have small absolute values.

Characteristics of accuracy of the motion parameter estimations are calculated within the framework of the least squares method. For example, we defined above the covariance matrix P_n of the estimation $z^{(n)}$. In terms of this matrix the covariance matrix $C_w(t)$ of the vector $w(t) = (z_1 + z_2 t, z_2, z_3 + z_4 t, z_4, \ldots, v_5 + v_6 t, v_6)^T \in R^{12}$ is calculated by formulas

$$C_w = \frac{\partial w}{\partial z} P_n \left(\frac{\partial w}{\partial z} \right)^T, \quad \frac{\partial w}{\partial z} = \mathrm{diag}\,(\mathrm{U, U, U}), \quad U = \begin{Vmatrix} 1 & t \\ 0 & 0 \end{Vmatrix}.$$

These formulas concern to the motion which was found by processing the n-th of a portion of the data.

Knowing $C_w(t)$, it is possible to calculate the standard deviations $\sigma_\rho(t)$, $\sigma_u(t)$, $\sigma_\alpha(t)$ and $\sigma_\beta(t)$ of the quantities $\rho(t)$, $u(t)$, $\alpha(t)$ and $\beta(t)$. The standard deviation $\sigma_\rho(t)$ has the form

$$\sigma_\rho = \frac{\partial \rho}{\partial w} C_w \left(\frac{\partial \rho}{\partial w} \right)^T, \quad \frac{\partial \rho}{\partial w} = \left(\frac{d_1}{\rho}, 0, \frac{d_2}{\rho}, 0, \frac{d_3}{\rho} \right)^T.$$

The similar formulas define the others standard deviations. The values of ρ, σ_ρ, u, σ_u, etc., referring to the last instant of the processed data portion, are written on the computer display.

4 EXAMPLES

The experiments with the described complex were carried out on various samples of the initial data. The adjustment of system on the images of the transport ship ATV was carried out on photos of a breadboard model of the ship in a hangar and on rendered samples of the 3D ship model trajectories. Other initial data were real video received from a board of the ship "Progress". These data were used for an estimation of reliability and correctness of the detection of real objects, and also for the verification of algorithms of relative movement parameters calculation.

Figures 5, 6 give examples of the operation of the described algorithm estimating the spacecraft motion. Figure 5 contains the plots of the functions $\rho(t)$, $u(t)$, $\alpha(t)$ and $\beta(t)$ and $\varphi_i(t)$ $(i = 1,2,3)$. On $\rho(t)$ and $u(t)$ plots the scale on a vertical axis differs on 2 order.

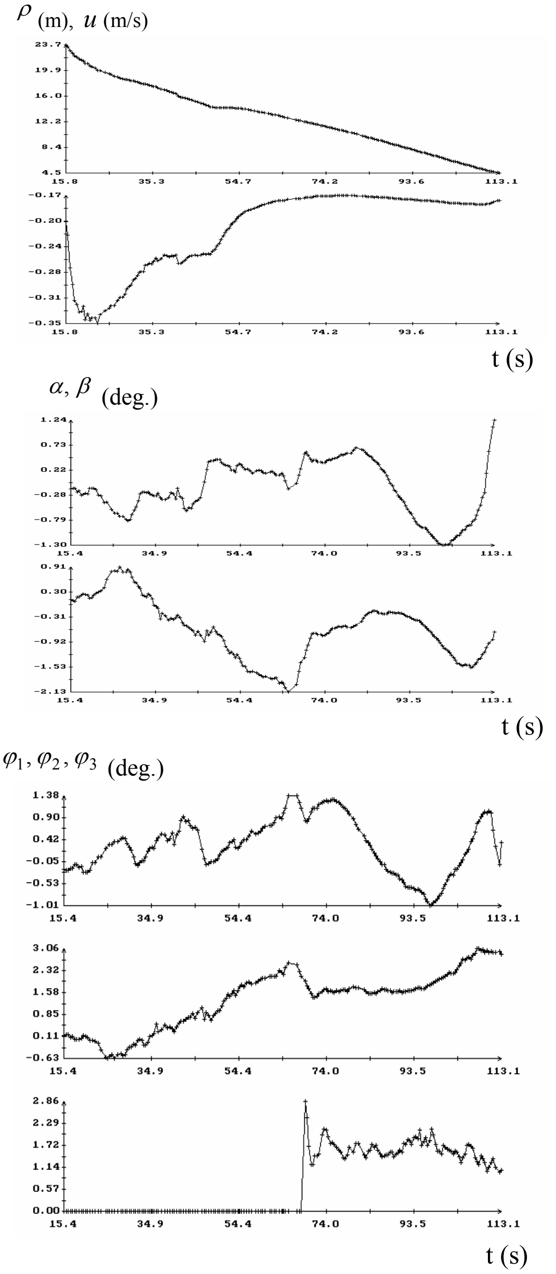


Figure 5: Results of the determination of the spacecraft motion in docking approach.

Figure 6 presents the plots of the standard deviations $\sigma_\rho(t)$, $\sigma_u(t)$, $\sigma_\alpha(t)$, $\sigma_\beta(t)$. The values of all these functions were calculated at the last instants of processed data portions. These values were shown by marks. Each portion contained 10 instants with measurements: $K_n - K_{n-1} = 10$. For clearness, the markers were connected by segments of straight lines, therefore presented plots are broken lines. Only the vertexes of these broken lines are significant. Their links are only interpolation, which is used for visualization and not always exact. As it is shown in figure 6, the spacecraft motion on the final stage of docking was defined rather accurately.

Figure 7 shows an example of the basic screen of the main program of a complex.

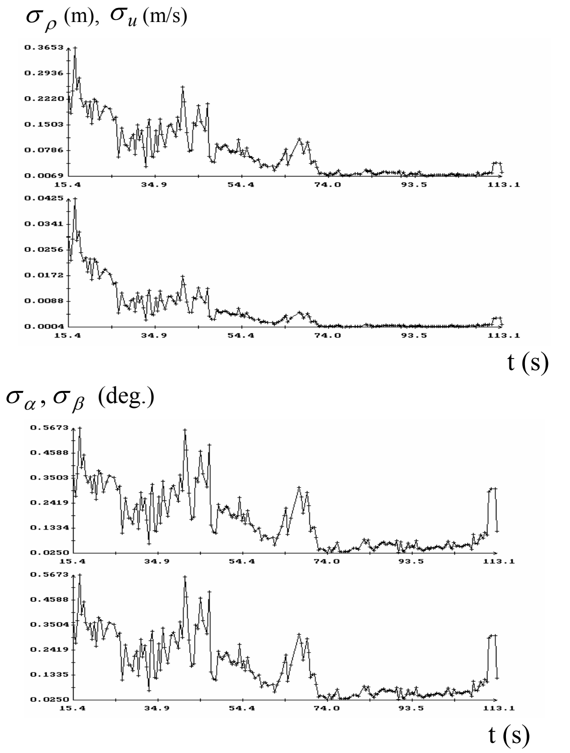
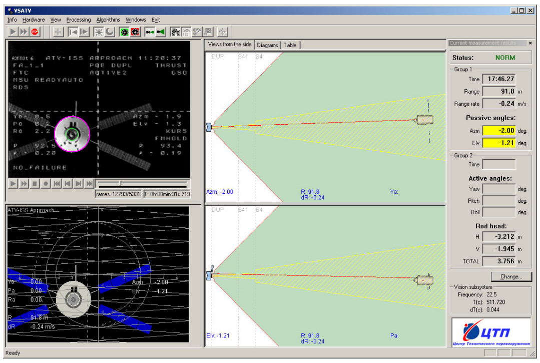


Figure 6: Accuracy estimations for the motion presented on Figure 5.

5 CONCLUSION

The described complex is being prepared for the use as a means allowing the ground operators to receive the information on the motion parameters of the spacecraft docking to ISS in real time.

The most essential part of this information is transferred to the Earth (and was always transferred) on the telemetering channel. It is also displayed on the monitor. However this so-called regular information concerns the current moment and without an additional processing can't give a complete picture of the process. Such an additional processing is much more complicated than the organizational point of view and more expensive than processing the video image. It is necessary to note, that the estimation of kinematical parameters of the moving objects on the video signal, becomes now the most accessible and universal instrument of solving such kind of problems in situations, when the price of a failure is rather insignificant.



Status: NORM

Group 1
Time 17:46.27
Range 91.8 m
Range rate -0.24 m/s

Passive angles:
Azm -2.00 deg.
Elv -1.21 deg.

Group 2
Time

Active angles:
Yaw deg.
Pitch deg.
Roll deg.

Rod head:
H -3.212 m
V -1.945 m
TOTAL 3.756 m

Vision subsystem
Frequency: 22.5
T(c): 511.720
dT(c): 0.044

Figure 7: Main screen of the VSATV program.

REFERENCES

Boguslavsky A.A., Sazonov V.V., Smirnov A.I., Sokolov S.M., Saigiraev K.S., 2004. Automatic Vision-based Monitoring of the Spacecraft Docking Approach with the International Space Station. In *Proc. of the First International Conference on Informatics in Control, Automation and Robotics (ICINCO 2004)*, Setúbal, Portugal, August 25-28, 2004, Vol. 2, p.79-86.

Chi-Chang, H.J., McClamroch, N.H., 1993. Automatic spacecraft docking using computer vision-based guidance and control techniques. In *Journal of Guidance, Control, and Dynamics*, vol.16, no.2: pp. 281-288.

Casonato, G., Palmerini, G.B., 2004. Visual techniques applied to the ATV/ISS rendezvous monitoring. In *Proc. of the IEEE Aerospace Conference*, vol. 1, pp. 619-625.

Canny, J. 1986. A computational approach to edge detection. In *IEEE Trans. Pattern Anal. and Machine Intelligence,* 8(6): pp. 679-698.

Mikolajczyk, K., Zisserman, A., Schmid, C., 2003. Shape recognition with edge-based features. In *Proc. of the 14th British Machine Vision Conference (BMVC'2003)*, BMVA Press.

Sonka, M., Hlavac, V., Boyle, R. 1999. *Image Processing, Analysis and Machine Vision.* MA: PWS-Kent.

Bard, Y., 1974. *Nonlinear parameter estimation.* Academic Press, New York.

Seber, G.A.F., 1977. *Linear regression analysis.* John Wiley and Sons, New York.

NEW APPROACH TO GET AUTONOMOUS AND FAST ROBOT LEARNING PROCESSES

R. Iglesias[1], M. Rodríguez[1], C.V. Regueiro[2], J. Correa[1], Pablo Quintía[2] and S. Barro[1]
[1]*Electronics and Computer Science, University of Santiago de Compostela, Spain*
[2]*Dept. of Electronic and Systems, University of Coruña, Spain*
elberto@usc.es

Keywords: Reinforcement learning, mobile robotics, robot control, autonomous agents, genetic algorithms.

Abstract: Research on robot techniques that are fast, user-friendly, and require little application-specific knowledge by the user, is more and more encouraged in a society where the demand of home-care or domestic-service robots is increasing continuously. In this context we propose a methodology which is able to achieve fast convergences towards good robot-control policies, and reduce the random explorations the robot needs to carry out in order to find the solutions. The performance of our approach is due to the mutual influence that three different elements exert on each other: *reinforcement learning*, *genetic algorithms*, and a *dynamic representation of the environment around the robot*.

The performance of our proposal is shown through its application to solve two common tasks in mobile robotics.

1 INTRODUCTION

Reinforcement learning seems to be 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 desired set of actions it should have carried out.

The maximization of the reinforcement allows the robot to learn a utility function of states and actions called *Q-function*, which reflects the consequences of executing every possible action in each state – we will assume that the robot interacts with the environment at discrete time steps and it is able to translate the different situations that it may detect through its sensors into a finite number of states, S –. Table 1 shows one of the many reinforcement learning algorithms that can be applied to teach a robot: the *truncated temporal differences* algorithm, $TTD(\lambda,m)$ (Cichosz, 1997). Basically the robot begins with an initial set of random negative Q-values: $Q(s,a) \leq 0$, $\forall s,a$, and then it performs a stochastical exploration of its environment. While the robot is moving around, it keeps track of the *m* last executed actions so that their corresponding Q-values will be decreased or increased depending on whether the robot receives or not negative reinforcements. The parameters γ, λ, and β (that appear in table 1), determine how much the Q-values are changed for every positive or negative reinforcement the robot receives. As the learning process progresses, the robot should tend to execute those actions which seem to be the best ones according to the Q-values, this is called *greedy policy*.

Table 1: Truncated temporal differences algorithm.

```
1. Observe the current state, s(t): s[0]=s(t)
2. Select an action a(t) for s(t): a[0]=a(t)
3. Perform action a(t)
4. Observe new state s(t+1) and reinforcement value r(t)
5. r[0]=r(t), u[0]=$max_a Q_t(s(t+1),a)$
6. For k=0,1,...,m-1 do:
   if k=0 then $z = r[k] + \gamma u[k]$
   else $z = r[k] + \gamma(\lambda z + (1-\lambda)u[k])$, $0 < \gamma,\lambda \geq 1$
7. Update the Q-values:
   $\delta = (z - Q_t(s[m-1],a[m-1]))$
   $\Delta Q_t(s[m-1],a[m-1]) = \beta\delta$
```

Despite the benefits of the RL paradigm in autonomous robot-learning, there are important prob-

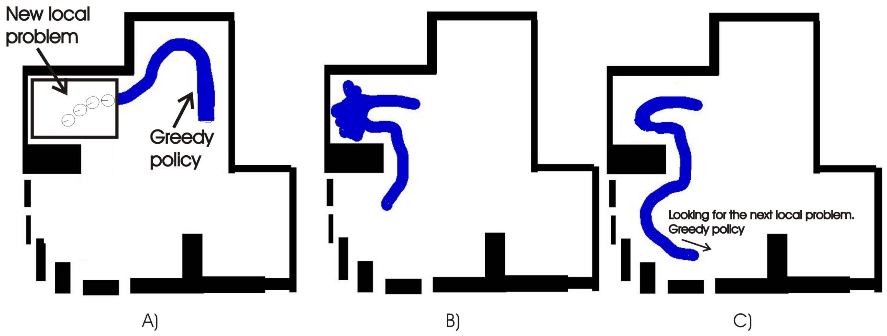


Figure 1: Schematic representation of our approach. Initially the robot moves using the greedy control policy until it finds a situation it doesn't know how to solve (a), a genetic algorithm is applied to find a solution (b), once the problem is solved the greedy policy is applied again (c).

lems to consider when it is applied. First, the time required to learn a good greedy policy increases exponentially with the number of states and the number of actions that are possible to execute in each state. On the other hand, the robot wastes an enormous amount of time trying actions that are clearly inappropriate for the task but that are selected randomly during the learning process.

2 OUR APPROACH

To solve part of the drawbacks just mentioned, we propose a learning strategy which combines three elements: *reinforcement learning (RL)*, *a genetic algorithm (GA)*, and a *dynamic representation of the environment around the robot (states)*. Basically, when our approach is applied the robot goes through three cyclical and clearly differentiated stages –figure 2–: a) *looking for a new starting position or convergence*. b) *exploration*, and c) *generation of a new population of solutions (chromosomes) for the next exploration stage*.

2.1 Looking for a New Starting Position

During this first stage the greedy policy is applied to control the robot. If the robot encounters a situation where it doesn't know how to move – local problem, (figure 1.a) –, it receives a sequence of consecutive negative reinforcements, and its position several steps before the failure is established as a new starting position for the next exploration stage.

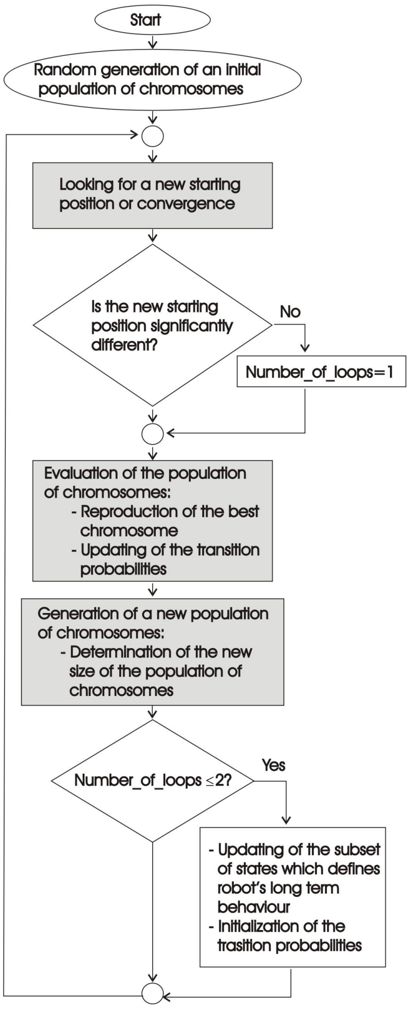

Start
Random generation of an initial population of chromosomes
Looking for a new starting position or convergence
Is the new starting position significantly different?
No
Number_of_loops=1
Evaluation of the population of chromosomes:
- Reproduction of the best chromosome
- Updating of the transition probabilities
Generation of a new population of chromosomes:
- Determination of the new size of the population of chromosomes
Number_of_loops ≤2?
Yes
- Updating of the subset of states which defines robot's long term behaviour
- Initialization of the trasition probabilities

Figure 2: Flow diagram which describes the combination of RL and GA.

2.2 Exploration Stage

Our strategy applies a GA (Davidor, 1991; Cobb and Grefenstette, 1993) in order to solve each local problem the robot finds. The GA starts with a population of solutions called chromosomes. Each chromosome –represented as π– determines the action the robot has to carry out at each state, s: $\pi(s)$. The population of chromosomes is evaluated according to an objective function called *fitness* function, which reflects for how long a chromosome is able to properly control the movement of the robot – figure 1.b –. Once a population of chromosomes has been evaluated, the sequence of states, actions, and rewards the robot received under the control of the best chromosome, is replicated off-line several times to speed up the convergence of the Q-values.

2.3 Generation of a New Population of Solutions (Chromosomes)

The population of chromosomes has to be evolved according to the fitness values. In order to do this, certain genetic operators like mutation –which carries out random changes in the chromosomes–, or chromosome crossover –which combines chromosomes to raise new solutions– have to be applied. We use the Q-values to bias the genetic operators and thus reduce the number of chromosomes which are required to find a solution. Given a particular chromosome π, the probability that mutation changes the action that this chromosome suggests for a particular state s: $\pi(s)$, depends on how many actions look better or worse than $\pi(s)$ according to the Q-values.

On the other hand, one of the chromosomes should always be the greedy policy because it brings together all that has been already learnt by the robot, and it represents the best chance to have a fast convergence towards the desired solution.

Finally, when the robot is looking for a new starting position and the greedy policy is being used to control it, if the robot moves properly during M steps before it receives negative reinforcements, only the states involved in the last K robot's movements are susceptible of being changed through the GA, while the states involved in the initial M-K actions are labelled as learnt, so that neither chromosome selection nor chromosome crossover can alter them.

The population of chromosomes is resized after its evaluation according to how close the GA is to the desired solution.

2.4 Dynamic Representation of States

We use the properties of the regular Markov chains (Bertsekas and Tsitsiklis, 1996) to reduce the number of states which are considered during the learning process. The transition matrix and what is called *steady vector* are estimated, so that only those states with a non-cero entry in the *steady vector* are considered in the learning procedure. The *steady vector* contains the probability of finding the robot in each possible state in the long-term.

3 EXPERIMENTAL RESULTS

We applied our approach to teach a mobile robot two common tasks: "wall following" and "door traversal". We have used a Nomad200 robot equipped with 16 ultrasound sensors encircling its upper part and bumpers. In all the experiments the linear velocity of the robot was kept constant (15.24 cm/s), and the robot received the commands it had to execute every 300ms.

We used a set of two layered Kohonen networks to translate the large number of different situations that the ultrasound sensors located on the front and right side of the robot may detect, into a finite set of 220 neurones – states – (R. Iglesias and Barro, 1998).

3.1 Wall Following

To teach the robot how to follow a wall located on its right at a certain distance interval, we used a reinforcement signal that is negative whenever the robot goes too far or too close from the wall being followed. The robot was taught how to perform the task in a simulated training environment, but its performance was tested in a different one. Convergence was detected when the greedy policy was able to properly control the movement of the robot for an interval of 10 minutes.

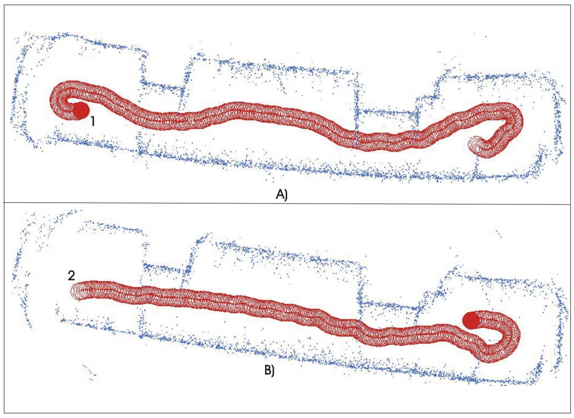


Figure 3: Real robot's trajectory along a corridor when a control policy learnt through our approach was used. For a clear view of the trajectory, figure a) shows the robot's movement in one direction and b) shows the movement along the opposite direction. Points 1 and 2 in both graphs correspond to the same robot position. The small dots represent the ultrasound readings.

When we applied our approach the average required learning time after 19 experiments was only 25 minutes and 9 seconds –the number of chromosomes varied dynamically within the interval [3,20]. When the $TTD(\lambda,m)$ was applied with the best combination of parameters we found – $\beta = 0.35$, $\lambda = 0.8$, $\gamma = 0.95$ and $m = 30$–, the average learning time after 6 experiments was 97 minutes and 30 seconds.

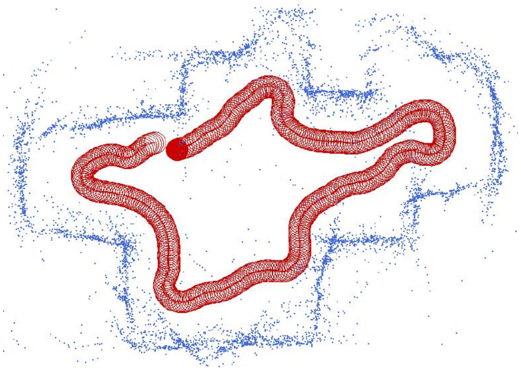


Figure 4: Real robot's trajectory when the same control policy as in figure 3 was used.

To prove that the behaviours learnt through our approach are useful, figures 3 and 4 show the movement of the robot in two real and noisy environments.

3.2 Door Traversal

We also applied our approach to teach a robot how to cross a door in the experimental scenario shown in figure 5. To learn this task the reinforcement is negative whenever the robot collides with the doorframe, the robot goes too far from the door, or the movement direction is so wrong that the robot is not facing the door any more. After 21 experiments, the average learning time was 86 minutes and 42 seconds.

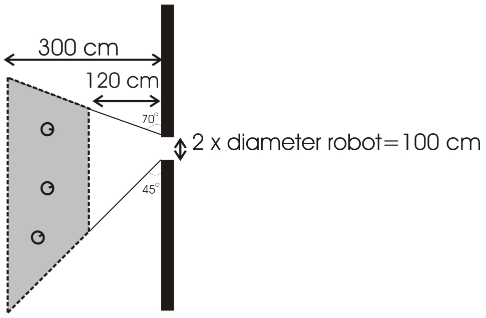


Figure 5: Experimental scenario for the door traversal behaviour. The initial positions of the robot were within the shaded area.

4 CONCLUSION

In this article we suggest the use of a new approach based on the combination of a genetic algorithm and the reinforcement learning paradigm to teach a robot how to accomplish different tasks. Our approach is mainly characterized by three aspects: 1) There are clearly differentiated exploration stages where the use of a GA which keeps less information than the RL, and through which the attention is only focused on those actions which seem to be profitable, allows the robot to carry out a fast search of solutions to those situations where the robot doesn't know how to move. 2) The information learnt through the reinforcements – Q-values – is used to bias the genetic operators – mutation, crossover, selection– and thus improve the performance of the GA. 3) Finally, the properties of the regular Markov chains represent a powerful tool to focus the attention on those states relevant in the robot's long term behaviour, avoiding the use of a big and unnecessary number of states which would delay the achievement of a good robot-control policy.

ACKNOWLEDGEMENTS

The authors thank the support from grants TIN2005-03844, PGIDIT04TIC206011PR, TIC2003-09400-C04-03.

REFERENCES

Bertsekas, D. P. and Tsitsiklis, J. N. (1996). *Neuro-Dynamic Programming*. Athena Scientific.

Cichosz, P. (1997). *Reinforcement Learning by Truncating Temporal Differences*. PhD thesis, Dpt. of Electronics and Information Technology, Warsaw University of Technology.

Cobb, H. G. and Grefenstette, J. J. (1993). Genetic algorithms for tracking changing environments. In *Proc. Fifth International Conference on Genetic Algorithms*.

Davidor, Y. (1991). *Genetic algorithms and robotics. A heuristic strategy for optimization*. World Scientific.

R. Iglesias, C. V. Regueiro, J. C. 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*.

OBSERVER BASED OPTIMAL CONTROL OF SHIP ELECTRIC PROPULSION SYSTEM

Habib Dallagi[1], Ali Sghaïer Tlili[2]
[1] Académie Navale Menzel Bourguiba, BP 7050 , [1, 2] Ecole Polytechnique de Tunisie
BP. 743 2078 La Marsa, Tunisie
habib.dallagi@ept.rnu.tn, ali.tlili@ept.rnu.tn

Samir Nejim[1]
[1] Académie Navale Menzel Bourguiba, BP 7050 ,[1, 2] Ecole Polytechnique de Tunisie
BP. 743 2078 La Marsa, Tunisie
nejim.samir @planet.tn

Keywords: Electric Propulsion Ship, Optimal Control, State Observer.

Abstract: This paper describes the synthesis of a linear state observer based optimal control of ship electric propulsion using permanent magnet synchronous motor. The proposed approach is used for the ship speed control by measuring the stator current and the motor speed. This strategy of control is made possible by using a ship speed state observer. A numerical simulation study, applied to the global system, has confirmed the efficiency and the good performances of the proposed control law.

1 INTRODUCTION

The characterization of industrial processes leads, in most cases, to nonlinear models which are generally difficult to control. The study of such processes was generally used by a linearization leading to a linear model on which the linear arsenal of controls can be applied. These different control laws use often a state feedback. However the state vector is not always measurable, so it is necessary to use state observers.

The work presented in this paper concerns the modelisation of a ship electric propulsion system. The obtained global model is strongly nonlinear, coupled and presenting non measurable variables. Indeed, a linearization was firstly elaborated and the synthesis of a control law with state feedback, for the regulation of the stator current and the ship speed of the synchronous motor, was secondly designed.

This control strategy is carried out using a linear state observer allowing the ship speed reconstruction.

This paper is organized as follows: the modeling of the different subsystems of the ship is developed in the section 2. The linearization model of the global system is elaborated in section 3. Section 4 is devoted to the optimal control development based on state observer and in section 5 simulation resultats are reported and discussed. Finally some conclusions ended this work.

2 MODELISATION OF THE ELECTRIC PROPULSION SYSTEM

2.1 Different Parts of the Ship Electric Propulsion System

An electric ship is generally composed by two principal parts (Dallagi and Nejim, 2004).

- a first part ensuring the energy production using several alternators drived either by diesel motors, or by gas turbines. It feeds the board network and the propulsion equipment.
- a second part of electric propulsion composed by one or two electric motors, each one of them driving a fixed blade propeller.

2.2 Modelling of the Permanent Magnet Synchronous Motor

By the Park transformation, the voltage equations of the permanent magnet synchronous motor are written as follows (Grellet and Clerc, 2000):

$$\begin{cases} v_d = R_s i_d + L_d \dfrac{di_d}{dt} - p\Omega L_q i_q \\ v_q = R_s i_q + L_q \dfrac{di_q}{dt} + p\Omega L_d i_d \end{cases} \tag{1}$$

The motor torque is given by:

$$C_e = p\Phi_f i_q + p\left(L_d - L_q\right) i_d i_q \tag{2}$$

The mechanical equation can be written as following:

$$I_m \dot{\Omega} = C_e - Q_{prop} \tag{3}$$

with:
Φ_f : inductor flux,
R_S : stator phase resistance,
v_d : stator voltage longitudinal component,
v_q : stator voltage transverse component,
i_d : stator current longitudinal component,
i_q : stator current transverse component,
L_d : longitudinal inductance,
L_q : transverse inductance,
I_m : shaft inertia,
P : pole pairs numbers,
Ce : electromagnetic torque,
Q_{prop} : propeller torque,
Ω: shaft speed.

2.3 Hull Resistance

During displacement, the ship is confronted to several constraints among them, the sea state, conditioned by the climatic data which is a significant factor influencing the ship behaviour. The sea applies a resistance which is opposed to the ship moving forward. Thus, to study the ship movement, it is necessary, on the one hand, to model its displacement and, on the other hand, to know the constraints which are opposed to its movement as presented in the figure 1 (Dallagi and Nejim, 2006).

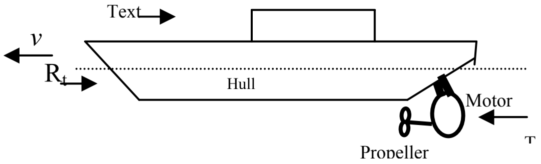


Figure 1: Ship movement.

The advance total resistance to is given by:

$$R_t = R_f + R_w + R_{app} + R_{air} \tag{4}$$

with:
R_t : advance total resistance,
R_f : friction resistance,
R_W : waves resistance,
R_{app} : appendices resistance,
R_{air} : air resistance.

This modeling is based on the resistance tests of the ship. Thus, total resistance to advance can be represented by the sum of four resistances (4). It is obtained by applying different practical pulling tests on the similarity model (Izadi-Zamanabadi and Blank, 2001. Doutreleau and Codier, 2001).

Resistance to advance can be modeled by a function of the following form (Dallagi and Nejim, 2006):

$$R_t = av^2 \tag{5}$$

with:
v: ship speed,
a: constant coefficient of the following curve.

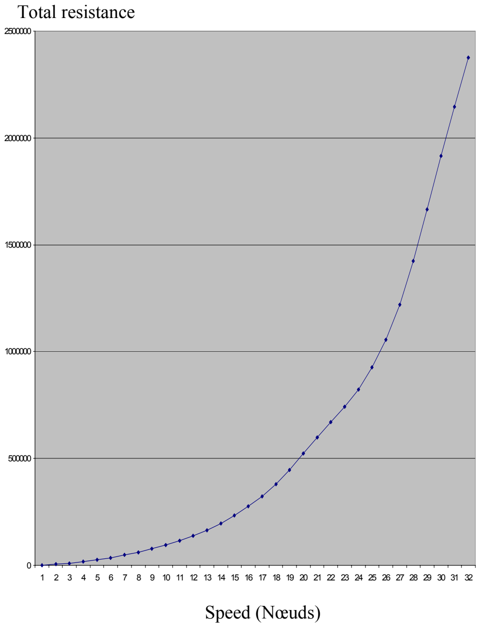


Figure 2: Advance total resistance.

2.4 Equations of the Propeller

The model of propeller thrust can be written as follows (Fosen and Blanke, 2000).(Guo, Zheng, Wang and Shen, 2005):

$$K_T = \frac{T}{\rho N^2 D^4} \tag{6}$$

with T the propeller thrust given by:

$$T = \rho n^2 D^4 K_T \tag{7}$$

The model propeller torque can be written as follows:

$$K_Q = \frac{Q_{prop}}{\rho n^2 D^5} \tag{8}$$

with Q$_{prop}$ the propeller torque given by:

$$Q_{prop} = \rho n^2 D^5 K_Q \tag{9}$$

The coefficients K$_T$ and K$_Q$ given respectively by (6) and (8) depend on the following parameters (Izadi-Zamanabadi and Blank):
v$_a$: advance speed (m/s), p$_a$: propeller pitch,
v : ship speed (m/s), J : advance coefficient,
w : wake coefficient , n : propeller speed (tr/s).
Coefficients K$_T$ and K$_Q$ are given from ship practical tests. The advance coefficient is given by (Devauchell, 1986). (Lootsma, Izadi-Zamanabadi and Nijmeijer, 2002):

$$J = \frac{2\Pi v_a}{nD} \tag{10}$$

and the advance speed is written as:

$$v_a = (1-w)v \tag{11}$$

Coefficients K$_T$ and K$_Q$ can be represented by the affine lines having the following forms:

$$K_T = r_1 + r_2 J \tag{12}$$

$$K_Q = s_1 + s_2 J \tag{13}$$

The substitution of equations (10), (11), (12) in (7) gives:

$$T = \rho n^2 D^4 (r_1 + r_2 \frac{2\Pi(1-w)v}{nD}) \tag{14}$$

by replacing the equations (10), (11) (13) in (9), it yields:

$$Q_{prop} = \rho n^2 D^5 (s_1 + s_2 \frac{2\Pi(1-w)v}{nD}) \tag{15}$$

The ship motion equation is given by (Fosen and Blanke 2000):

$$m\dot{v} = -R + (1-t)T - T_{ext} \tag{16}$$

2.5 Global Model of the Ship Electric Propulsion System

The global model of the ship electric propulsion using synchronous motor is represented by the following system.

$$\begin{cases} \dot{x} = f(x) + g(x)u \\ y_s = h(x) \end{cases} \tag{17}$$

with:

$$f(x) = \begin{pmatrix} (1/2\Pi I_m)[(p(L_d - L_q)i_d i_q + p\Phi_f i_q) \\ \quad -(s_1 \rho n^2 D^5) - (s_2 \rho n D^4 2\Pi(1-w)v)] \\ \\ (1/m)[-av^2 + (1-t)r_1 \rho D^4 n^2 \\ \quad +(1-t)2\Pi(1-w)r_2 \rho D^3 nv - T_{ext}] \\ \\ \frac{-R_s}{L_d} i_d + \frac{p2\Pi n L_q}{L_d} i_q \\ \\ \frac{-p2\Pi n L_d}{L_q} i_d - \frac{R_s}{L_q} i_q - \frac{p2\Pi n \Phi_f}{L_q} \end{pmatrix}$$

$$g(x) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{1}{L_d} & 0 \\ 0 & \frac{1}{L_q} \end{bmatrix} \text{ and } h(x) = \begin{bmatrix} n \\ id \end{bmatrix}$$

The following figure gives the structure of the ship electric propulsion system and its different subsystems:

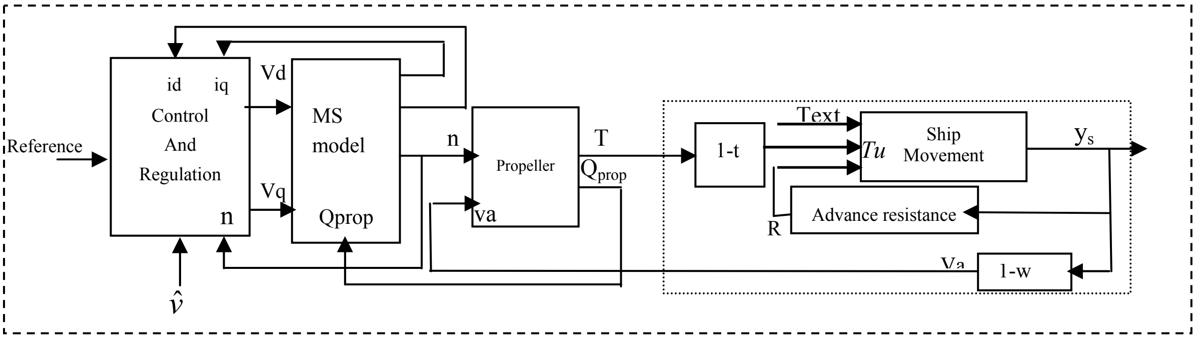


Figure 3: Synoptic of the ship propulsion system.

3 LINEARIZATION OF THE SHIP ELECTRIC PROPULSION SYSTEM

An industrial system is often intended to operate in regulation mode, i.e. the system output has to track an imposed the reference signal despite of the various disturbances. Under these conditions, the use of nonlinear state representation for the purpose of control is not necessary. A linear local state representation is sufficient.

The linearization of (17), around an operating point characterized by (x_0, y_0, u_0), is given by:

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx \end{cases} \tag{18}$$

with:

$x = \left[n, v, i_d, i_q\right]^T$ the state vector

$u = \left[v_q \quad v_q\right]^T$ the input vector,

$y_s = \left[n \quad i_d\right]^T$ the output vector.

A, B and C are the Jacobien matrices given by:

$A = \frac{\partial f}{\partial x}\Big|_{x=x_0}$; $B = \frac{\partial f}{\partial u}\Big|_{u=u_0}$ and $C = \frac{\partial h}{\partial x}\Big|_{x=x_0}$

$$A = \begin{bmatrix} \frac{a_n}{2\Pi I_m} & \frac{a_v}{2\Pi I_m} & \frac{a_{id}}{2\Pi I_m} & \frac{a_{iq}}{2\Pi I_m} \\ \frac{b_n}{m} & \frac{b_v}{m} & 0 & 0 \\ c_n & 0 & c_{id} & c_{iq} \\ d_n & 0 & d_{id} & d_{iq} \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{1}{L_d} & 0 \\ 0 & \frac{1}{L_q} \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

with:

$$a_{iq} = 2\Pi p \Phi_f + p(L_d - L_q) i_{d0}$$

$$a_v = -\rho 2\Pi D^4 s_2 (1 - w_0) n_0$$

$$a_{id} = p(L_d - L_q) i_{d0}$$

$$a_n = -2 s_1 \rho D^5 n_0 - s_2 \rho 2\Pi D^4 (1 - w_0) v_0 + p(L_d - L_q) i_{d0}$$

$$b_n = 2 a_1 \rho D^4 (2\Pi)^2 (1 - t_0) n_0 + (1 - t_0) 2\Pi (1 - w_0) a_2 \rho D^3 v_0$$

$$b_v = -2 a v_0 + a_2 \rho D^3 (1 - t_0) 2\Pi (1 - w_0) n_0$$

$$c_n = p \frac{L_q}{L_d} i_{q0}, \quad c_{id} = -\frac{R_s}{L_d}, \quad c_{iq} = \frac{2\Pi p n_0 L_q}{L_d}$$

$$d_n = -p \frac{L_d}{L_q} i_{d0} - p 2\Pi \frac{\Phi_f}{L_q}, \quad d_{iq} = -\frac{R_s}{L_q}$$

$$d_{id} = \frac{p 2\Pi n_0 L_d}{L_q}$$

4 OPTIMAL CONTROL OF THE SHIP ELECTRIC PROPULSION

4.1 Principle of the Optimal Control

To obtain an optimal control law for the ship electric propulsion system, we minimize the following criterion (Toscano, 2005). (Rachid and Mehdi, 1997 (Corriou , 1996.):

$$J = \frac{1}{2} \int_0^\infty (u^T R u + \varepsilon^T Q \varepsilon) dt \tag{19}$$

with:

R a symmetric positive definite matrix,

Q a symmetric non-negative definite matrix,

$\varepsilon(t) = e(t) - y(t)$ is the difference between the reference and the output vector.

The control law is then given by:

$$u(t) = Fe(t) - Kx(t) \tag{20}$$

where:

- $e(t) = \left[i_{dref} \quad v_{ref}\right]^T$ is the reference vector.
- K is control gain matrix defined by:

$$K = R^{-1} B^T P \tag{21}$$

- F is reference gain matrix given by:

$$F = R^{-1} B^T (A^T - PBR^{-1} B^T)^{-1} PC^T Q \tag{22}$$

with P the solution of the following Riccati equation:

$$A^T P + PA - PBR^{-1} B^T P + C^T QC = 0 \tag{23}$$

4.2 The Ship Speed State Observer

To design the sate feedback optimal control law, it is necessary to reconstruct the ship speed v in order to be controlled. For this purpose, we propose a linear state observer using the output vector $y_s = \left[i_d \quad n\right]$ and the vector $u = \left[u_d \quad u_q\right]$.

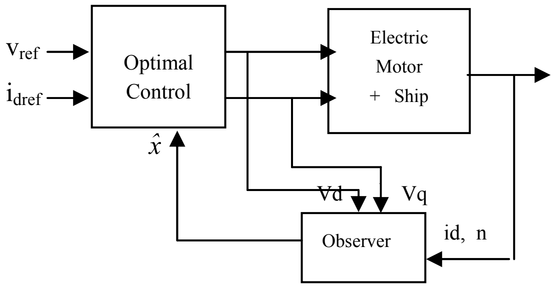


Figure 4: Control with ship speed observer.

The structure a luenberger observer is given by (Stoorvogel , Saberi and Chen 1994). (Mieczarski, 1988):

$$\begin{cases} \dot{\hat{x}} = A\hat{x} + Bu + L(y_s - \hat{y}) \\ \hat{y} = C\hat{x} \end{cases} \tag{24}$$

where:

$\hat{x}$ is the output vector of the state observer

L is the observer gain

This structure can be written in this form:

$$\dot{\hat{x}} = \hat{A}\hat{x} + Bu + Ly_s \tag{25}$$

with : $\hat{A} = A - LC$

To have an asymptotic convergence of the observed state towards the real state, it is necessary to choose the gain L such that the matrix $(A - LC)$ has negative real part eigen values. The control law using the state observer is presented as follows:

$$u(t) = Fe(t) - K\hat{x}(t) \tag{26}$$

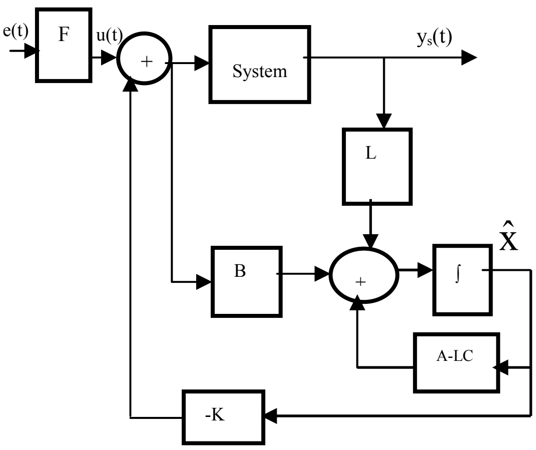


Figure 5: Observed state feedback control.

5 SIMULATION RESULTS SECOND SECTION

A digital simulation of the proposed control law with the designed state observer has been carried out with on the ship electric propulsion system using the following characteristics.

Para.	value		Par.	Value
ρ	1025 Kg/m ^{3}		Q $_n$	3.1480010 ^{5} Nm
D	5.9 m		T $_n$	3.82000 10 ^{5}N
m	20690000 kg,		Q $_f$	0.382 10 ^{5}
t	0.178		T $_{ext.}$	-1.8*0.1*10^5N
w	0.2304		s $_1$	0.075
a	1.54 10 ^{6}		s$_2$	0.1375
r2	1.1		r $_1$	0.5

The resolution of the Riccati equation (23) yields to:

- The optimal control gain:

$$K = \begin{bmatrix} 0.9807 & 0.0520 & 10.3830 & -0.0002 \\ 0.1064 & 0.0467 & 9.7618 & 0.0030 \end{bmatrix}$$

- The observer gain:

$$L = \begin{bmatrix} 3.5 & 1964.8 \\ -10.6 & -339.6 \\ 0.1 & 6.6 \\ 10.3 & 1362.9 \end{bmatrix}$$

The resolution of the equation (22) gives the reference gain:

$$F = \begin{bmatrix} 0.9963 & -3.05078 \\ 0.1064 & 26.3032 \end{bmatrix}$$

For the designed control, we impose i_{dref}=0, so that the electromagnetic torque C_e will be proportional to i_q.

In order to control the ship speed v it is necessary to change the motor speed n through the stator component i_q which modify the electromagnetic torque (Dallagi and Nejim, 2005).

The performances of the proposed strategy control law are depicted in the figures 6, 7, 8 and 9.

The ship speed is needed to reach the reference speed value V_{ref}=8m/s in the interval [0 150s] and V_{ref}=10m/s in the interval [150 300s].

It appears from figure 6, that the proposed control law allows a convergence towards the desired value of the ship speed v.

The figure 7 shows the behavior of the motor speed. It's clear that the ship speed changes where the variation of the propeller speed changes. Furthermore, we impose i_{dref}=0, so the electromagnetic torque becomes proportional to statoric current i_q.

In order to control the motor speed n, one modify the electromagnetic torque C_{em} by changing the stator current i_q through the regulation of the voltage component $v_{q.}$(Grellet and Clerc,2000).

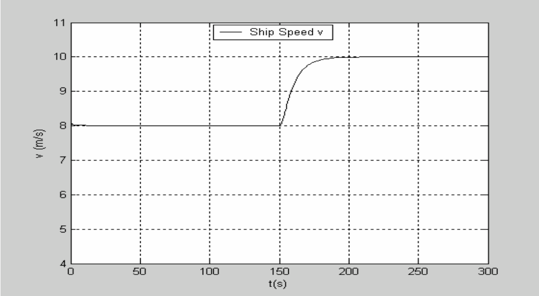


Figure 6: Ship Speed.

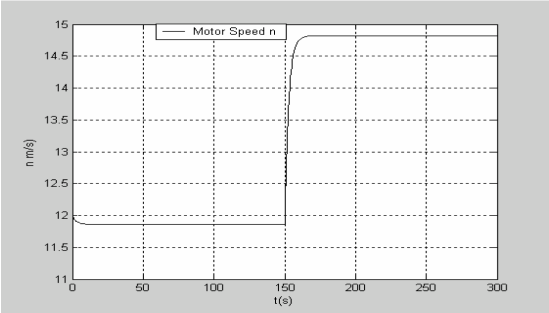


Figure 7: Motor Speed.

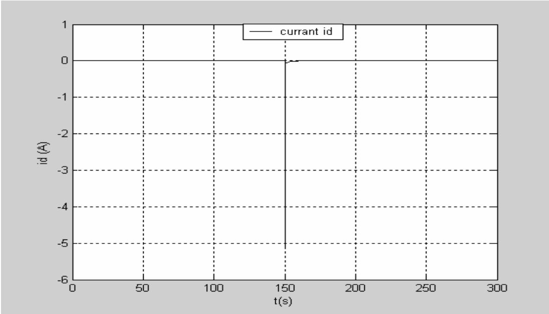


Figure 8: Current id.

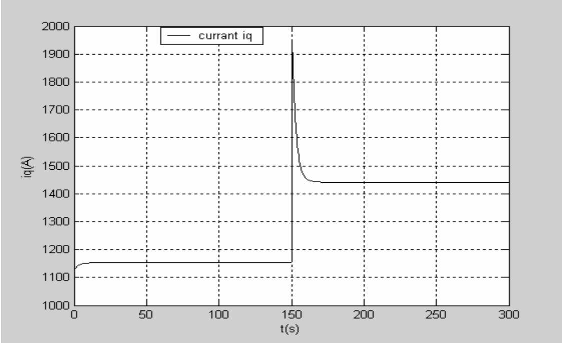


Figure 9: Current iq.

6 CONCLUSIONS

In this paper we have proposed an optimal control law using a Luenberger sate observer to control the ship speed.

The designed observer is used to reconstruct the ship speed in order to complete the control strategy. It has been shown from the simulated results that the proposed estimated state feedback optimal control permits the regulation of the ship speed which converges exactly to the imposed reference.

REFERENCES

Dallagi, H., Nejim, S., 2004. Modélisation et simulation d'un système de propulsion diesel-électrique de Navire. In *3ème Conférence Internationale, JTEA,* Tunisie.

Grellet,Guy., Clerc, Guy., 2000. Actionneurs Electriques, principe modèle et commande. In *Editions Eyrolles.*

Dallagi, H,. Nejim, S., 2006. Conception d'un programme de prédiction et d'estimation de la puissance propulsive des navires à propulsion électrique. In *4ème Conférence Internationale, JTEA,* Tunisie.

Izadi-Zamanabadi, R, Blanke, M., Katebi, S.2001. Cheap diagnostisis using structural modelling and fuzzy-logic based detection.

Fosen, T.I Blanke, M. 2000. Nonlinear output feedback control of underwater vehicle propellers usingfeedbak from estimated axial flow velocity. In *IEEE journal of oceanic engineering, vol 25, no2*

Guo Y, Zheng, Y Wang B Shen A. 2005. Design of ship electric propulsion simulation system. In *Proceeding of the fourth international conference on machine learning and cybernetics*. Guangzhou.

Devauchell, P., 1986 "Dynamique du Navire. In *Masson*, Paris.

Toscano, R,. 2005. *Commande et diagnostic des système dynamique*,. In *Elipses Edition* , Paris.

Corriou J.P,. 1996. Commande des procédés. In *Techniques et documentations,* Paris.

Rachid, A,. Mehdi, D,. 1997. Réalisation réduction et commande des systèmes linéaire. In *édition technip Paris.*

Stoorvogel , A.A., Saberi, A., Chen, B.M., 1994. A Reduced order observer base control design for optimization. In *IEEE transaction automatic*

Mieczarski.,W., 1988. Very fast linear and nonlinear observer. In *int. J. control.*

Dallagi, H., Nejim,S., 2005. Modélisation and Simulation of an Electric Drive for a Two Propeller Ship. In 17$^{\text{ème}}$ Congrès Mondial IMACS, *Calcul Scientifique, Mathématiques Appliquées et Simulation*, Paris, France.

Doutreleau, Y., Codier ,.S. 2001. Les problèmes des navires à grande vitesse. In *J.T* .

Snitchler G., Gambe B.B., Kalsi S., Ige, S.O. 2006. The stats of HTS ship propulsion motor developments. In *IEEE Transaction on applied superconductivity.*

Kalsi S, Gambe B.B, Snitchler G,. 2005. The performance of 5Mw high temperature superconductor ship propulsion motor. In *IEEE Transaction on applied superconductivity, vol, 15, n°2*

Dallagi, H., Nejim, S., 2005. Optimization of an integrated power system of an electric ship. In *International Conference on ship propulsion and railway traction systems,* Bologna–Italy,

Zimin, W.Vilar., Roger, A.Douglas., *2005*. Effectiveness of generator Strategies on meeting pulsed load requirements in ship electric system. In *IEEE electric ship technologies symposium*.

Izadi-Zamanabadi R. Blank, M,. A ship propulsion system model for fault-tolerant Control," *Department of control Engineering Aalborg university, Dennemark.*

Gillmer T.C., Jonson B,. 1982. Propulsive force and propulsion system. In N*aval Institue Press, Annapolis*, Maryland.

Blanke, M., Izadi-Zamanabadi, R,. 1998. Reconfigurable control of a ship propulsion plant. In *Control Applications in Marine Systems*, *CAMS*, Fukuoka, Japan.

Lootsma, T.F., Zamanabadi, R.I., Nijmeijer, H., 2002. A Geometric approach to diagnosis applied to a ship propulsion problem. In *15th Triennal World Congress, IFAC* , Barcelona, Spain.

Steinar J.Dale., 2005. Ship power system testing and simulation. In *IEEE Electric ship technology symposium*.

Rudly, Limpaecher., 2000. Novel converters for electric ship propulsion system and shipboard power distribution. In *IEEE transactions on energy conversion*.

A NOVEL BLOCK MOTION ESTIMATION MODEL FOR VIDEO STABILIZATION APPLICATIONS

Harish Bhaskar and Helmut Bez
Research School of Informatics, Loughborough University
H.Bhaskar@lboro.ac.uk, H.E.Bez@lboro.ac.uk

Keywords: Video stabilization, motion compensation, motion estimation, genetic algorithms, kalman filtering.

Abstract: Video stabilization algorithms primarily aim at generating stabilized image sequences by removing unwanted shake due to small camera movements. It is important to perform video stabilization in order to assure more effective high level video analysis. In this paper, we propose novel motion correction schemes based on probabilistic filters in the context of block matching motion estimation for efficient video stabilization. We present a detailed overview of the model and compare our model against other block matching schemes on several real-time and synthetic data sets.

1 INTRODUCTION

Video data obtained from compact motion capture devices such as hand-held, head mounted cameras, etc. has gained significant attention in recent years. Video stabilization, as the name suggests, deals with generating stabilized video sequences by removing unwanted shakes and camera motion. Several methods have been proposed in the literature for accomplishing video stabilization. However, the accuracy of motion estimation is a key to the performance of video stabilization. (Y.Matsushita and H.Y.Shum, 2005) propose a combination of motion inpainting and deblurring techniques to accomplish robust video stabilization. Several other research contributions have been made to video stabilization including, probabilistic methods (A.Litvin and W.C.Karl, 2003), model based methods, etc. Methods such as (M.Hansen and K.Dana, 1994)(Y.Yao and R.Chellappa, 1995)(P.Pochec, 1995)(J.Tucker and Lazaro, 1993)(K.Uomori and Y.Kitamura, 1990), propose to combine global motion estimation with filtering to remove motion artifacts from video sequences. These schemes perform efficiently only under restricted conditions and are again limited by the efficiency of the global motion estimation methodology. (K.Ratakonda, 1998) have used an integral matching mechanism for compensating movement between frames. (T.Chen, 2000) propose a 3 stage video stabilization algorithm based on motion estimation. The process includes motion estimation for computing local and global motion parameters, motion smoothing for removing abrupt motion changes between subsequent frame pairs and finally a motion correction methodology for stabilization. In this paper we extend the work presented in (T.Chen, 2000) to accommodate a novel motion correction mechanism based on moving average filters and Kalman filtering alongside a motion estimation strategy that combines vector quantization based block partitioning with a genetic algorithm based block search for motion estimation.

2 PROPOSED MODEL

The video stabilization model proposed in this paper extends a parametric motion model proposed in (T.Chen, 2000). A detailed overview of the proposed model in the form of a pseudo code is as follows.

- Input at a time instant t two successive frame pairs of a video sequence, f_t & f_{t+1} where $1 \leq t \leq N$, where N is total number of frames in the video
- Image frame f_t is initially partitioned into 4 blocks using the vector quantization algorithm

described in the subsection below, **Note:** Every block represents an image region

- For every block b
  - The centroid (x_c, y_c) of the block is computed
  - A genetic algorithm as described below is used to accurately match the block in the successive frame f_{t+1}
  - If the genetic algorithm accurately matched the block in frame f_t to frame f_{t+1} (with error = 0), then the motion vector is evaluated as $(x^* - x, y^* - y)$ where (x^*, y^*) is the estimated transformed centroid of the block in frame f_{t+1}
  - If the genetic algorithm returned non-zero matching error then the process is repeated by further sub dividing block.
- The process is terminated either when no further splitting is needed or a predefined block size is reached.
- If the processed frame pair is (f_t, f_{t+1}) where $t = 1$, then proceed to next frame pair, otherwise if $t > 1$, then run motion correction using any of the proposed filter mechanisms specified to generate smoothed motion vectors $MV_\aleph$
- Compute the difference between the original motion vectors MV and the smoothed motion vectors $MV_\aleph$ adjust the original motion vectors by the factor of difference $MV_{comp} = MV \pm (MV - MV_\aleph)$
- Generate Stabilized frames using the original motion vector MV and compensated motion vectors MV_{comp} and represent them as f^*_{t+1} and f^{*comp}_{t+1}
- Deduce the PSNR of the two versions of stabilized frames using, PSNR for a gray scale image is defined as:

$$10\ log_{10} \left[\frac{255^2}{\frac{1}{HW}\sum_H \sum_W \|f_{t+1} - f_{comp}\|^2} \right] \quad (1)$$

  where, (H, W) is the dimensionality of the frames and f_{t+1} and f_comp are the intensity components of the original target and the motion compensated images which will equal f^*_{t+1} and f^{*comp}_{t+1} respectively. PSNR values generally range between 20dB and 40dB; higher values of PSNR indicate better quality of motion estimation.
- If $PSNR_{comp} \geq PSNR$ then use f^{*comp}_{t+1} as stabilized frame for subsequent analysis otherwise use f^*_{t+1}.

2.1 Motion Estimation

A brief description of the algorithms is specified.

2.1.1 Block Partitioning Based on Vector Quantization

For the block partitioning phase, we start by using vector quantization to provide the block matching scheme with the position of partitioning.

- Set the number of codewords, or size of the codebook to 4. This assumes that we need 4 regions to emerge out of the image frame during the quantization process.
- Initialize the positions of the codewords to $(\frac{w}{4}, \frac{h}{4})$, $(\frac{w}{4}, \frac{3h}{4})$, $(\frac{3w}{4}, \frac{3h}{4})$, $(\frac{3w}{4}, \frac{3h}{4})$ where (w, h) is the width and height of the block respectively. By this we assume that the worst case partition could be the quad-tree partition.
- Determine the distance of every pixel from the codewords using a specific criterion. The distance measure is the sum of differences in the gray intensity and the locations of the pixels.
- Group pixels that have the least distance to their respective codewords.
- Iterate the process again by recomputing the codeword as the average of each codeword group (class). If m is the number of vectors in each class then,

$$CW = \frac{1}{m} \sum_{j=1}^{m} x_j \quad (2)$$

- Repeat until either the codewords don't change or the change in the codewords is small
- Associated with these 4 codewords, there are 4 configurations possible for partitioning the image frame into blocks. The configurations arise if we assume one square block per configuration. It is logical thereafter to find the best configuration as the center of mass of these 4 possible configurations. The center of mass will now be the partition that splits the image frame into blocks.

2.1.2 Genetic Algorithm Search

The inputs to the genetic algorithm are the block b_t and the centroid (x_c, y_c) of the block.

- Population Initialization: A population P of these n chromosomes representing (T_x, T_y, θ) is generated from uniformly distributed random numbers where,
  - $1 \leq n \leq limit$ and $limit$ (100) is the maximum size of the population that is user defined.
- To evaluate the fitness $E(n)$ for every chromosome n:

  - Extract the pixels locations corresponding to the block from frame f_t using the centroid (x_c, y_c) and block size information
  - Affine Transforming these pixels using the translation parameters (T_x, T_y) and rotation angle θ using,
    $\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & T_x \\ 0 & 1 & T_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$
  - If b_t represents the original block under consideration, b^*_{t+1} represents the block identified at the destination frame after transformation and (h, w) the dimensions of the block, then the fitness E can be measured as the mean absolute difference (MAD).

$$MAD = \frac{1}{hw} \sum_{i=1}^{h} \sum_{j=1}^{w} \left| b_t(i,j) - b^*_{t+1}(i,j) \right| \tag{3}$$

- Optimization: Determine the chromosome with minimum error $n_{emin} = n$ where E is minimum. As this represents a pixel in the block, determine all the neighbors (NH_k) of the pixel, where $1 \leq k \leq 8$.
  - For all k, determine the error of matching as in Fitness evaluation.
  - If $E(NH_k) < E(n_{emin})$, then $n_{emin} = NH_k$
- Selection: Define selection probabilities to select chromosomes for mutation or cloning. Perform cross-over and mutation operations by swapping random genes and using uniform random values.
- Termination: Three termination criterion such as zero error, maximum generations and stall generations. Check if any condition is satisfied, otherwise iterate until termination.

2.2 Motion Smoothing

The work of (T.Chen, 2000) suggested the use of a moving average low pass filter for this process. In this paper, we extend the moving average filter to an exponentially weighted moving average filter.

2.2.1 Exponentially Weighted Moving Average Filter

A detailed pseudo code describing the process is as follows.

- Set the number of frame pairs across which the moving average filter to be any scalar J
- Compute the parameter alpha $\propto$ = $(1 \div J)$
- Compute the weighting factors for every frame pair between 1 and J as $w = \propto^{i-1} \times (1 - \propto)$, where, $1 \leq i \leq J$ (Use these weighting factors as a kernel for the convolution process)
- Generate a vector of the motion vectors and rotation parameter theta across all frames; MV and θ
- Perform Convolution to generate the smoothed motion vectors, $MV_{\aleph} = MV \otimes w$ and $\theta_{\aleph} = \theta \otimes w$

2.2.2 Kalman Filter

A 2D Kalman filter can be used to predict motion vector of successive frames given the observation or motion vectors of the previous frames. An algorithm describing the smoothing process is listed below.

- Initialize the state of the system using (x, y, dx, dy), where (x, y) is the observation (i.e. the centroid of the block) and (dx, dy) is the displacement of the centroids. The values of state can be initialized using the motion estimates between the first successive frame pair.
- The state of the system S at time instant $t + 1$ and the observation M at time t can be modeled using

$$S(t+1) = AS(t) + Noise(Q) \tag{4}$$

$$M(t) = S(t) + Noise(R) \tag{5}$$

- Initialize A and noises Q, R as Gaussian.
- Perform the predict and update steps of standard Kalman filter
  - Initialize state at time instant t_0 using $S_0 = B^{-1}M_0$ and error covariance $U_0 = \begin{bmatrix} \in & 0 \\ 0 & \in \end{bmatrix}$
  - Iterate between the predict and update steps
  - Predict: Estimate the state at time instant $t + 1$ using $S^-_k = AS_{k-1}$ and measure the predicted error covariance as $U^-_k = AU_{k-1}A^T + Q$
  - Update: Update the correct, state of the system $S_k = S^-_k + K(M_k - BS^-_k)$ and error covariance as $U_k = (I - KB)U^-$
  - Compute K, the Kalman gain using $K = U^-_k B^T (BU^-_k B^T + R)^{-1}$
- Smooth the estimates of the Kalman filter and present the smoothed outcomes as $MV_{\aleph}$

3 RESULTS AND DISCUSSION

Here, in this section, we present some sample results of the stabilization task on wildlife videos taken at a zoological park. Performance of the video stabilization scheme can only be visually evaluated. We provide some sample frames illustrating the quality of video stabilization. Figure 1 compare the video stabilization quality of the base-line model versus the proposed model. As we can clearly visualize there is quite a increased quality in the stabilized version of the proposed model in comparison to the stabilized version of the base model. The motion correction scheme using the Kalman filter was sufficient to smooth the motion vector correctly. The reason to this is because, the changes observed in the capture was linear. Similarly in figures 2, we compare the quality of video stabilization using another sample clip from the same wildlife video. The movement of the camera in this sequence was more abrupt and random in directions. We observed that the proposed model using Kalman filter could not handle these changes well and as well generate a good quality stabilized output. However, the motion correction mechanism using the exponentially weighted moving average filter could produce much better results.

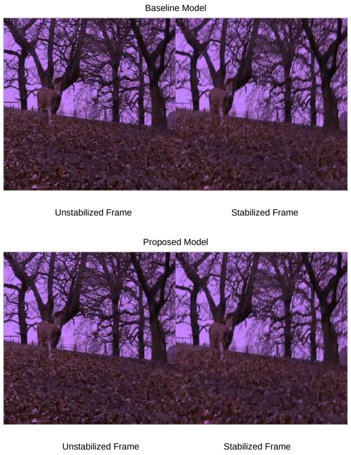


Figure 1: Model Performances on Video Sample Clip 3.

4 CONCLUSION

In this paper, we have presented a novel mechanism of motion correction and block based motion estimation strategy that combines vector quantization based block partitioning mechanism with the genetic algorithm based block search scheme applied to video stabilization. The model was tested on several real time datasets and the results have revealed a high degree of performance improvement when compared to existing video stabilization model based on motion estimation and filtering.

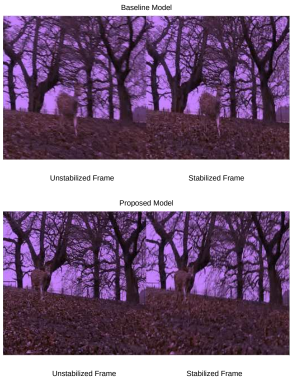


Figure 2: Model Performance on Video Sample Clip 6.

REFERENCES

A.Litvin, J. and W.C.Karl (2003). Probabilistic video stabilization using kalman filtering and mosaicking. *Image and Video Communication and Processing*.

J.Tucker and Lazaro, A. S. (1993). Image stabilization for a camera on a moving platform. In *IEEE Conference on Communications, Computers and Signal Processing*, pages 734 – 737.

K.Ratakonda (1998). Real-time digital video stabilization for multi-media applications. In *Proceeding of the 1998 IEEE International Symposium on Circuits and Systems*, pages 69–72.

K.Uomori, A.Morimura, H. T. and Y.Kitamura (1990). Automatic image stabilizing system by full-digital signal processing. In *IEEE Transactions on Consumer Electronics*, pages 510–519.

M.Hansen, P. and K.Dana (1994). Real-time scene stabilization and mosaic construction. *Image Understanding Workshop Proceedings*.

P.Pochec (1995). Moire based stereo matching technique. In *Proceedings of ICIP*, pages 370–373.

T.Chen (2000). Video stabilization algorithm using a block based parametric motion model. Master's thesis.

Y.Matsushita, E.Ofek, X. and H.Y.Shum (2005). Full-frame video stabilization. In *IEEE International Conference on Computer Vision and Pattern Recognition*.

Y.Yao, P. and R.Chellappa (1995). Electronic image stabilization using multiple image cues. In *Proceedings of ICIP*, pages 191–194.

THE PROTOTYPE OF HUMAN – ROBOT INTERACTIVE VOICE CONTROL SYSTEM

Miroslav Holada, Igor Kopetschke, Pavel Pirkl, Martin Pelc, Lukáš Matela, Jiří Horčička
and Jakub Štilec
Faculty of Mechatronics and Interdisciplinary Engineering Studies, Technical University of Liberec
Hálkova 6, Liberec, Czech Republic
{miroslav.holada, igor.kopetschke, pavel.pirkl, martin.pelc, lukas.matela, jiri.horcicka, jakub.stilec}@tul.cz

Keywords: Distributed speech recognition, robot control, image processing.

Abstract: This contribution shows our recent activities of human – robot voice-control system. The keynote lays in design of a system that allows easy control and programming of various robots by a uniform interactive interface. Generally the robots are actuated by sets of control commands, sometimes by a manual control interface (touchpad, joystick). The operator has to know the control commands, syntax rules and other properties necessary for successful robot control. Our system offers commands like "move left" or "elevate arm" that are translated and sent into the corresponding device (robot). Speech recognition is based on an isolated-word HMM engine with distributed recognition system (DSR). The vocabulary may contain thousands of word and short phrases. It allows us to design an easy dialogue scenario for the robot control system. The system allows direct robot control like moving or programming short sequences. A video camera is included for displaying the working place and employed algorithms of image processing allow the system to detect the position of objects in the working area and facilitate robot's navigation.

1 INTRODUCTION

The goal of this project is a PC-software-based interactive system for general-purpose robot voice-control. This paper describes the designed prototype, its structure and the dialogue strategy in particular.

The interactive control of robots could be used in special situations, when a robot is working in dangerous areas and no programming beforehand is possible. It could also be used in a situation when supervised learning for robot's later autonomous operation has to be done, without knowledge about the robot's programming language. The presented paper follows on this reality.

2 PROJECT FEATURES

The project is based on a former research. The research involved a voice-control dialog system, speech recognition vocabulary design and speech synthesis feedback for user command confirmation. Together with a scene manager and a digital image processing module, it forms the core of the control system as shown in figure 1. The components are described below.

2.1 Scene Manager

The scene manager forms a connection between the main program (engine) and the image processing part. It actually controls the image processing module and initiates image acquisition and processing. Using the processed image data, it updates the scene database, keeps track of objects exposed on the scene and provides the scene object and image data to the main engine. It is also aware of the robot's coordinate system and plans the robot's movement when requested by the engine.

The database itself consists of two types of data. It contains the list of parametrized objects detected on the scene as well as the robot calibration data. The latter allows mutual image-space to robot-space coordinate translation which is used in robot navigation. Each object detected on the scene is internally represented as a data object (class instance), all the objects are stored in a dynamic list. Some of the attributes are: a unique object identifier, object's shape descriptor, central point coordinates,

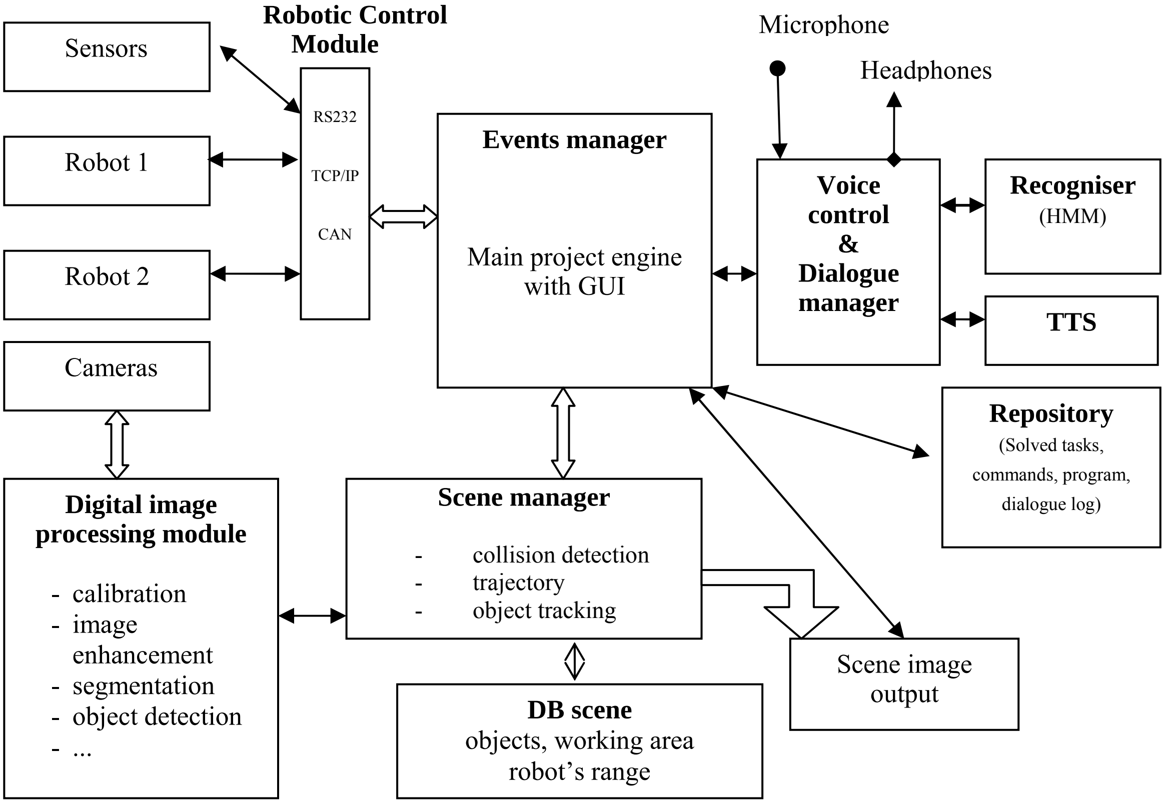


Figure 1: Scheme of the designed system.

bounding rectangle etc. Such data allows smooth object manipulation and serves as a base for object collision avoidance along the manipulation trajectory.

The scene manager also combines unprocessed camera image with scene data to highlight detected objects and to present them to the user via a GUI as shown in figure 2. The user has a view of the computer's scene understanding and may correctly designate objects of interest in his or her commands.

Being in its early stages, the project currently works only with 2D data and relies on the user's z-axis navigation aid. The system is expected to incorporate a second camera and 3D computer vision in the future to become fully 3D aware.

2.2 Image Recognition

The robot's working area is captured by a colour high-resolution digital camera (AVT Marlin F-146C, 1/2'' CCD sensor). The camera is placed directly above the scene in a fixed position. We implemented a simple interactive method to synchronize the robot's coordinate system (XY) and the camera's one using pixel units. We prepare modifications to compensate geometric distortions introduced by a camera lens.

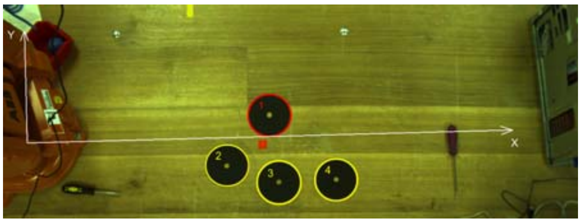


Figure 2: Working scene with highlighted objects of circular shape and robot's coordinate system.

Digital image processing methods are placed in a DIP library which is served by the scene manager with the object database. Figure 2 shows the circular object detection using the reliable Hough transform (HT). HT is commonly used for line or circle detection but could be extended to identify positions of arbitrary parametrizable shapes. Such edge-based object detection is not too sensitive to imperfect input data or noise. Using a touch-display or verbal commands it is possible to focus the robot onto a chosen object (differentiated by its color or numbering) and than tell the robot what to do.

2.3 Robots Description

The prototype system uses a compact industrial general-purpose robotic arm (ABB IRB 140). The robot is a 6-axes machine with fast acceleration, wide working area and high payload. It is driven by a high performance industrial motion control unit (S4Cplus) which employs the RAPID programming language. The control unit offers extensive communication capabilities - FieldBus, two Ehternet channels and two RS-232 channels. The serial channel was chosen for communication between the robot and the developed control system running on a PC.

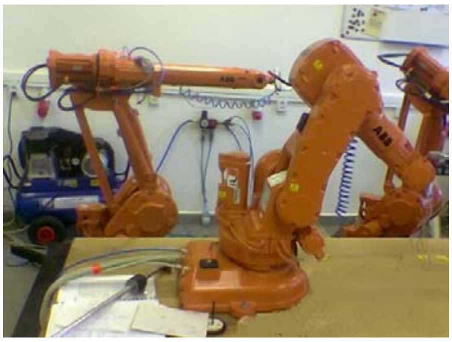


Figure 3: Robots used.

The robotic control SW module simplifies the robot use from the main engine's point of view. It abstracts from the aspects of physical communication and robot's programming interface. It either accepts or refuses movement commands issued by the core engine (depending on command's feasibility). When a command is accepted, it is carried out asynchronously, only notifying the engine once the command is completed.

2.4 Distributed Computation

Most of the system's modules are developed and run on a standard PC to which the robot is connected. Since some of the software modules require significant computational power, the system's response time was far from satisfactory when the whole system ran on a single computer. Therefore, the most demanding computations (namely the object recognition and the voice recognition) were distributed to other (high performance) computers via network (TCP connections).

3 DIALOGUE STRATEGY

The voice interface between an operator and the controlled process is provided by a speech recogniser and a text-to-speech synthesis (TTS) system (both for Czech language). The TTS synthesis system named EPOS was developed by URE AV Prague. It allows various male or female voices with many options of setting.

The speech recognition is based on a proprietary isolated word engine that was developed in previous projects. The recogniser is speaker independent, noise robust, phoneme based with 3-state HMM (Hidden Markov Models) and 32 Gaussians. It is suitable for large vocabularies (up 10k words or short phrases) and allows us to apply various commands and their synonyms.

The dialog system is event-driven. We can categorize the events into three fundamental branches: operator events, scene manager events and device events.

3.1 Operator Events

Operator events usually occur in response to operator's requests. For example commands which are supposed to cause robot's movement, object detection, new command definition or detection of a new object. This kind of event can occur at any time, but the dialog manager has to resolve if it was a relevant and feasible request or if it was just a random speech recognition error.

Although the acoustic conditions in robotic applications usually involve high background noises (servos, air-pump), the speech recogniser works commonly with 98% recognition score. If the operator says a wrong command or a command out of context (for example, the operator says "drop" but the robot doesn't hold anything) then the manager asks him or her for a feasible command in the stead of the nonsensical one.

3.2 Scene Manager Events

This sort of event occurs when the scene manager detects a discrepancy in the scene. For example when the operator says "move up" and the robot's arm moves all the way up until the maximum range is reached. When this happens a scene event is generated and the system indicates that the top position was reached.

Other scene event occurs when the operator wants to take up an object, but the system does not know which one because of multiple objects detected on the scene. This event generates a query to the operator for proper object specification.

3.3 Device Events

These events are produced by external sensors and other components and devices connected to the system. They are processed in the event manager where corresponding action is taken. The response manifests itself in the form of a request for the operator, or more often causes a change in robot's behaviour.

The difference between scene manager events and device events is that scene events are generated by the system itself (based on a known scenario, robot geometry, object shape and position). They are computed and predictable. On the other hand, device events' time cannot be exactly predicted before they actually happen.

3.4 Example of Dialog

For a simpler robot orientation and navigation the positions on the scene are virtualized. They are named by Greek letters like "Position α" or "Position β". These virtual positions may be redefined to suit the operator's needs. A blind-area may also be defined and it is completely omitted from any image processing and anything in this area is completely ignored.

We can gather up black disks (see figure 2.) and put them to some other place on the scene. This place is defined like "Position α" and we setup the "blind area" on the same coordinates. After that the operator starts a dialog, for example:

"Start recording new command."

"I'm recording" ...robot says

"Search objects"

"I'm searching ... Four objects were found"

"Move on first"

"I'm moving ... Done"

"Take it."

"Ok"

"Move on position alpha."

"I'm moving ... Done"

"Put it"

"Ok"

"Stop recording."

"I stop the recording. Please, say new command"

"Search" "Disks" "Done"

"New command is entered and named: Search disks. Is it right."

"Yes"

"Repeat command"

"Enter command"

"Search disks"

"OK"

....

"No object found. Repeating done."

The robot finds the remaining three disks and puts them in the selected area. If any disk cannot be found then the robot interrupts executing the given command and waits for next action.

4 CONCLUSIONS

The system is especially usable as an accessory robot control interface for assistant and second-rate operations. The presented prototype now cooperates only with one industry robot (ABB) but the robotic control module may easily be extended to support other robots (Katana, mobile robots, etc.) as well.

The designed system offers robot control and robot task programming even to people without explicit programming language knowledge. It is sufficient for the operator to know the Czech voice interface of the presented system.

ACKNOWLEDGEMENTS

This work was supported by the internal grant IG FM 2007/002.

REFERENCES

Nouza, J., Nouza, T.: A Voice Dictation System for a Million-Word Czech Vocabulary. In: *Proc. of ICCCT* 2004, August 2004, Austin, USA, pp. 149-152,

Holada, M.: The experiences and usability of distributed speech recognition system DUNDIS. In: *Proc. of 14th Czech-German Workshop „Speech Processing"*, September 2004, Prague, Czech Republic, pp. 159-162,

Hanika, J, Horak, P.: Text to Speech Control Protocol. In: *Proc of the Int. Conf. Eurospeech'99*, Budapest, Hungary, September 5-9, 1999, Vol. 5, pp. 2143-2146.

Šonka, M., Hlaváč, V., Boyle, R. D., *Image Processing, Analysis and Machine Vision.* PWS, Boston, USA, second edition, 1998.

DEVELOPMENT OF AN AUTOMATED DEVICE FOR SORTING SEEDS
Application on Sunflower Seeds

Vincent Muracciole, Patrick Plainchault
INRA;ESEO;LISA;GEVES-SNES, Université d'Angers, Angers, France
CER ESEO, Angers, France
vincent.muracciole@eseo.fr, patrick.plainchault@eseo.fr

Dominique Bertrand, Maria Rosaria Mannino
ENITIAA-INRA, Sensométrie - Chimiométrie, Nantes, France
GEVES-SNES, Beaucouzé, France
bertrand@enitiaa-nantes.fr, maria-rosaria.mannino@geves.fr

Keywords: Machine vision, computer vision, seed identification, grain quality, seed quality, embedded systems.

Abstract: Purity analysis and determination of other seeds by number are still made manually. It is a repetitive task based upon visual analysis. Our work objective is to create and use a simple and quick automated system to do this task. A first step of this machine has been reached by validating the image acquisition and feeding process. The principle of this machine is based on a seeds fall with stroboscopic effect image acquisition. This article presents the first step of creating a dedicated and autonomous machine which combines embedded constraints and real time processes.

1 INTRODUCTION

In most countries, seeds cannot be commercialized without an assessment of their quality. Quality control tests generally includes the evaluation of unwanted materials in the batches. This control is currently performed manually by operators able to separate the pure seeds from impurities.

In France, the official assessment of seeds is performed by GEVES (Groupe d'Etude et de contrle des Varits; Variety and seed study and control group). The standardized methods of the ISTA (International Seed Testing Association) and of the AOSA (Association of Official Seed Analysts) are applicable to two kinds of analyses of the seeds: purity analysis and determination of other seeds by number. In the analytical analysis of purity, all the elements in a sample (pure seeds (PS), inert matter (IM) and seeds of other plants (SOP)) are identified, separated and weighed. The result is then given as percentage of each part's weight.

In the determination of other seeds by number analysis, only the impurities are separated, identified and counted. The standardized methods of purity analysis are very requiring since a one hundred percent correct identification is expected. The visual inspection of seeds is time-consuming and needs operators to be trained. It is thus very desirable to develop an automaton making it possible to do such controls.

Some mechanical devices are commonly used to clean commercial seeds and separate them according to their qualitative nature. It seems possible to add an artificial vision system for completing the mechanical step. The artificial vision domain follows closely the quick growth of computer power. This evolution allows creating new applications which was not possible until today. (Jayas D.S and Bulley, 1999) (Moltó Enrique, 1998) (Egelberg P., 1994) (Pearson, 1994) (Wan, 2002) (Bennett, 2005)

Image analysis is not currently restrained to simple geometrical shapes and can be applied to complex objects such as biological materials.

All the work dealing with the identification of seeds proceed in two steps: extraction of features from a digital images of seeds, processing the features for building up a discriminant model. In most of the published studies, the proportions of correctly classified seeds reported by the authors hardly exceed 95 percent. Such figures are encouraging but probably not sufficient to allow the replacement of human operators by machines.

Most of the recent studies are based on the use of very complex discriminant methods involving neural or Bayesian approaches (Chtioui, 1997) (Granitto P.M. and Ceccatto, 2003) (Granitto Pablo M and Ceccatto, 2003) (Majumdar S., 2000b). These methods are often efficient but involve a very time-consuming training phase. The parameters of these predictive models (Kernels, neural weights) require to be stored in computer memories.

In a previous work, a vision system for the automatic acquisition of images of individual seeds has been designed and developed. Unlike other systems, our device is based on dedicated hardware. This dedicated hardware has some limitations. These limitations are directed by the embedded constraints: a limited code size implementation, a low memory size available, a limited binary mathematical operation, a low frequency compared to a PC processor, limited temperature dissipation and a limited integrated circuit size. Design and development of the vision system include these different constraints (Plainchault P., 2003). This is why it is not possible to implement very complex algorithms for discrimination into an embedded board and some choices have been made in the design and the development of system.

A linear discriminant analysis (LDA) has been chosen as a decision algorithm. LDA is applied on the features extracted from the individual image of seeds.

In the present work, the features include morphological, colour and texture characteristics. According to the application, the discriminating power of variables may change. Another way to reduce the algorithms complexity is to select a few predictive variables adapted to a given application. Moreover in practical applications, a large part of the sample can be accurately identified and it may remain only a small proportion in which the discrimination is more difficult. A relevant approach consists in using an automated system for a first screening and let the more difficult part of the sample to be manually analysed.

Instead of computers, we propose to use an FPGA (Field Programmable Gate Array). These hardware components allow the implementation of a dedicated architecture for vision chain. FPGA gives the possibility to apply massively parallel processes. We present below the chain of vision and the mechanism of image acquisition already in place. We finish by the presentation of the method and the results.

2 ARTIFICIAL VISION

The chain of artificial vision (Figure 1) includes several phases: acquisition, image processing, feature extraction and decision. Each element of this chain must be optimised in order to reduce computing time and increase efficiency.

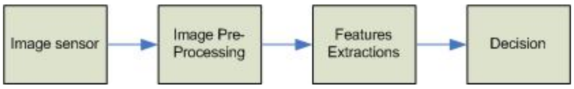


Figure 1: Vision chain.

2.1 Image Acquisition

Acquisition is a very important part of the performance system. It determines the quality of the image and the image processing time. A good image quality is without noise and without defect of illumination. These conditions are determinant for the image processing speed and the level of the results.

The quality depends on the choice of three principal types of sensor, on the choice of the lighting system and the selected background.

2.1.1 Sensor

The general principle of the sensors lies in the projection of a 3D scene on a 2D matrix where each cell carries out the summation of the photons reflected by the object into the scene.

- Mono CCD:
  A mono-CCD sensor is generally organized according to the Bayer mosaic. This matrix represents an alternation of separating cells of red, green and blue colours organized as shown in Figure 2. This matrix indeed leaves zones where the two others colours are not represented. In order to fill this absence, some methods of interpolation have been presented in this article (Gunturk Bahadir K., 2005) . Some methods can privilege certain features like anti-aliasing, shape enhancement or noise reducing.

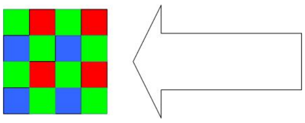


Figure 2: Mono CCD Sensor : Bayer Mosaic.

- Three CCD:
  Three ccd sensors do not need an interpolation phases. Colour separation is made by a prism which projects it onto three two-dimensional matrices, one red, one green, one blue as presented in Figure 3.
- Mono-chip Multichannel CCD:
  This sensor has been developed by the Foveon

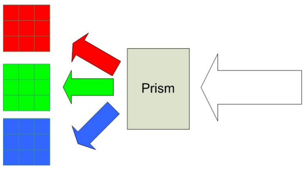


Figure 3: Three CCD Sensor.

company. It performs colour separation on the silicon. This sensor comes back to the photosensitive film principle as represented in Figure 4.

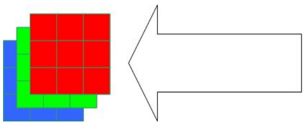


Figure 4: X3 Sensor.

2.1.2 Illumination Scene

Illumination is an important part to reduce image processing time. The more homogeneous illumination, we have. The quicker image processing, we have. In our case, a good separation between background and object is needed to maintain the real-time process. The illumination system can be composed of incandescent light, discharge lamps or light-emitting diodes. The choice of these systems is determined by the application and their characteristics.

2.1.3 Interpolation

The interpolation is a demosaicing phases for mono-CCD sensors. There are two methods to obtain images in three channels. The first method, is to reduce image size by calculating the average of green channel (1) and leave the red and blue channel as represented in the figure 5. The second is to use interpolation methods that can maintain sensor size.

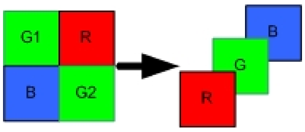


Figure 5: Representation of the image into the three channel.

$$\left\{ G = \frac{G1+G2}{2} \right\} \tag{1}$$

The requirements of embedded systems are elementary binary mathematical operators. For that reason, only bilinear and associated interpolation methods have been retained. These methods are totally independent and deterministic which respect the real time and parallel processing system objectives. These interpolations are reversible, avoid memory overloading and reduce image transfer time.

There are two possible methods : the bilinear interpolation represented in figure 6 and equation 2 and 5 (equation for red and blue channels are the same) and the constant-difference-based interpolation represented in Figure 7.

$$G22 = \frac{G12+G32+G21+G23}{4} \tag{2}$$

$$R22 = \frac{R11+R13+R31+R33}{4} \tag{3}$$

$$R12 = \frac{R11+R13}{2} \tag{4}$$

$$R21 = \frac{R11+R23}{2} \tag{5}$$

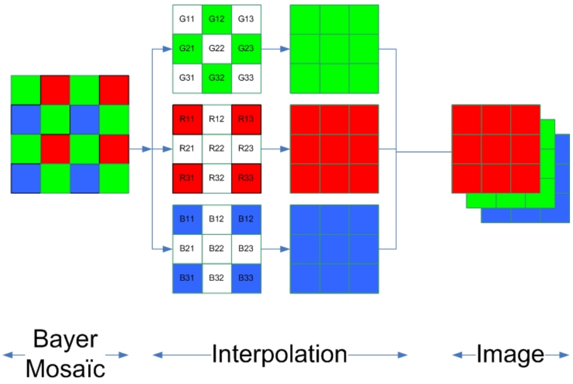


Figure 6: Bilinear Interpolation

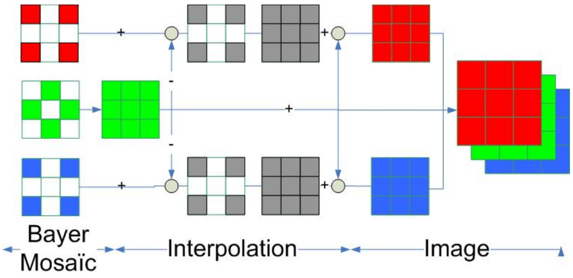


Figure 7: Constant-difference-base Interpolation.

2.1.4 Color Space

In general, the embedded systems operate using the sensor's RGB color space. Each channel in this space are strongly correlated. The choice of the colour space is dependant on the colour background. Generally a blue background is used for seed inspection. This background needs a new color space to perform seed extraction. The RGB space does not perform very well this process, so the conversion in the YCbCr (equation 6) space can be possible and is the best to perform this process. The choice of this space has been made for the following reasons:

1. It is a matrix based conversion
2. It is a reversible transformation (linear transformation)
3. The distance between seed and background is maximized

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.5 \\ 0.5 & -0.419 & -0.081 \end{bmatrix} * \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (6)$$

2.2 Feature Extraction

The extraction aims at defining object based on its morphology, its colour and its texture. For this extraction we need to extract the object and define his position, we use for that a segmentation task and a labelling task.

The segmentation algorithms used to perform the seed positionning task is a variance-based algorithm (Otsu, 1979).

The extracted features defining seeds are detailled by many publications. These features are summarized in (Majumdar S., 2000a) (Majumdar S., 2000b) (Majumdar S., 2000c) and in the thesis of Younes Chtioui (Chtioui, 1997). These features are around 110. There are three groups: morphological, colour and textural. In the morphological group, there is dimension, invariant moment and fast fourrier transform type. In the colour group, there is standard deviation, mean, variance. In the textural group, there is descriptor of texture, co-occurence, entropy, kurtosis etc...

2.3 Pattern Recognition: Linear Discriminant Analysis

Many identification methods (Chtioui, 1997) (Visen N.S., 2002) (Majumdar S., 2000d) in the domain of seeds are presented. Some of them cannot be retained because they do not answer the embedded constraints. This is why we privilege the simple methods based on the distances between groups and the methods of the regression type.

Some previous work show the feasibility of pattern recognition in FPGA based systems (Miteran J., 2005). An LDA method has been chosen. This method is based on the principle of the minimal distance to a group of a given seed as shown in figure 8 and the equation. It is the type of decision algorithm easily parallelisable, where each class are independent of its neighbours. These methods and algorithm has been associated to the Mahalanobis distance (equation 7) in order to increase the distance between classes. The x vector is the unknown seed to be classified. The μ vector is the mean value of a labelled group features. The Σ value is the covariance matrix for multivariate vector.

$$d^2(x) = (x-\mu)^T \Sigma^{-1} (x-\mu) \quad (7)$$

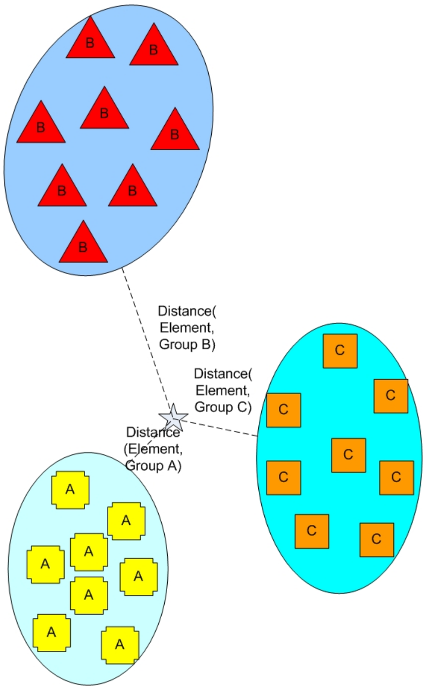


Figure 8: Principle of Linear Discriminant Analysis.

2.4 Hardware

The architectures presented in this section allow the creation of dedicated systems based on work from the computer world. They combine good computing efficiency with low power consumption, low memory and device monitoring constraints. These various points make them the ideal components for the embedded world.

2.4.1 MCU

Microcontrollers are circuits largely used for the management of the automated control processes. Although not intended for mathematical computation, the new evolutions of these components integrate more and more mathematical calculating units (example EP9315A from Cirrus Logic with mathematical coprocessor), and even cores DSP (example architecture OMAP Texas Instrument(TI)). However their performance is still limited for signal processing.

2.4.2 DSP

DSP are processors dedicated to signal processing. They combine the flexibility of a programmable processor with performance in real time signal processing. The new evolutions of these architectures saw the capacities of calculation increased to the detriment of consumption. However now they represent now a true general purpose processor for the embedded domain.

2.4.3 FPGA

FPGAs see their uses and their applicability grow. They combine the performances of a dedicated architecture and the flexibility of programming. FPGAs are composed of independent logical cells which have to be activated to create dedicated processes. Nowadays some signal and image processing are implemented into these components and represent new application for high efficiency and low cost systems.

3 MATERIALS AND METHOD

The automaton is based on a principle of free fall seed (Plainchault P., 2003). It acquires an image of each seed. It adds to it a stroboscopic effect breaking up the movement of fall. In the long term, the automaton must reach a sorting at the rate of one seed per second. The sorting in the broad sense covers acquisition, the image processing, the extraction of the parameters, the identification and, finally, the sorting at the same time.

3.1 Device

The automaton is composed of a sensor, a flash, a detection barrier, a background, an electronic system. Figure 9 shows this association.

3.1.1 Seed Sorting System

- Mechanical feeding:
  The mechanical feeding is based on an electromagnet which attracts a support heightened by blades of metal.
- Sensor :
  The sensor provides the frame grabbing function. It is a Kodak KAC-1310 sensor with a resolution of 1280*1024 and a clock frequency of 10MHz. It has the advantage of having a window activation authorizing the activation of only a part of the sensor. This function can increase speed of acquisition because this speed is proportional to the activated dimension.
- Flash Light :
  The flash is ensured by 8 diodes luxeon lumileds with a unit power of one Watt. This choice has been done for the response time and robustness of this component for the rate of one seed per second.
- Barrier :
  The detection barrier is based on infra red cells functioning in saturated mode. It determines the cut of the beam. For the moment, this barrier is able to treat only sunflower seeds, sunflower kernels, or sclerotia.
- Background :
  A blue background was selected because it allows a better segmentation of the object by using the YCbCr colour space and extent the seeds varieties
- Embedded Card :
  The electronic system lies on an embedded card. It is a sundance SMT 355 card composed of a TI DSP TMS320C32 and an ALTERA FPGA Flex 10K130E and 512 Kbytes of RAM. Its task is to control the peripherals and perform simple processing. Due to the architecture of the card, it is not simple to implement image processing into the FPGA. The FPGA does not have direct access to memory and the memory space is too low to implement image processing. But this card has tested and validates the feasibility of acquisition in free fall of object into embedded card.

3.1.2 Computer Validation of Embedded Algorithm

- Computer
  The electronic system also includes a computer. It is a DELL computer with an Intel Pentium IV 3GHz with HyperThreading processor, 1GB of DDR-SDRAM, a graphics board AMD/ATI X300 with dedicated memory and a 160GB hard disk. The computer provides the Human-Machine interface to control the automaton. It saves the image acquired in Bitmap format on the hard disk drive. The computer also plays a part in the algorithm validation. The identification algorithm

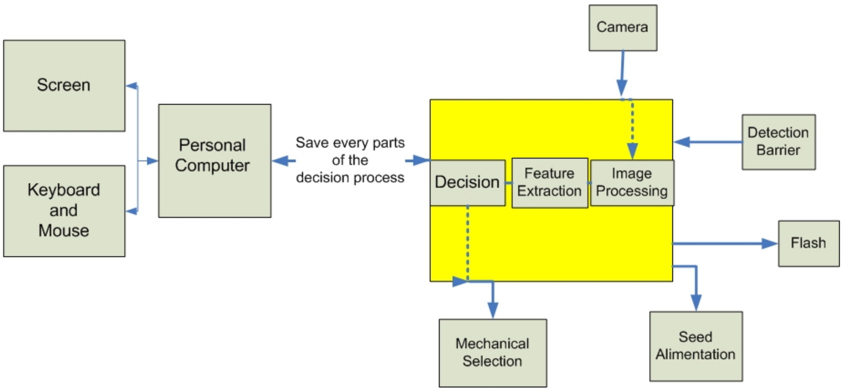


Figure 9: Presentation of the wanted system.

has been tested off line in order to determine its efficiency for the sorting.

- Software

  The automaton is controlled by a program implemented in the computer using Visual C++. It provides a window interface to control the acquisition process. The algorithm validation is carried out off line with the image acquired by the automaton under the environment of matrix algebra MATLAB 7. Firstly, we calculate the parameters for all the datasets. In order to have all the features algorithms we just use the bwmorph function from the matlab image processing toolbox. Secondly, we train and test the identification algorithm on the dataset to select the best algorithm.

3.2 Methods

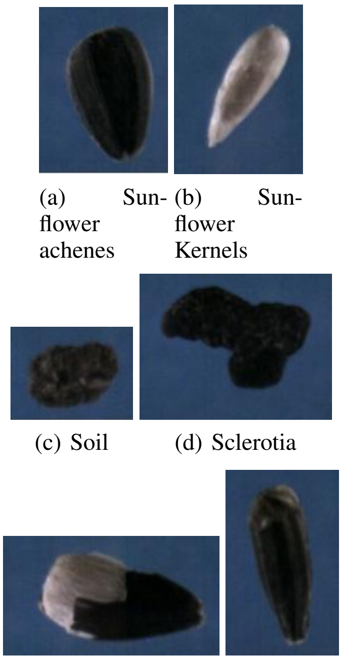
Seeds and impurities observed by the vision system were extracted from 23 samples of sunflower seeds representing commercial seed lots of various origins and 18 varieties, differing by seed colour and size. The dataset for impurities with a low frequency in the samples was enlarged using the reference collection of the GEVES-SNES. A dataset including 1051 images has been created. This dataset includes 6 classes: sunflowers kernels, broken seeds (fragments of seeds with a size higher than 50 percent of the size of the seed), mutilated seeds (fragments of seeds with a size lower or equal to 50 percent), sunflower achenes (intact seeds), sclerotia and soil.

At this stage of the study, our dataset includes all the categories (seeds and impurities) that have to be identified in official analysis of commercial seed lots (table 1) with the exception of seeds belonging to other species than the sunflower. Figure 10 shows colour, texture and morphology differences among the different classes.

This dataset has been labelled by an expert of the

Table 1: Principle of the analysis in the framework of purity and counting seed.

Object	Purity	counting
sunflower kernels	PS	
sunflower achenes	PS	
broken seeds	PS	
mutilated seeds	IM	
soil	IM	
sclerotia	IM	counting
Seeds of other plants	SOP	counting


(a) Sunflower achenes (b) Sunflower Kernels

(c) Soil (d) Sclerotia

(e) Broken seeds (f) Mutilated seed

Figure 10: Image of pure seeds (a,b,e) and impurities (c,d,f).

GEVES-SNES. With this dataset we have conducted a linear discriminant analysis using cross-validation. We have randomly selected 348 images for the training sets and 703 for the testing sets in accordance with table 2. We obtain a reduction of the feature space with a stepwise canonical discriminant analysis according to the article (Bertrand D., 1990). The selection of parameters is made incrementally by testing the increase of identification percentage.

Table 2: Dataset representation of each groups.

Object	training set	testing set	total
sunflower kernels	56	113	169
broken seeds	72	145	217
mutilated seeds	36	72	108
sclerotia	61	123	184
sunflower achenes	66	134	200
soil	57	116	173
total	348	703	1051

4 RESULTS

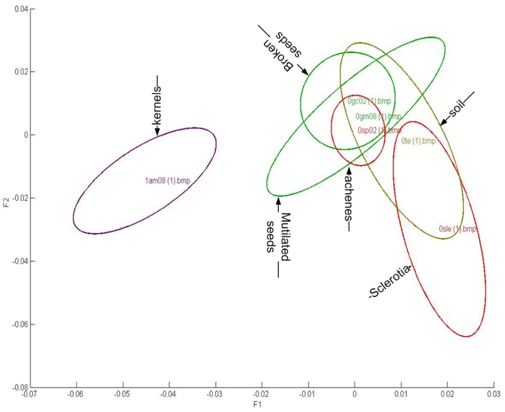
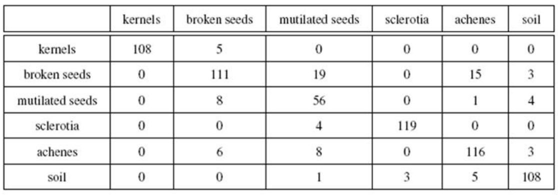


Figure 11: representation map of the sample.

The parameters selected by the LDA method show a great extent of texture (energy, entropy, occurrence), colour (mean, variance) and a few morphological (perimeter, area, moments, Fast Fourier Transform) parameters. Figure 11 shows the representation map of the sample and table 12 indicates the identification results for all the classes. A high percentage of each class has been correctly identified: from 75% of broken seeds to 97% of sclerotia. Some misclassification occurred, with the extent and importance that vary in relation to the classes. If we seek to identify independently each group, the selected features change but follow what we have said before in the part 3.2.1. The results expressed in table 12 is in accord with the map of labelled groups (figure 11). The table 12 shows all of the misclassifications identified on the map. Many of the errors of classification are due to a lack of information in the image. The free fall principle with only one camera has the same problem as on a rolling carpet (Egelberg P., 1994): it is possible that the camera does not take the image of the face bringing information, which is the case after analysis of the badly identified seeds. As we can note in table 12, there is a strong confusion between broken seeds and mutilated seeds. The algorithm can not perform a virtual representation of the associated pure seeds as the human expert can. From a practical point of view, we could accept a misclassification when it affects pure seeds (kernels, achenes and broken seeds) identified as impurities (e.g. sclerotia classified as soil). In this case, to achieve the analysis, the analyst will observe only a fraction of the sample classified as impurities by the system. Following this principle, the system classification was less satisfactory for mutilated seeds and soil: 13% of mutilated seeds were classified as pure seeds (broken seeds or achene), 4% of soil as achene. This first result shows the interest of the system when a determination of other seeds is carried out in sunflower: the algorithm makes possible 100% of correct separation of pure seed (achenes or kernels of sunflower) from sclerotia that are searched for in this kind of analysis.



	kernels	broken seeds	mutilated seeds	sclerotia	achenes	soil
kernels	108	5	0	0	0	0
broken seeds	0	111	19	0	15	3
mutilated seeds	0	8	56	0	1	4
sclerotia	0	0	4	119	0	0
achenes	0	6	8	0	116	3
soil	0	0	1	3	5	108

Figure 12: Differentiation sclerotia, achenes, broken seeds, mutilated seeds, soil.

5 CONCLUSION

The widening of the dataset of image seeds to new species is under acquisition.

The remaining work will consist of performing acquisition, features extractions and decision in a complete automaton. Optimisation and testing on other varieties of the decision algorithm have to be conducted.

Moreover the creation of a multi-camera system of acquisition according to the same principle would allow on certain ambiguities at the time of the phase of decision. It is also possible to make this approach more robust by creating a unknown class for seeds at equal distance from the various groups.

The testing on the identification algorithm on seeds of other plants has to be made. System enhancements have to be made like changing the infrared barrier in order to extend seed varieties acquisition. The design and the development of a new hardware system have to be made in order to implement a three camera systems.

ACKNOWLEDGEMENTS

This work was supported by the National Institut of Agronomical Research (INRA), the variety and seed study and control group (GEVES), the ESEO graduate school of engineering in electronic, and the LISA laboratory of the University of Angers.

For their financial support the region Pays de Loire and the CER Vegetal (French contract).

REFERENCES

Batlle J., M. J. (2002). A new fpga/dsp-based parallel. *Real time imaging*, 8:345–356.

Bennett, T. D. K. F. M. M. M. (2005). A computer-based system for seed identification. *Seed Technology*, 27:190–202.

Bertrand D., Coucroux P. Autran J.-C., M. R. R. P. (1990). Stepwise canonical discriminant analysis of continuous digitalized signals : applicatoin to chromatograms of wheat proteins. *Journal of chemometrics*, 4:413–427.

Chtioui, Y. (1997). *Reconnaissance automatique des semences par vision artificielle basée sur des approches statistiques et connexionnistes*. PhD thesis, IRESTE.

Davies, E. (2005). *Machine Vision - Theory, Algorithms, Practicalities*.

Egelberg P., Mansson O., P. C. (1994). Assessing cereal grain quality with a fully automated instrument using artificial network processing of digitized color video images. In *Forestry and Biological Processing*, volume 2345, pages 146–156. SPIE congress.

Granitto Pablo M, P. F. V. and Ceccatto, H. A. (2003). Automatic identification of weed seeds. *ASAI*.

Granitto P.M., P.A. Garralda, P. V. and Ceccatto, H. (2003). Boosting classifiers for weed seeds identification. *JCS&T*.

Gunturk Bahadir K., e. a. (2005). Demosaicking : Color filter array interpolation. *IEEE Signal Processing Magazine*, pages 44–54.

Jayas D.S, M. C. and Bulley, N. (1999). An automated seed presentation device for use in machine vision identification of grain. *Canadian agricultural engineering*, 41(2):113–118.

Majumdar S., J. D. S. (2000a). Classification of cereal grains using machine vision i. morphological models. *American Society of Agricultural Engineers*, 43:1669–1675.

Majumdar S., J. D. S. (2000b). Classification of cereal grains using machine vision ii. color models. *American Society of Agricultural Engineers*, 43:1677–1680.

Majumdar S., J. D. S. (2000c). Classification of cereal grains using machine vision iii. texture models. *American Society of Agricultural Engineers*, 43:1681–1687.

Majumdar S., J. D. S. (2000d). Classification of cereal grains using machine vision iv.combined morphology, color, and texture models. *American Society of Agricultural Engineers*, 43:1689–1694.

Miteran J., M. J. (2005). Automatic hardware implementation tool for a discrete adaboost-based decision algorithm : Prototyping for machine perception on a chip. *EURASIP*, 2005:1035–1046.

Moltó Enrique, Blasco José, B. J. V. (1998). Computer vision for automatic inspection of agricultural produces. In SPIE, editor, *Precision agriculture and biological quality*, volume 3543, pages 91–100.

Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9:62–66.

Pearson, T. (1994). Machine vision system for automated detection of stained pistachio nuts. In SPIE, editor, *Optics in Agriculture, Forestry, and Biological Processing*, volume 2345.

Plainchault P., Demilly D., F. A. T. S. B. D. (2003). Development of an imaging system for the high speed identification of seeds. *IEEE Sensor*.

Visen N.S., J. D. (2002). Specialist neural network for cereal grain classification. *Biosystems Engineering*, 82:151–159.

Wan, Y.-N. (2002). Automatic grain quality inspection with learning mechanism. *Information technology in Agriculture*.

MATHEMATICAL MODEL FOR WALKING ROBOT WITH SHAPE MEMORY ALLOY ANKLE

Anca Petrişor, Nicu George Bîzdoacă
Faculty of Electromechanical Engineering, University of Craiova, Romania
apetrisor@em.ucv.ro, nicu@robotics.ucv.ro

Daniela Roşca, Sonia Degeratu, Adrian Roşca, Raducu Petrisor
Faculty of Electromechanical Engineering, University of Craiova, Romania
drosca@em.ucv.ro, sdegeratu@em.ucv.ro

Keywords: Robotics, shape memory alloy applications, robotic ankle, walking robot, mathematical model.

Abstract: The paper presents a simultaneous force and length variation mode in shape memory alloy (SMA) robotic application. The robotic ankle contains four SMA actuators and a spherical articulation. In order to assure a high efficient robotic architecture, the mechanical and the control structure have to assure a real-time response to the work environment changes. The load variations or the difference between the moment of full contact step and the non contact moment for a waking robot are the standard situations for a SMA robotic ankle. The paper is divided in five sections. First section makes a short introduction in the physical description and conventional applications of shape memory alloy materials. Then, are presented the mathematical model for robotic ankle, the walking robot geometrical structure and the causality ordering of the active pair of legs, in this case with one free joint. In the last section some experimental results are presented. These results were obtained by using MATLAB programs, conceived by authors, for design and simulation of walking robots control algorithms.

1 INTRODUCTION

The shape memory effect was first noted over 50 years ago; it was not until 1962, however, with the discovery of a nickel titanium shape memory alloy by Buehler, that serious investigations were undertaken to understand the mechanism of the shape memory effect (Bîzdoacă and Pană, 2004), (Funakubo, 1987), (Graesser and Cozarelli, 1994), (Schroeder and Boller, 1993).

The shape memory alloys possess the ability to undergo shape change at low temperature and retain this deformation until they are heated, at which point they return to their original shape.

The nickel titanium alloys, used in the present research, generally refereed to as Nitinol, have compositions of approximately 50 atomic % Ni/ 50 atomic % Ti, with small additions of copper, iron, cobalt or chromium. The alloys are four times the cost of Cu-Zn-Al alloys, but it possesses several advantages as greater ductility, more recoverable motion, excellent corrosion resistance, stable transformation temperatures, high biocompatability and the ability to be electrically heated for shape recovery.

Shape memory actuators are considered to be low power actuators and such as compete with solenoids, bimetals and to some degree was motors. It is estimated that shape memory springs can provide over 100 times the work output of thermal bimetals.

The use of shape memory alloy can sometimes simplify a mechanism or device, reducing the overall number of parts, increasing reliability and therefore reducing associated quality costs.

Because of its high resistivity of 80 – 89 micro ohm-cm, nickel titanium can be self heated by passing an electrical current through it. The basic rule for electrical actuation is that the temperature of complete transformation to martensite M_f , of the actuator, must be well above the maximum ambient temperature expected (Delay and Chandrasekaran, 1987).

2 MATHEMATICAL MODEL OF SHAPE MEMORY ALLOY ANKLE

The robotic researches develop up to the present a various mechanical architecture for ankle structure. All projects use the human ankle as model.

The problem in developing efficient ankle structure, concern the dimension and the efficiency of actuators.

The proposed robot ankle structure contains units with SMA actuators. The unit has 4 SMA actuators and a spherical articulation.

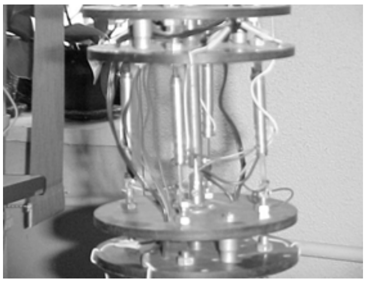


Figure 1: The proposed SMA robotic ankle.

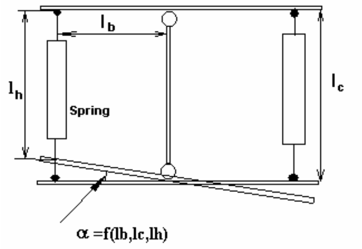


Figure 2: Schematically representation of SMA ankle.

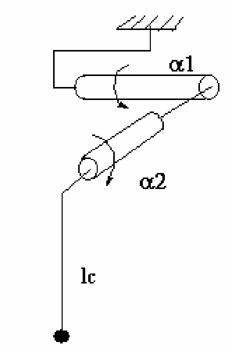


Figure 3: Kinematics representation of SMA ankle.

The mathematical model of the ankle is very simple:

$$H_{ankle}\left(\alpha_1,\alpha_2,l_c\right)=$$
$$\begin{matrix} \cos\alpha_2 & 0 & \sin\alpha_2 & l_c\sin\alpha_2 \\ \sin\alpha_1\sin\alpha_2 & \cos\alpha_1 & -\sin\alpha_1\cos\alpha_2 & -l_c\sin\alpha_1\cos\alpha_2 \\ -\cos\alpha_1\sin\alpha_2 & \sin\alpha_1 & \cos\alpha_1\cos\alpha_2 & l_c\cos\alpha_1\cos\alpha_2 \\ 0 & 0 & 0 & 1 \end{matrix} \qquad (1)$$

Analysing the angle dependence versus the SMA spring variation, a highly nonlinear function results:

$$\alpha=\arcsin\left[\cos\left(\mathrm{arctg}\left(l_h\right)\right)\frac{2l_b^2+l_c^2}{2l_b}\right]+\mathrm{arctg}\left(l_h\right) \qquad (2)$$

where l_h is the length after the heating process; l_c is the spring length after cooling and l_b the base length.

As the real variation is restricted (between 100% and 92 %), the linearization can occur because of linear behaviour for the specified evolution (Bîzdoacă and Pană, 2003).

3 WALKING ROBOT GEOMETRICAL STRUCTURE

It is considered the walking robot structure as depicted in Fig.4, 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= O^0, so the robot body RB is univocally characterized by the set,

$$RB=\{O^0,O^1,\lambda^i,\lambda^j,\lambda^p,\lambda^0\} \qquad (3)$$

where $\lambda^0=0$.

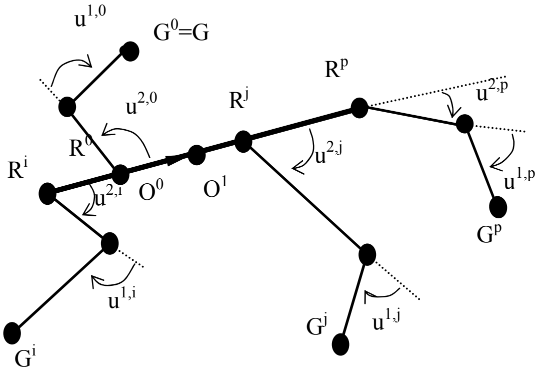


Figure 4: Geometrical structure of the robot.

The robot position in the vertical plane is defined by the pair of the position vectors O^0, O^1 where $|O^1 - O^0| = 1$, or by the vector O^0 and the scalar θ, the angular direction of the robot body.

Each of the four robot legs L^i, L^j, L^p, L^0 is characterized by a so-called Existence Relation ER(L) depending on specific variables as it is presented in (Petrişor, 2005), (Petrişor, Marin, Bîzdoacă and Cazacu, 2006).

The mathematical model of this object is a Variable Causality Dynamic Systems VCDS and it is analiyzed from this point of view.

A pair of legs $\{L_i, L_j\}$ constitutes the so called Active Pair of Legs (APL) if the robot body position is the same irrespective of the feet position of all the other legs different of L_i and L_j. A label is assigned to each possible APL. The APL label is expressed by a variable q called Index of Activity (IA), which can take N_a values, numbers or strings of characters. All the other legs that at a time instant do not belong to APL are called Passive Legs (PL). The leg in APL, having a free joining point is called slave leg, the opposite of the motor leg whose both joining points are external controlled.

4 CAUSALITY ORDERING OF AN ACTIVE PAIR OF LEGS WITH ONE FREE JOINT

In this structure, only one angle is free so three joints are external controlled (EC).It is denoted this by

$$q = 'ij', \; c = [\text{motor12}, \; \text{motor01}, \; c^p] \tag{4}$$

This causality ordering is corresponded to the state having the leg L^i as a motor leg which controls two degree of freedom and the leg L^j, a slave leg, which can control only one scalar component , so the angle $u^{2,j}$ is free and the angle $u^{1,j}$ is EC.

In this structure $u^{2,j}$ is obtained from the following relations:

$$\tilde{u}^{2,j}_{(t)} = \begin{cases} \psi_{2j} & \text{if } \; u^{2,j}(t-\varepsilon) > 0 \Leftrightarrow \hat{s}^j = \text{up} \\ -\psi_{2j} & \text{if } \; u^{2,j}(t-\varepsilon) \le 0 \Leftrightarrow \hat{s}^j = \text{down} \end{cases} \tag{5}$$

$$\tilde{u}^{2,j}_{(t)} = \begin{cases} -\varphi_{2j} + \psi_{2j} & \text{dacă } \; -\varphi_{2j} + \psi_{2j} \le 0 \\ -\varphi_{2j} + \psi_{2j} - \pi & \text{dacă } \; -\varphi_{2j} + \psi_{2j} > 0 \end{cases} \tag{6}$$

$$\tilde{u}^{2,j}_{(t)} = \begin{cases} -\varphi_{2j} + \psi_{2j} + \pi & \text{dacă } \; -\varphi_{2j} + \psi_{2j} \le 0 \\ -\varphi_{2j} + \psi_{2j} & \text{dacă } \; -\varphi_{2j} + \psi_{2j} > 0 \end{cases} \tag{7}$$

Therefore, in this causality structure the kinematics restriction $|R^j - R^i| = r_{ij}$ is accomplished by changing the value of $u^{2,j}$ at $\tilde{u}^{2,j}$ giving by the equation (5), (6), (7).

5 EXPERIMENTAL RESULTS

The causal ordering c=[motor12, motor 01, c^p] is implemented together with other causal orderings, in the RoPa platform for design and simulation of walking robots control algorithms. The RoPa platform is a complex of Matlab programs to analyze and design walking robots, evolving in uncertain environments according to a new control approach called Stable State Transition Approach (SSTA), also conceived by the authors.

The causal ordering developed in this paper is activated by selecting the causal variable cz=[12 1 0]. The RoPa platform allows animation, recording of the evolutions and playback them.

In the following there are presented some experimental results of walking robot behaviour considering this causal ordering.

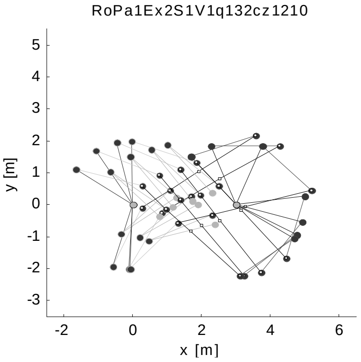


Figure 5: The robot kinematics evolution.

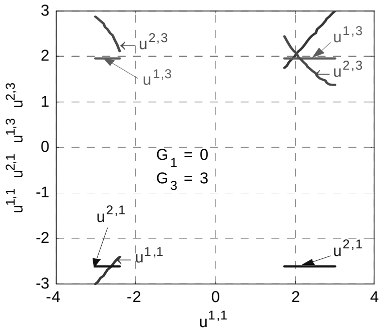


Figure 6: Controlled angles with respect to the input angle.

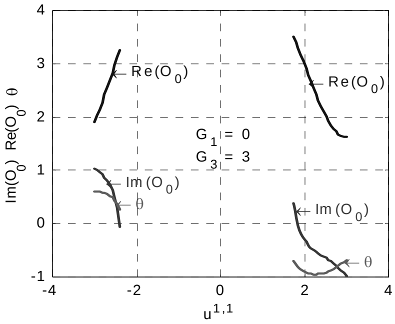


Figure 7: Robot body position with respect to the input angle.

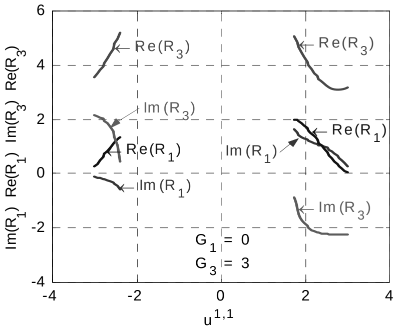


Figure 8: Joints positions with respect to the input angle.

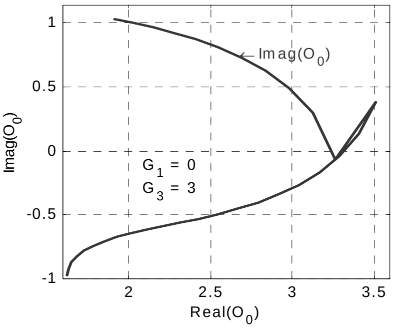


Figure 9: Reference point O^0 locus in evolution scene.

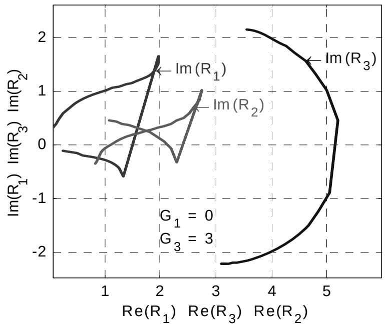


Figure 10: Joints points' locus in evolution scene.

6 CONCLUSIONS

Shape memory alloy ankle add to the walking robot architecture additional control possibility for adapting the structure to various environment.

The causal ordering developed in the paper is useful when it must control the shoulder position R_i irrespective of the other variables. It can be controlled by the angle $u^{1,j}$, the shoulder angle of the second active leg L_j.

The approach of the kinematics structure by complex number representation of the variables allowed to solve the equations system that describe the position of the variables that are involved, between witch there are kinematics correlations. In this way, it is obtained an explicit representation of the input-output dependence, in this causality structure.

As it can be seen from the above experimental results, this causal ordering is perfectly integrated in the RoPa structure proving the correctness of the theoretical results.

The mathematical model developed in the paper becomes an element of the VCDS walking robot model. The simulations used MATLAB environment show the robustness of the mathematical model.

REFERENCES

Bîzdoacă, N., Pană D., 2003. Shape memory alloy tendon driven finger. *In Proceedings of the 14th international conference on control systems and computer science, CSCS15*, Bucharest, Romania, May 25-27.

Bîzdoacă, N., Pană D., 2004. Shape Memory Alloy Based Robotic Ankle. *In Proceedings of the 5th International Carpathian Control Conference, ICCC 2004,* Vol. I.

Delay, L., Chandrasekaran M., 1987. *Les Editions Physique*. Les Ulis.

Funakubo, H., 1987. *Shape Memory Alloys*. Gordon and Breach Science Publishers.

Graesser, E.J., Cozarelli F.A., 1994. *Journal of Intelligent Material Systems and Structures*, no.5.

Petrişor, A., 2005. Walking mouvement with controlled style of the passive leg. *In Proceedings of the 15th international conference on control systems and computer science, CSCS15*, Bucharest, Romania, May 25-27, 2005, pag.483-487.

Petrişor, A., Marin, C., Bîzdoacă, N., Cazacu, D. 2006. Variable Causality Dynamic System Approach for Robot Control. *WSEAS Transactions on Systems and Control Journal*, Issue 1, Volume 1, Nov., pp.26-30

Schroeder, B., Boller, Ch., 1998. *Comparative Assessment of Models for Describing the Constitutive Behaviour of Shape Memory Alloy, Smart Materials and Structures.*

WEB-BASED INTERACTIVE POSITIONING CONTROL OF AN ELECTRIC FLATCAR VIA WIRELESS NETWORK

Ken Ishii, Koki Abe and Yoshimi Takao

Natl. Res. Inst. of Fisheries Engineering, Fisheries Research Agency,
7620-7, Hasaki, Kamisu city, Ibaraki 314-0408, Japan
kenmaru@affrc.go.jp

Keywords: Electric flatcar, manipulator probe, model of WEB direct-access monitoring, double exclusive control, controller and multiple-monitor system.

Abstract: A large tank has been used for target strength pattern measurements of fish. It is equipped with an electric flatcar. Further an elevation-rotating unit runs on the rails above it. The probe on the top of its elevation unit is equipped with an attachment for an ultrasonic transducer. The manipulator probe is movable in the four directions of the x, y, z and θ axes. Installation of a remote control switch has been required for the purpose of efficient operation of an electric flatcar. A remote control system using a notebook personal computer has been developed with good cost performance. The PC is equipped with a wireless network interface card. A model of WEB direct-access monitoring has been designed newly on the basis of the concept that an operator can control a mechanical device using a WEB Browser via LAN. Furthermore it performs double exclusive control for access from multi PCs, and has made possible a controller and multiple-monitor system. The mission was performed for the purpose of evaluation of WEB operation. The result has made clear the specifications for motion, and an external interface of the electric flatcar is applicable to the new protocol developed for WEB Browser control.

1 INTRODUCTION

An indoor tank filled with fresh water has been used for target strength pattern measurements of fish (Ishii and Takao, 2000), and its shape is 10(W) x 15(L) x 10(H) m^3. It is equipped with an electric flatcar. Further an elevation-rotating unit runs on the rails above it. The probe on the top of its elevation unit is equipped with an attachment for an ultrasonic transducer as shown in Figure 1 (Ishii et al., 1995). Installation of a remote control switch has been required for the purpose of efficient operation of an electric flatcar. A remote control system by a WEB Browser using a personal computer (PC) has been developed with good cost performance. Generally, WEB Browser displays a page from a web sever responding to a request of a client. A long reply time is not applicable to WEB Browser. But, the reply time of a target mechanical device accessed by a web server varies widely. The issue is development of a protocol for receiving reply messages from a target machine. The communication method of this system is described in detail in this paper.

2 METHOD

The probe can move four-dimensionally. The nominal precision is ±1 cm along the longitudinal and transverse axes, ±2 cm along the elevation axis, and ±0.2 degree around the rotation axis. Resolutions of absolute encoders for positioning are 1 mm and 0.01 degree, respectively. Parameters of velocity are limited between 1 and 20 cm/s along the longitudinal axis, 1 and 10 cm/s along the transverse axis, and 0.2 and 3.0 degree/s around the rotation axis.

The sequencer (PLC, Programmable Logic Controller) directly controls the electric flatcar every 10ms. Interval of communication between the PC and the sequencer is one second using DDE (Dynamic Data Exchange) protocol via P-LINK (Processor Link) (Ishii et al., 1999).

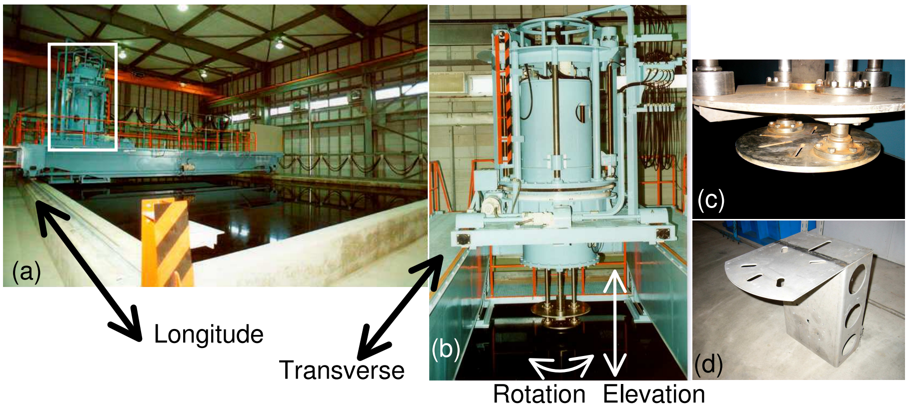


Figure 1: The electric flatcar and its movable directions shown by arrows. (a) The electric flatcar on a large tank. (b) An elevation-rotating unit on it includes a movable probe. Two devices of (c) and (d) are used in the joint mission. (c) An attachment on the top of a probe in the electric flatcar. (d) An adaptor for a transducer.

The function of a remote control switch has been realized by a notebook PC equipped with a wireless network interface card that is implemented more easily than a specified low power radio wave unit. The model of WEB direct-access monitoring has been designed newly on the basis of the concept that multiple users control or monitor a mechanical device using a WEB Browser via LAN (Local Area Network). It consists of a main server program that performs exclusive control for access from multi PCs, and makes possible a controller and multiple-monitor system (Ishii and Sawada, 2005). Figure 2 shows its application to the remote control switch function for an electric flatcar. The driving control command through the main server program does not access the database that holds the values, but accesses the device driver program that operates the instrument instead.

The mission was performed for the purpose of evaluation of WEB operation. The mission is achieved by bringing an attachment of a probe close to an adaptor for a transducer shown in Figure 1. In this mission, the speed is one or two cm/s in the transverse direction, one cm/s in the elevation direction, and 0.2 degree/s in the rotation direction (Ishii et al., 2006).

3 RESULTS

A WEB Browser display of a remote control switch function has dual modes of ”DRIVE” and ”JOG”. Furthermore, each mode consists of two independent frames of ”SUBMIT COMMAND” and ”POLLING”. An example of ”DRIVE” mode is shown in Figure 3. The time chart is the following:

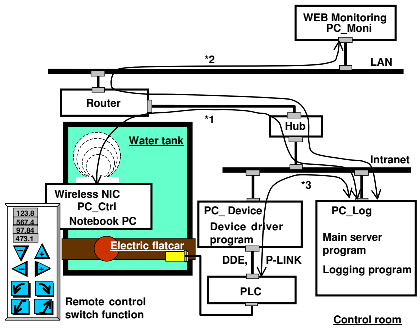
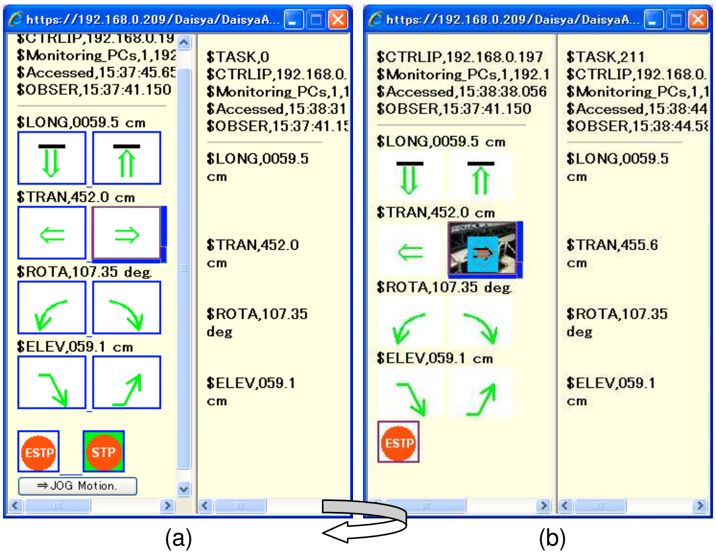


Figure 2: The model of WEB direct access monitoring applied to a remote control switch function. Widely located WEB Browsers post request commands to the main server program respectively via the routes *1 or *2 on a network. The main server program directly accesses, via route *3, a device driver program of an electric flatcar in order of a session.

- A negotiation command for priority was submitted from the controller PC at a time of 15:38:38.040, and a transverse driving command was submitted at 15:38:38.056. The left command frame was shifted to Figure 3-(b) from Figure 3-(a).
- During movement, polling is repeated by a monitoring frame.
- After a submission of a STOP command at 15:38:51.415, an end flag of driving was detected at a time of 15:38:52.087.
- By control from a right polling frame, a page of left command frame was renewal to a waiting page of Figure 3-(a) at a time of 15:38:52.368.

Among the mission for evaluation, access of a PLC from a remote control switch and monitor PCs



```
https://192.168.0.209/Daisya/DaisyaA...
$CTRLIP,192.168.0.19
$Monitoring_PCs,1,192
$Accessed,15:37:45.65
$OBSER,15:37:41.150
$LONG,0059.5 cm
$TRAN,452.0 cm
$ROTA,107.35 deg
$ELEV,059.1 cm

$TASK,0
$CTRLIP,192.168.0.
$Monitoring_PCs,1,1
$Accessed,15:38:31
$OBSER,15:37:41.15
$LONG,0059.5
cm
$TRAN,452.0
cm
$ROTA,107.35
deg
$ELEV,059.1
cm
```

```
https://192.168.0.209/Daisya/DaisyaA...
$CTRLIP,192.168.0.197
$Monitoring_PCs,1,192.1
$Accessed,15:38:38.056
$OBSER,15:37:41.150
$LONG,0059.5 cm
$TRAN,452.0 cm
$ROTA,107.35 deg
$ELEV,059.1 cm

$TASK,211
$CTRLIP,192.168.0.
$Monitoring_PCs,1,1
$Accessed,15:38:44
$OBSER,15:38:44.58
$LONG,0059.5
cm
$TRAN,455.6
cm
$ROTA,107.35
deg
$ELEV,059.1
cm
```

Figure 3: Design of the DRIVE mode with START/STOP. From upper icon, control buttons show the four directions ”longitudinal”, ”transverse”, ”rotation”, and ”elevation”. A driven direction is selected by a tab key or a mouse. The left figure (a) shows a mode of WAITING, and a foreward direction of ”transverse” is focused. The right figure (b) shows a status of driving operation.

was performed via Intranet or LAN as shown in Figure 4 for a typical example. In this figure, the X-axis shows time in seconds. ”PC_Ctrl” of the Y-axis works as a controller with a remote control switch function. A WEB Browser performed polling in intervals of five seconds for the monitoring, or two seconds for the driving. Each access to a main server is classified by a session ID, and is ordered by double exclusive control. A transverse driving command ($DRIVE) was submitted from the controller PC (named PC_Ctrl) at a time of 17:35:38.571, and a stopping command ($STOP) was submitted at 17:35:56.096. Positioning information of the electric flatcar is given per access. On the other hand, a logging PC (named PC_Log) receives renewal data at an interval of about one second linked with changing of values of position. The probe of the electric flatcar moved from 384.6 cm to 370 cm at the minimum velocity of one cm/s along the transverse axis. It has been confirmed that a WEB Browser on the controller PC is available to be performed in repetition of one second. Furthermore, the number of PCs for WEB monitoring has been confirmed normally for Four PCs.

Three kinds of commands such as a negotiation command for priority, driving commands and a stopping command, are submitted sequentially from a command frame of a controller PC. On the other hand, in a monitoring frame, polling is performed for the purpose of acquisition of positioning information. Averaged reply times for drive commands are shown in Table 1 in addition to needed times for acquisition of moved position information.

4 DISCUSSIONS

Dual modes of motion operation are available to be changed to the other interactively. The DRIVE mode with START/STOP is used for driving of long distance. The JOG mode is useful for short range driving, about one cm. And Driving of a fixed distance such as JOG mode is needed for safety of operations.

In Figure 4-(b), it is shown that access timing of multiple PCs (PC_Moni and PC_Ctrl) was successfully ordered by double exclusive control of a main server program.

In Table 1, it takes several seconds for acquisition of first positioning data owing to lower initial speed. After second positioning data, it takes about one second. It takes about six ms for negotiation of priority. Positioning data is obtained at the same time of negotiation of priority. And in the monitoring frame of a WEB Browser, a negotiation command also works as a polling command for receiving a reply message from the mechanical device (PLC). A reply to a monitoring command is performed immediately. The return time of driving commands is about 30ms.

Furthermore, submission of a driving command hardly coincides with the renewal of a WEB page for polling, because of the two independent frame

Table 1: Reply times for commands and needed times for position information.

Drive axis	Priority	$DRIVE	First position	Priority	$STOP	Last position
	ms	ms	sec.	ms	ms	sec.
Transverse	6	32	2.5	6	31	1.6
Elevation	6	28	1.5	7	28	1.0
Rotation	7	30	3.2	7	27	1.2

architecture. Therefore retry of the selection is not required. Consequently remote control by a WEB Browser has been working well.

Residual issues are the following:

- A notebook PC lacks solidity.
- It takes longer time in a FORCED STOP operation on a wireless PC than in an operation by an EMERGENCY STOP button on an operation desk.

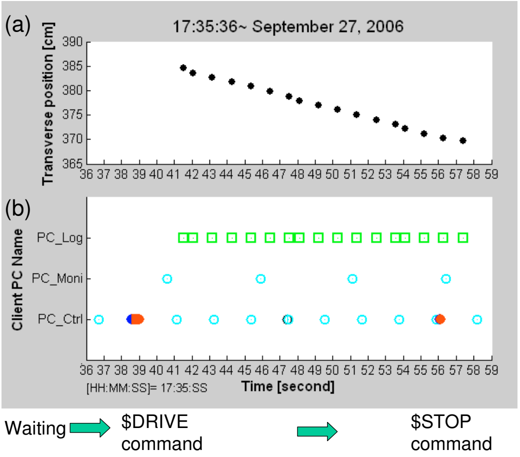


Figure 4: Ordered accesses by three PCs to a main server PC via LAN. The lower figure (b) shows access timing from multi PCs. PC_Mon and PC_Ctrl use WEB Brower with an interval time of two or five seconds. PC_Log uses an exclusive application program for logging positioning data with an interval time of about one second. The upper figure (a) shows positioning information acquisitioned by PC_Log. In the lowest row, a transition of motion modes about PC_Ctrl is shown.

5 CONCLUSION

The development of a remote operation system via a wireless PC didn't require editing of a complex sequencer program in the electric flatcar, so that this system starts working well as soon as possible. Using a function of PRESET DRIVE and a simple communication protocol that are implemented in the electric flatcar, the server program for WEB pages that are available for control and monitoring of a machine has developed in a sever PC. Usage of a WEB browser makes possible a flexible constitution of the remote terminal system. Then it has a merit of a smaller task to a terminal PC.

This system has been developed on the basis of two contents such as a WEB direct-access model, and a controller / multi-monitor system model. It was clear by the above-described mission that the communication speed of the electric flatcar is applicable to the model of WEB direct-access control and monitoring, in the case of attainability of the electric flatcar.

An attachment of a probe is also available for installation of underwater cameras. The optical calibration of an underwater observation bench is scheduled using an electric flatcar. It is hopeful that the wireless operation on the spot instead of a push button operation on an operation desk is effective.

REFERENCES

Ishii, K., Abe, K., and Takao, Y. (2006). Positioning control of the electric flatcar using web browser via wireless lan. In *Abstracts of 2006FY Autumn Annual Meeting*. AMSTEC.

Ishii, K. and Sawada, K. (2005). Network-integrated control program for a transducer-rotating mechanism to measure target strength (ts) pattern of squid. In *Journal of Advanced Marine Science and Technology Society*. AMSTEC.

Ishii, K., Sawada, K., and Miyanohana, Y. (1999). Design of transducer-positioning system scanning around an aquatic target in a tank. In *Collected Papers of FORUM ACUSTICUM BERLIN 99*. ASA and DEGA.

Ishii, K., Sawada, K., Takao, Y., Miyanohana, Y., and Okumura, T. (1995). Control system of electric flat-car equipped above an acoustic tank. In *Proceeding of the Annual Meeting*. the Marine Acoustics Society of Japan.

Ishii, K. and Takao, Y. (2000). Evaluation of transducer-positioning system scanning around an aquatic target in a tank. In *Abstracts of 2000FY Spring Annual Meeting*. Advanced Marine Science and Technology Society.

THEORY OF STRUCTURED INTELLIGENCE
Results on Innovation-based and Experience-based Behaviour

Meike Jipp and Essameddin Badreddin
Automation Laboratory, University of Mannheim, B6, 23-29, Mannheim, Germany
mjipp@rumms.uni-mannheim.de, badreddin@ti.uni-mannheim.de

Keywords: Intelligence, intelligent interfaces, innovation, human computer interaction.

Abstract: An agreed-upon general theory of intelligence would enable significant scientific progress in all disciplines doing research on intelligence. Such a theory, namely the theory of structured intelligence is tested by relating it to other theories in the field and by empirically testing it. The results demonstrate (1) that the theory of structured intelligence uses a similar concept of intelligence as do other theories but offers greater scientific insights in the how intelligent behaviour emerges and (2) that its distinction between innovation- and experience-based solutions can be found in the behaviour of the study's participants. This yields the opportunity to (1) allow technically testing intelligence in an easier and less time-consuming ways as do traditional intelligence tests, and (2) allow technology classifying the intelligence of its user and using adaptive interfaces reducing the possibility of serious handling errors.

1 INTRODUCTION

Many different theories of intelligence have been developed in the varying disciplines (for a summary see Badreddin & Jipp, 2006). So far, these approaches have been isolated not sufficiently taking into account results from other disciplines. An opposite example is the theory of structured intelligence developed by Badreddin and Jipp (2006). The authors make use of research results in neurophysiology, psychology and system theory and present the theory itself and discuss ways of technical implementation. The theory defines intelligence as the ability to solve problems using limited space and time resources. The concepts of Innovation, Experience, Fusion, and Learning are distinguished to explain problem solving behavior. *Innovation* reflects the capability to come up with totally new, unpredictable solutions to the current problem. *Experience* refers to using past, known, and successful solutions for the current problem. If a problem is faced, two solutions are worked out, one based on innovation ("new solution"), the other based on experience ("past solution"). These two solutions are fused by appropriate algorithms and the final solution will be applied to the current problem. This derived solution is saved, so that it is available the next time the same or a similar problem is faced. Hence, any combination of new and well-known solutions to a problem can be developed. Fig. 1 gives an overview over the described structure.

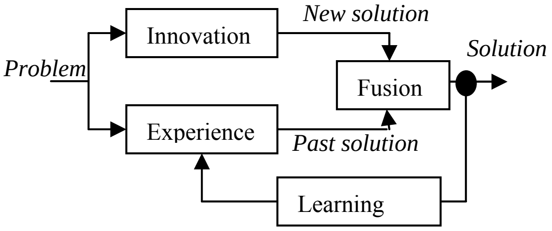


Figure 1: Theory of structured intelligence (see Badreddin & Jipp, 2006).

2 PROBLEM STATEMENT

The, by Badreddin and Jipp (2006) developed theory of structured intelligence needs further testing to see whether it (1) allows the realization of a qualitatively different concept of artificial intelligence, (2) allows easier testing of human intelligence, (3) enables technology to test their user's intelligence and adapt interfaces according to the level and structure of intelligence to avoid possible errors, which is especially of importance in safety-critical systems.

3 SOLUTION APPROACH

To test the theory at hand, a way similar to what is known as construct validation has been taken (Campbell & Fiske, 1959): First, the concept of intelligence as described in Section 1 will be set into relation with other existing theories (see Section 2.1). To empirically test the defined relations and related hypotheses, a study has been conducted in a second step, which results are presented in Section 2.2.

3.1 Theoretical Foundations

In order to perform the construct validation, the theory of structured intelligence must be put in theoretical relation with another theory which has proven empirical adequacy. The theory used here is the theory of skill acquisition, which was developed by Ackerman (1988) and is based on research performed by Fitts (1964), Anderson (1982), Fleishman (1972), as well as Schneider and Shiffrin (1977). Ackerman (1988) distinguishes three phases of skill acquisition:

- The first phase is characterized by a relatively strong demand on the cognitive-attentional system, so that performance is slow and error prone. Ackerman (1988) explains this phase as the one in which potential ways for executing the current task are worked out and (mentally) tested. Attention is focused on thoroughly understanding the task's constraints in question. With consistent practice, performance gets faster (see Schneider & Shiffrin, 1977) and attentional demands are reduced (see Fisk & Schneider, 1983).
- During the second phase, the applied and successful ways of executing the task in question are strengthened and fine-tuned. More efficient ways of solving the task in question are found.
- Finally, performance is fast and accurate. The task is automated and can be completed without much attention.

Performance in each of these three phases is determined by abilities, namely by general intelligence, perceptual speed ability, and psychomotor abilities. *General intelligence* was defined by Ackerman (1989) in accordance to Humphreys (1979), as the ability to acquire, store, retrieve, combine, compare, and use information in new contexts. *Perceptual speed* refers to the ability to complete very easy cognitive tasks. The core cognitive activity is to generate very simple potential solutions to effectively solve tasks as quickly as possible. The key is the speed with which symbols can be consistently encoded and compared (Ackerman, 1989). Last, *psychomotor abilities* represent individual differences in the speed of motor responses to problems without information processing demands.

Ackerman (1988) proposes that general intelligence determines initial performance on a task with new information processing demands, i.e. the first phase of skill acquisition. The influence of general intelligence diminishes, when potential ways for the solution have been formulated (for empirical support, see e.g., Ackerman, 1988). The learner proceeds to the second phase of the skill acquisition process, when an adequate cognitive representation of the task has been built. Then, performance depends more on psychosensoric abilities. It is required to fine-tune and compile the determined solutions, which equals the definition of the abilities underlying psychosensoric abilities. Sequences of cognitive and motor processes get integrated, ways of solution adapted for successful task performance. With further practice the impact of psychosensoric abilities on performance decreases and psychomotor abilities play a more important role. In this third phase of the skill acquisition process, the skill has been automated, so that performance is only limited by psychomotor speed and accuracy (Ackerman, 1988).

Fig. 2 summarizes the described relationship between skill acquisition and ability-performance correlations.

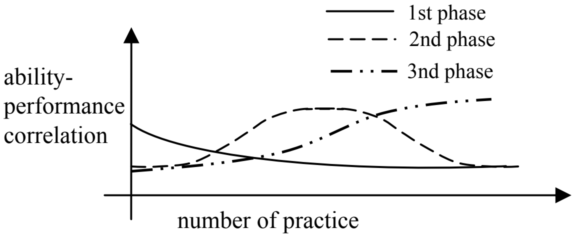


Figure 2: Theory of Skill Acquisition (adapted from Ackerman, 1989).

Taken into account the above described theory of structured intelligence it is to be assumed that in the first phase of skill acquisition, in which the learner is confronted the first time with a new problem, innovation processes affect the derived solution. In contrast, the solution to a well-known problem will be chosen based on the memory traces of the already successfully applied solution alternatives. Hence, in this case, behaviour is experience-based.

3.2 Conducted Study and Relevant Research Results

3.2.1 Research Questions and Hypotheses

The, in the previous section described theory of skill acquisition has been tested by various researchers (see e.g., Jipp, Pott, Wagner, Badreddin, & Wittmann, 2004) and proved its empirical adequacy in various settings and circumstances. The relationship to the theory of structured intelligence has been established theoretically in Section 2.1 and will be tested empirically. Therefore, the following research questions and hypotheses are of interest:

- It is hypothesized that the degree of consistency with which given tasks are tackled decreases with the familiarity of the task. This is the case as formerly applied successful solutions will be used to solve the problem at hand. Variation does only occur when the former solution has not lead to a satisfying result.
- It is hypothesized that the transition from innovation-related processes to experience-based solutions is determined by intelligence factors as measured with traditional measurement scales for intelligence factors. This is based on Ackerman's theory (1989) combining skill acquisition processes with factors of individual differences, i.e., general intelligence, perceptual ability and motor skills. The last is not considered in this paper due to the focus on structured intelligence.

3.2.2 Description of the Sample and Course of the Study

To be able to answer these research questions, a study has been conducted at the vocational school of the *Protestant Foundation Volmarstein* (Evangelische Stiftung Volmarstein, Germany). Data from 13 students (6 male, 7 female students) was at hand for the present analyses. The students were wheelchair-users and have been disabled for more than 12-15 years. Their average age was 22.5 years (SD = 1.6 years). The disabilities of the participants were spasticity, spina bifida, dysmelia or incomplete paralysis.

The study was conducted within two sessions. The first session lasted between one and two hours depending on the speed with which the participants performed the designated tasks (see also Jipp, Bartolein, & Badreddin, 2007). The tasks the participants conducted referred to leading a little garden market. More specifically, the participants had to prepare products potential customers requested. These customer wishes were sorted in two categories: sowing seeds (either sunflower or ramson seeds) and setting in seedlings (either flowering or foliage plants). The following actions were required in order to sow the seeds:

- The pots had to be placed in a seed box.
- The pots had to be filled with loosened soil.
- A hole had to be made into the soil.
- One seed had to be put in each hole.
- If the seeds were light seeds (as indicated on the customer wish), the holes had to be covered with wet pieces of newspaper.
- If the seeds were dark seeds (as indicated on the customer wish), the holes had to be covered with a 0.5 cm level of soil.
- The pots had to be watered. The water had to be prepared so that it had a temperature of 25°C and a, in the instructions specified acid value.

For setting in the seedlings, the following actions had to be performed by the participants:

- The required pots had to be filled half with soil, which had to be loosened before.
- The seedlings had to be put into the pot.
- The correct fertilizer had to be chosen (as indicated on the instructions, which were handed out to the participant).
- The pot had to be filled with layers of soil and fertilizer until the roots of the seedlings were covered.
- The seedling had to be watered with appropriate water (25°C and an acid value of 5-6).

In order to acquire the task of leading the market garden, four customer requirements had to be executed: the first required the participants to sow sunflower seeds, the second to set in flowering seedlings, the third to set in foliage plants and the last one to sow ramson seeds. The two categories of tasks have been defined based on only minor differences in order to allow the participants to acquire the skill in question. Further customer wishes could not be executed by the participants due to problems related to maintaining attention for such a long time frame. The actions of the participants were filmed with a standard web camera.

In the second session of the study, the participants performed tasks of the Berlin Intelligence Structure Test (BIS, Jäger, Süß & Beauducel, 1997). These tasks were based on the Berlin Intelligence Structure Model (Jäger, 1982), which is a hierarchical model of intelligence. General intelligence, at the top, is composed of two facets, which are categories for factors at the next lower level (Guttman, 1954). Jäger (1982) distinguished the facet *operations* and *contents*. The last subsumes three content abilities (i.e., numerical abilities, verbal abilities, and numerical abilities), which refer to how a person cognitively deals with the different types of contents. The facet operation subsumes what is cognitively done with the given contents. Four operations are distinguished: *Reasoning* is the ability to solve complex problems (Jäger, Süß, and Beauducel, 1997). *Memory* asks the participants to memorize pieces of information and retrieve them from short-term memory or recognize them after a short time period. *Creativity* refers to the ability to produce a variety of differing ideas controlled by a given item. Last, *perceptual speed* is the ability to work as fast as possible on simple tasks, requiring no or only little information processing demands. The BIS tests all these factors of intelligence. However, the original test has been shortened. Test items were deleted which required the participants to write a lot, as – due to the given time constraints for working on the test items – especially participants with spacticity would have been disadvantaged. The conducted test comprised the tasks as indicated in Table 1 and took the participants about two hours to complete.

Table 1: The, from the BIS chosen and in this study applied test items and their sorting in the factors of intelligence according to the Berlin Intelligence Structure Model.

General Intell-igence	*Figural abilities*	*Verbal abilities*	*Numerical abilities*
Perce-ptual speed	- Erasing letters - Old English - Number Symbol Test	- Part-Whole - Classi-fying words - Incomplete records	- X-Greater - Calcul-ating characters
Memory	- Test of orientation - Company's symbols - Remem-bering routes	- Mean-ingful text - Remem-bering words - Language of fantasy	- Pairs of numbers - Two-digit numbers
Reason-ing	- Analogies - Charkow - Bongard - Winding	- Word analogies - Fact opinion - Comparing conclusions	- Reading tables - Arithmetic thinking - Arrays of letters

3.2.3 Data Analyses and Results

The following variables were derived:

- general intelligence, perceptual speed, reasoning, memory, figural abilities, verbal abilities, numerical abilities, figural perceptual speed, verbal perceptual speed, numerical perceptual speed, figural reasoning, verbal reasoning, numerical reasoning, figural memory, verbal memory, numerical memory were derived based on the reduced set of test items applied from the BIS
- number of strategic changes in the participants' behavior for each of the four customer wishes
- index for the continuity of the order of actions while performing each of the four customer wishes

In order to derive the numerical values for the intelligence factors, the test items were analyzed as indicated in the BIS's handbook (Jäger, Süß, and Beauducel, 1997).

Altogether eight variables have been used to operationalize the degree of the innovation in the observable behavior (i.e. in this case the gardening tasks): the first is the number of strategic changes in the behavior of the participants for each of the customer wishes. For this purpose, the videos have been transliterated: A list of possible actions has been identified and the order of actions conducted has been analyzed. Each participant used a typical order of how to perform the task in question. The number of changes to this typical order has been counted as indicating the number of strategic changes in the behavior.

To derive the four indices for the continuity of the order of actions while performing the customer wishes, the number of grouped actions was counted. A participant conducting all required actions for one pot received a very low index of continuity (i.e. 1); whereas a participant executing one action for all pots received a high level of continuity (i.e., 10). Participants with a medium-sized index changed their strategy within the task. In order to be able to distinguish the participants who did not change their strategy and did change their strategy, a dichotomization was performed – the medium-sized numbers received the final index number 0, the high- and low-sized numbers received the final index number 1.

In order to test the hypotheses, repeated measurement analyses have been performed with each of the intelligent variables derived (list see above) as independent variables and either the number of strategic changes for all four customer wishes or the index of continuity for all four customer wishes as dependent variables. The significant results are given in Table 2. More specifically, the tested variables, the value of the used test statistic with the number of degrees of freedom, the probability that the expected effect did not occur due to chance and the size of the detected effect using a classification presented by Cohen (1992) are given.

Table 2: Results of the repeated measurement analyses.

Tested variables	Value of the used test statistics	Prob-ability	Effect size
1) Learning factor of the number of strategic changes	F(3,36) = 5.80	p = 0.00	f^2 = 0.49[1]

2) a) Learning factor of the number of strategic changes
b) Two way interaction between the learning factor and the verbal perceptual speed factor
	F(3, 30) = 4.09
	F(3, 30) = 2.75
		p = 0.02
		p = 0.06
			f^2 = 0.41[1]
			f^2 = 0.28[2]

3) a) Learning factor of the number of strategic changes
b) Two way interaction between the learning factor and the verbal memory factor
	F (3, 30) = 4.20
	F (3, 30) = 3.31
		p = 0.01
		p = 0.03
			f^2 = 0.42[1]
			f^2 = 0.33[2]

4) a) Learning factor of the index of continuity b) Two way interaction between the learning factor and the numerical memory factor	F (3, 30) = 4.34 F (3, 30) = 4.66	p = 0.01 p = 0.01	f^2 = 0.43[1] f^2 = 0.47[1]

*significant with $\alpha < 0.01$

[1] = large effect, [2] = medium-sized effect

As Table 2 indicates, the first analysis testing the learning factor of the number of strategic changes over the performed tasks is significant. A large effect has been found: The number of strategic changes shrinks with the number of customer wishes performed.

The second analysis tested the learning factor and the two way interaction effect between the learning factor and the verbal perceptual speed factor. Figure 3 shows this interaction. The participants with greater verbal perceptual speed abilities demonstrate more continuity regarding the strategic changes. The graph showing the course of the number of strategic changes while performing the four customer wishes for the participants with less verbal perceptual speed abilities demonstrates (1) that the general level of the number of strategic changes is greater compared to the number of strategic changes executed by the participants with higher verbal perceptual speed abilities and (2) that the variance of change is bigger.

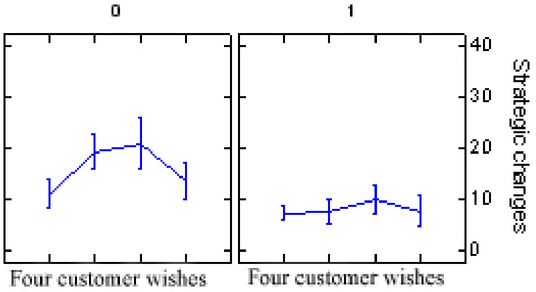


Figure 3: Line graph with standard error bars of the relationship between the strategic changes for the four customer wishes and the verbal perceptual speed of the participants (Graph 0 shows the less intelligent participants, Graph 1 the more intelligent participants).

The third significant analysis tested the learning factor and the two way interaction effect between the learning factor and the verbal memory factor. Figure 4 shows the direction of the effect. The two graphs comparing the course of the number of the strategic changes while performing the four customer wishes for the two groups of participants with high and low scores on the verbal memory factor closely resemble the graphs displayed before (see Fig. 3). Again, the less able participants demonstrate (1) a greater number of strategic changes and (2) a bigger change in the course of the skill acquisition/learning process.

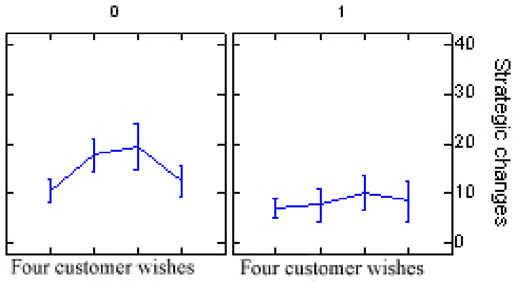


Figure 4: Line graph with standard error bars of the relationship between the strategic changes performed for four customer wishes and the verbal memory factor of the participants (Graph 0 shows the less intelligent participants, Graph 1 the more intelligent participants).

The fourth analyses tested (1) the learning factor of the index of continuity and (2) the two way interaction between the learning factor and the numerical memory factor. The results were significant as well. Hence, the index of continuity changes with the number of practice trials performed and this change depends on the level of numerical memory abilities of the participants.

4 CONCLUSIONS AND TECHNICAL IMPACT

Summarizing, relevant research results regarding the theory of structured intelligence are presented (Badreddin & Jipp, 2006). More specifically, the study confirmed the hypotheses: First, the degree of consistency in the behaviour of the participants gets greater with the task's familiarity. This transition shows the learning effect, i.e. the transition from innovation-based to experience-based, i.e., well known solutions. Second, a relationship between this transition and traditional measures of intelligence has been found. Significant predictors were intelligent factors such as the verbal perceptual speed factor. However, compared to traditional research on intelligence, the theory of structured intelligence goes one step further: it provides an explanation of how intelligent behaviour emerges and not only a classification of intelligent behavior. Drawbacks of the study refer to the small number of participants, which has reduced the power of the study, so that some possibly existing effects might not have been detected. Further, the creativity items had to be deleted from the intelligence test as they required the participants to draw solutions, which would disadvantage some of the participants due to their disability. Hence, future research should investigate the relationship between traditional creativity tests with innovation-based behaviour.

The study's main contribution is twofold: First, the theory of structured intelligence demonstrates links to traditional measurements of intelligence, but also gives an explanation of how intelligent behaviour emerges and provides the opportunity to measure intelligence in easier and less time-consuming ways and. It allows intelligence to be judged on (1) by using activity detection and (2) by observing the participants' actions when being aware of the familiarity of the task. The degree of consistency gives valuable information on intelligence. This has not only the potential to revolutionize intelligence diagnostics but also intelligent interface design: if an intelligent machine were capable of judging on its user's intelligence, the interface can be adapted to the user and different levels of support given. This might have a big impact on safety-critical applications with the user is in the loop of controlling e.g., nuclear power plants. Second, the theory of intelligence gives the artificial intelligence research a new direction, as not only experience is relevant, but also innovation-driven behaviour, which can be modelled as chaotic behaviour (see also Badreddin & Jipp, 2006).

REFERENCES

Ackerman, P. L., 1988. Determinants of individual differences during skill acquisition: Cognitive abilities and information processing. *Journal of Experimental Psychology: General, 117 (3),* pp. 288-318.

Anderson, J. R., 1982. Acquisition of cognitive skill. *Psychological Review,* 89, pp. 369-406.

Badreddin, E., Jipp, M., 2006. Structured Intelligence. *International Conference on Computational Intelligence for Modelling, Control, and Automation* CIMCA 2006, Sydney, Australia.

Campbell, D. T., Fiske, D. W., 1959. Convergent and discriminant validity by the multitrait-multimethod matrix, *Psychological Bulletin*, 56, pp. 81-105.

Fisk, A. D., Schneider, W., 1983. Category and word search: Generalizing search principles to complex processing. Journal of Experimental Psychology: Learning, Memory & Cognition, 9, pp. 177-195.

Fleisman, E. A., 1972. On the relations between abilities, learning, and human performance. American Psychologist. 27, 1017-1023.

Guttman, L. (ed.), 1954. Mathematical thinking in the social sciences. Glencoe, IL. The French Press.

Humphreys, L. G., 1979. The construct of general intelligence. *Intelligence,* 3, 105-120.

Jäger, A. O., 1982. Mehrmodale Klassifikation von Intelligenzleistungen: Experimentell kontrollierte Weiterentwicklung eines deskriptiven Intelligenzstrukturmodells, Diagnostica, 28(3), 195-225.

Jäger, A. O., Süß, H.-M., Beauducel, A., 1997. Berliner Intelligenzstruktur-Test, Göttingen, Hogrefe.

Jipp, M, Bartolein, C., Badreddin, E., 2007. Assisted Wheelchair Control: Theoretical Advancements, Empirical Results and Technical Implementation, Submitted for publication.

Jipp, M., Pott, P., Wagner, A., Badreddin, E., Wittmann, W. W., 2004. Skill acquisition process of a robot-based and a traditional spine surgery, 1st International Conference on Informatics in Control, Automation and Robotics, pp. 56-63.

Schneider, W., Shiffrin, R. M., 1977. Controlled and automatic human information processing: 1. Detection, search, and attention. *Psychological Review*, 84, pp. 1-66**.**

X3DIRECTOR
A Front-end For Modeling Web Virtual World Navigation And Visualization Parameters

Berta Buttarazzi and Daniele Pizziconi
University of Rome "Tor Vergata", Rome, Italy
buttarazzi@info.uniroma2.it,theone84@fastwebnet.it

Keywords: Virtual Reality, Virtual Environment, interactivity, fully immersive viewing systems, post-production phase, digital technologies.

Abstract: With the advent of digital communication technologies, a huge quantity of information can be stored and transferred. However, with increased possibility comes increasing complexity. As indicated by the dissatisfaction often experienced by users of information technology, current systems of accessing, organizing and navigating information are proving insufficient. For this reason, in the Web world, besides new media contents, new technologies for implementing more dynamic and user-friendly interfaces are being developed. Currently, the last step in the evolution of digital technologies for web interface and contents is represented by virtual environments which, despite the appearances, are, in some cases, very accessible and economic. This paper will provide a panorama on the functionalities of language X3D and the identification of the cases in which is possible to use X3Director.

1 INTRODUCTION

With the advent of digital communication technologies, a huge quantity of information can be stored and transferred. The last step in the evolution of digital technologies for web interface and contents is represented by virtual environments, and so by X3D.

The syntax and data structure of X3D provide an excellent tool for the modeling of three-dimensional worlds that are functional and interactive and that can, ultimately, be transferred into fully immersive viewing systems.

X3D allows to combine other Virtual Reality (VR) technologies and real environment, for example motion trackers are employed to monitor the movements of dancers or athletes for subsequent studies in immersive VR. These 'Augmented Reality' technologies allow for the viewing of real environments with superimposed virtual object; for example telepresence systems (e.g., telemedicine, telerobotics) immerse a viewer in a real world that is captured by video cameras at a distant location and allow for the remote manipulation of real objects via robotic arms and manipulators.

As the technologies of VR evolve, the applications of VR become literally unlimited. It is assumed that VR will reshape the interface between people and information technology by offering new ways for the communication of information, the visualization of processes, and the creative expression of ideas. Note that a virtual environment can represent any three-dimensional world that is either real or abstract. This includes real systems like buildings, landscapes, underwater shipwrecks, spacecrafts, archaeological excavation sites, human anatomy, and so on and abstract systems like magnetic fields, molecular models, mathematical systems and any other conceivable system including artistic and creative works of abstract nature.

This paper and therefore X3Directior is proposed like an aid in virtual environment refining those already existing, but not complete and not still ready for the distribution in the Web world.

In particular, the paper mainly deals with a specific category of users who might be called "non-expert developer". They are users that use, for work or pleasure, 3D graphics development software which supports virtual development language as X3D. These users can get a virtual representation of their 3D projects without having any knowledge of the X3D language. The goal is therefore to provide a software that helps this category of users in the refining phase, again without the exigency of any particular knowledge.

The second chapter introduces the X3Director software and its features.

2 X3DIRECTOR

The production of the three-dimensional structure of an X3D environment can completely be delegated to the software of authoring 3d and, that concurs to customers non-expert of X3D to develop, in immediate and practically free way, virtual environments compatible with the Web. After such phase it is necessary, for the effective fruibility of the virtual representation, a phase of post-production that proceeds along one series of tried and modifications. Phase that cannot be delegated to the software of authoring 3D is because often, being thought for other, do not support some of the demanded characteristics (e.g. the Background node), or, in case they come supported, their implementation demands specific knowledge of language X3D.

X3Director helps the "non-expert developer" to accomplish the post-production phase. The software provides a series of functions to manage the 3D aspect of an X3D file. It allows realizing different actions so as to add localized audio contexts or sounds that start after having clicked on a particular area or object, thanks to the use of the TouchSensor node.

X3Director proposes a linear, intuitive and provided of continuous assistance interface that allows the user in few mouse clicks to edit or to add ex novo the Background node in order to better characterize a three-dimensional environment, the NavigationInfo node that describes the view and the characteristics of the avatar, user's alter-ego, and the Sound and AudioClip nodes. Moreover, thanks to integration of parser DOM with the data structure JTree provided from the Java Swing package, X3Director provides, at nodes level, for the loaded X3D file, an interface characterized from tree structure that reflects the characteristic hierarchical structure of XML files. As far as the attributes, instead, X3Director has integrated an other data structure, the JTable, in order to organize and to edit in "expert" way the attributes of the nodes.

Such a software, conceived in order to come towards requirement of a non-expert user in the modeling of virtual environment in X3D, allows to verify the user's modifications, before the final saving, thanks to the preview function (it is necessary to have a player X3D installed in the operative system). This function can be used in order to evidence the modifications carried out globally in the file X3D and therefore to have a preview of the final complete file X3D or to evidence single geometries and therefore to have a visual reply on the single Transform or Group nodes. In order to complete this global view on the X3Director it is to be underlined that such a software does not forget the more expert users that they do not have need of the driven procedure. In order to edit or to add the above mentioned nodes, these users can either use simple functions of "add" or "delete" or can directly modify the values of the attributes editing them with a simple double click on the data structure JTable left on purpose editable for this function.

2.1 Characterizing Features

The absence of the necessity of specific knowledge differentiates X3Director from the others editor X3D and it renders it ideal for the phase of post-production. The below table shows a comparison of main editors X3D and X3Director. From the comparison, that refers exclusively to the phase of post-production, tool of optimization of the code and the software of authoring 3d have been omitted.

Table 1: Comparison between X3D post-production tools.

Software	Knowledge X3D	Potentiality	Complexity
X3Director	null	medium	low
SwirlX3D	medium	medium	medium
X3D-Edit	high	high	medium

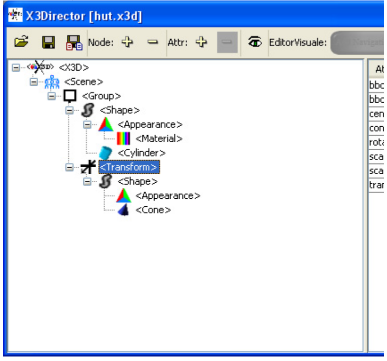
Everyone of the software shown can be used to modify file X3D exported from the software of authoring. It is obvious that, with the exception of X3Director, is however necessary a sure level of knowledge of language X3D.

SwirlX3D is a powerful X3D authoring environment that gives full access to the data structures in an X3D file and is able to preview the completed scene. The specific knowledge of X3D are not excessive but of against it does not offer, as instead X3Director offers, no visual procedures that it renders the formulation of the parameters independent from the knowledge of language X3D.

The graphics file editor X3D-Edit includes the typical characteristics of this type of editor: context-sensitive editing prevents creation of incorrect scene-graph structures and detects invalid numeric values, context-sensitive tooltips provide concise explanations and warnings of each X3D node and field, tag-inspectors, syntax highlighting, tag hints, code fragments, objects, edit dialogs and much more. They demand a deepened knowledge of language X3D with the exception of X3Director that has a complexity of use practically null thanks to the friendly and intuitive interface, and offers to the developer the possibility to act on the independent, from the three-dimensional structure, aspects of the virtual environment (phase of post-production for non-expert customers), or to modify whichever other parameter (expert way). In the next sections the several features are described in more detail.

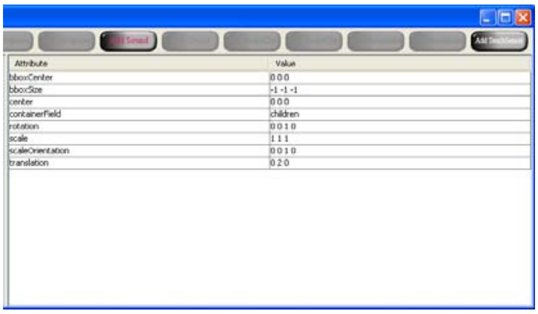
2.2 File Selection and Parser

When the X3D file is selected and loaded, X3Director displays it using a JTree that reflects the hierarchical structure of an X3D/XML file (see Figure 1 below).



```
<X3D>
  <Scene>
    <Group>
      <Shape>
        <Appearance>
          <Material>
        <Cylinder>
      <Transform>
        <Shape>
          <Appearance>
          <Cone>
```

Figure 1: Node X3D in tree structure.



Attribute	Value
bboxCenter	0 0 0
bboxSize	-1 -1 -1
center	0 0 0
containerField	children
rotation	0 0 1 0
scale	1 1 1
scaleOrientation	0 0 1 0
translation	0 2 0

Figure 2: Attribute X3D in JTable structure.

Now it is possible to have access of all functionalities provided by X3Director. As an example clicking on any node, such software will show to right, organized in a JTable, the attributes and their values of the selected node (Figure 2): then, as pointed out in the previous cross-section, it is possible to edit manually, by the more expert users, the values of the attributes.

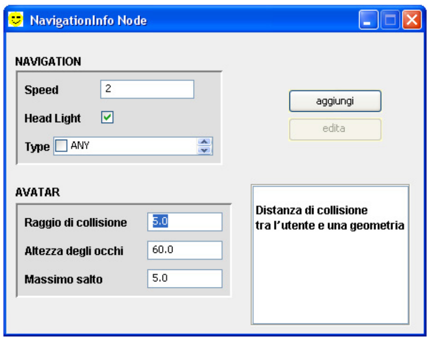
2.3 Update

Once the file is loaded, a series of operations can be carried out on it as addition or the modification of nodes Background, NavigationInfo, Sound, AudioClip and TouchSensor.

In the next subsections the driven procedures provided by X3Director will be analyzed one by one.

2.3.1 Navigation and Avatar Parameters

Navigation and Avatar parameters are included in the NavigationInfo node. In order to open frame (Figure 3), where it is possible to update such parameters, it is necessary in a loaded X3D file, either the presence of a NavigationInfo node to be modified, or clicking on the Scene or Group node and to push the AddNavigationInfo button present in the toolbar of X3Director in order to add a node ex novo.



NAVIGATION

Speed: 2
Head Light: ☑
Type: □ ANY

AVATAR

Raggio di collisione: 5.0
Altezza degli occhi: 60.0
Massimo salto: 5.0

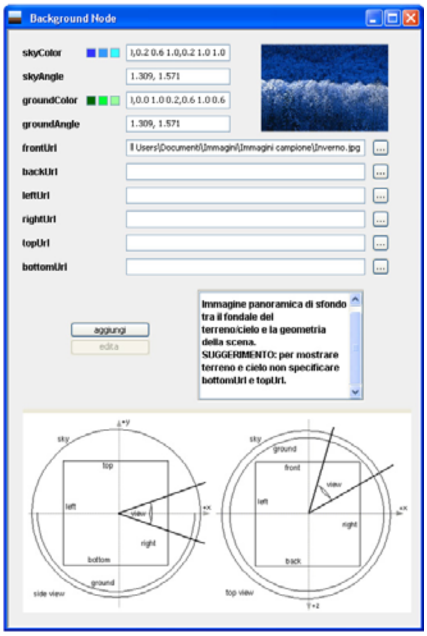
Distanza di collisione tra l'utente e una geometria

Figure 3: NavigationInfo procedure.

2.3.2 Background Parameters

Also for this node, as for the NavigationInfo, the driven procedure (Figure 4) can be opened in order to modify an existing Background node or in order to add a Background node ex novo. Such procedure allows the user to select three colors for sky and three for the land through the JColorChooser class supplied from the Java Swing package.

X3Director provides an interface with a text field for every cube's face with relative file browser button. The image path can be directly put in the relative text field or it can be obtained by file browser. The software, moreover, features a preview of images selected.



Background Node

skyColor	l,0.2 0.6 1.0,0.2 1.0 1.0
skyAngle	1.309, 1.571
groundColor	l,0.0 1.0 0.0.2,0.6 1.0 0.6
groundAngle	1.309, 1.571
frontUrl	l Users\Documenti\Immagini\Immagini campione\Inverno.jpg
backUrl	
leftUrl	
rightUrl	
topUrl	
bottomUrl	

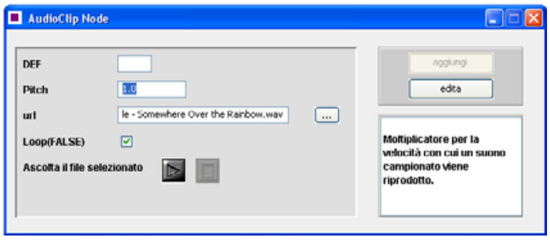
Immagine panoramica di sfondo tra il fondale del terreno/cielo e la geometria della scena. SUGGERIMENTO: per mostrare terreno e cielo non specificare bottomUrl e topUrl.

Figure 4: Background procedure.

2.3.3 Sound Parameters

For the adjustment of sound parameters X3Director provides a further driven procedure (Figure 6). Here it is possible to modify the parameters regarding the ellipsoids representing localization and propagation of the audio source (it is also possible to listen in preview the file audio selected) that otherwise have to be specified in a procedure regarding the AudioClip node (Figure 5).

Taking up the Sound procedure again, X3Director gives the possibility to add audio file with use of TuchSensor node. In this case the audio will start from the user's click on the geometry before selected in the Sound procedure. Once added the Sound node with TouchSensor it is possible to add to a further TouchSensor node and respective node ROUTE required to assume the task of "stop" to the file audio enabled with a click thanks to the previous TouchSensor.



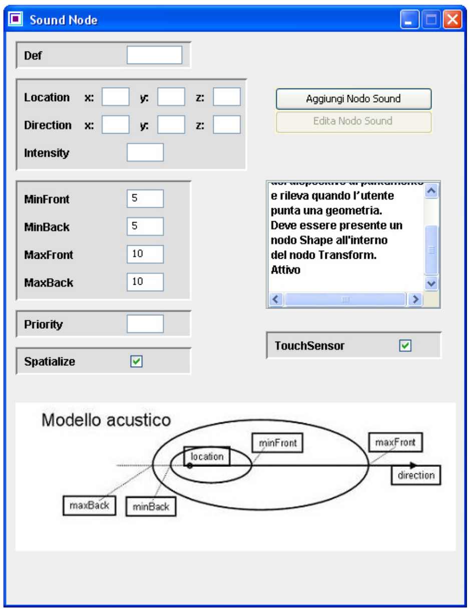
AudioClip Node

DEF	
Pitch	1.0
url	le - Somewhere Over the Rainbow.wav
Loop(FALSE)	☑

Ascolta il file selezionato

Moltiplicatore per la velocità con cui un suono campionato viene riprodotto.

Figure 5: AudioClip procedure.



Sound Node

Def	
Location	x: y: z:
Direction	x: y: z:
Intensity	
MinFront	5
MinBack	5
MaxFront	10
MaxBack	10
Priority	
Spatialize	☑
TouchSensor	☑

e rileva quando l'utente punta una geometria. Deve essere presente un nodo Shape all'interno del nodo Transform. Attivo

Figure 6: Sound procedure.

3 CONCLUSIONS

The Web represents today a fantastic showcase for commercial or non-commercial messages. Its diffusion and inexpensiveness makes it the best way to obtain visibility and in this scenario it is fundamental to offer a good Web experience that is not frustrating and that has a good impact on the visitors. It is therefore developed X3Director, a front-end which helps the X3D non-expert users (but it could also be helpful to expert users) for modeling navigation and visualization parameters in X3D environment. The software allows realizing the post-production phase without any knowledge of X3D language.

An X3D non-expert user could therefore develop and deliver a X3D content on the Web by using the combination 3D authoring software-X3Director.

REFERENCES

Pohja, M., 2003. *X3D and VRML Sound Components.*

Bouras, Ch., Panagopoulos, A., Tsiatos, Th., 2005. *Advanced in X3D Multi-user Virtual Environments,* Seventh IEEE International Symposium on Multimedia (ISM'05).

Behr, J., Dahne, P., Roth, M., 2004. *Utilizing X3D for Immersive Environments.*

Figueroa, P., Medina, O., Jimenez, R., Martinez, J., Albarracin, C., 2005. *Extensions for Interactivity and Retargeting in X3D.*

A STUDY OF TWO COLOR SYSTEMS USED IN CONTENT-BASED IMAGE QUERY ON MEDICAL IMAGERY

Liana Stanescu, Dumitru Burdescu, Cosmin Stoica and Marius Brezovan
University of Craiova, Faculty of Automation, Computers and Electronics
stanescu@software.ucv.ro, burdescu@topedge.com, stoica_cosmin@software.ucv.ro, brezovan_marius@software.ucv.ro

Keywords: Image feature extraction, image processing, content-based visual query, color, histogram, HSV color space, l color space, medical images.

Abstract: The article presents a comparative study over two methods used in content-based visual query. The two methods refer at two different color systems used for representing color information from images: HSV quantized to 166 colors and l1l2l3 quantized to 64 colors. The originality of the study comes from the fact that it was made on a database with medical images from digestive tract area, captured by an endoscope. The scope was to check the quality of the content-based visual query on images representing five different diagnoses (colitis, ulcer, esophagitis, polyps and ulcerous tumor) and taking into consideration that some parameters were modified during the capturing process: viewing direction, intensity and direction of the illumination, parameters that affect mostly the medical images captured during the diagnosis process.

1 INTRODUCTION

As the world is in the middle of the digital era, the quantity of visual information is increasing (Sebe and Lew, 2001). More than 2700 digital pictures are made in every second (in total 85 billion images yearly). For example, PhotoWorks includes tens of millions of images on its web site. The Internet allows us to have access to a big part of these images. The common images are completed by images with special purpose, like medical images with an estimation of 2 billion per year. Because of the tendency for digital (television, movies) and because everybody will have access to everything, the number of images will be increasing. The world production of digital information in 2007 is estimated to be more than 10^9 GB (250 MB for each man on the planet, ignoring his technological development). It is estimated that in the next 10 years, each of us will manipulate terabytes of information (video, static images, music, photos and documents) every day.

These image databases are associated with the problem of the content-based retrieval, solved in two steps (Sebe and Lew, 2001).

In the first step, when inserting a new image, it will be pre-processed and some features will be automatically extracted: color, texture and shape. The result will be a characteristics vector that will be stored in the database.

In the second step the content based retrieval is made, by choosing a query image, calculating the characteristics vector, comparing this vector with each vector of the images stored in the database and viewing the most similar images.

The color is one of the base image properties. In content based retrieval on color feature it is desired to find the images from the database having the color composition closest to the color composition of the query image (Del Bimbo, 2001, Gevers and Smeulders, 1999).

The color content of an image is best represented by color histograms.

Comparing color histograms of the query image and target image is done by histograms intersection or by the quadratic distance between histograms that takes into consideration the conceptual similitude between two colors (Sebe and Lew, 2001, Smith, 1997).

As was said before, one of the domains where a large number of images are accumulated is the medical domain. The advantages of using content-based visual query on medical images are on the following directions (Muller et al, 2004):

- Medical teaching
- Medical research
- Diagnostic aid
- Electronic patient records

The medical images are being produced directly by medical equipment used in patient diagnosis, or by digitalizing the images stored on other devices.

In each of these methods some changes can occur:

- Changes in viewing direction
- Changes in direction of the illumination
- Changes in the intensity of the illumination

As a result, the purpose of the paper is to make a comparative study of the content-based query results effectuated on medical images database where the color information is represented by HSV and l1l2l3 color systems.

The originality of the study is given by the fact that the experiments are made on medical images from digestive area produced by an endoscope. The ill area is seen from different directions and in different illumination intensity. This study, unlike the others made on CorelDraw images, uses images produced in real condition, in patient diagnosis.

The paper has the following structure: In section 2 the two color systems are presented. In section 3 the conditions and the results of the experiments are presented, and in section 4 the conclusions of the comparative study are discussed.

2 CONTENT-BASED IMAGE QUERY ON COLOR FEATURE

The color is the visual feature that is immediately perceived on an image. The color space used for representing color information in an image has a great importance in content-based image query, so this direction of research was intensely studied (Del Bimbo, 2001).

There is no color system that it is universal used, because the notion of color can be modeled and interpreted in different ways. Each system has its own color models that represent the system parameters (Gevers, 2004).

There were created several color spaces, for different purposes: RGB (for displaying process), XYZ (for color standardization), rgb, xyz (for color normalization and representation), CieLuv, CieLab (for perceptual uniformity), HSV (intuitive description) (Gevers, 2001, Gevers, 2004). The color systems were studied taking into consideration different criteria imposed by content-based visual query (Gevers and Smeulders, 1999):

- The independence of the imaging device
- Perceptual uniformity
- Linear transformation
- Intuitive for user
- Robust against imaging conditions: invariant to a change in viewing direction, invariant to a change in object geometry, invariant to a change in direction and intensity of the illumination and invariant to a change in the spectral power distribution of the illumination.

The studies have shown that two of these color systems can be used, with good results in a content-based visual query process, namely HSV and l1l2l3 (Gevers et al, 2006).

It was proved that the HSV color system has the following properties (Gevers, 2004):

- It is close to the human perception of colors
- It is intuitive
- It is invariant to illumination intensity and camera direction

The studies made on nature and medical images have shown that in the case of the HSV, RGB and CieLuv color systems, the HSV color space produces the best results in content based retrieval (Stanescu et al, 2006).

Still, the HSV color space has several problems (Gevers, 2004) :

- Nonlinear (but still simple) transformation from RGB to HSV
- Device dependent
- the H component becomes instable when S is close to 0
- the H component is dependent of the illumination color

Gevers and Smeulders have proposed a new color system, named l, whose components are defined using the equations (Gevers and Smeulders, 1999):

$$\begin{aligned}
l1(R,G,B) &= \frac{(R-G)^2}{(R-G)^2+(R-B)^2+(G-B)^2} \\
l2(R,G,B) &= \frac{(R-B)^2}{(R-G)^2+(R-B)^2+(G-B)^2} \qquad (1) \\
l3(R,G,B) &= \frac{(G-B)^2}{(R-G)^2+(R-B)^2+(G-B)^2}
\end{aligned}$$

Where R, G, B are the color values in the RGB color space. They also showed that the l color system is invariant to viewing direction, illumination direction and intensity. In this case it is also a nonlinear, but simple, transforming from RGB space to l space.

In case of medical images the main problems are regarding changing illumination intensity and viewing direction. That is why the two color spaces presented above are chosen.

The operation of color system quantization is needed in order to reduce the number of colors used in content-based visual query: from millions to tens. The quantization of the HSV color space to 166 colors, solution proposed by J.R. Smith, is the idea used in this study (Smith, 1997). For the color space l1l2l3 the solution of quantization to 64 colors is chosen, keeping 4 values for each component of the system. The fact that a color system is quantized to 166 colors and the other to 64 colors does not influence the quality of the content-based image query process, the research studies showing clearly this aspect (Stanescu et al, 2006). The color histograms represent the traditional method of describing the color properties of the images. They have the advantages of easy computation and up to certain point are insensitive to camera rotating, zooming, and changes in image resolution (Del Bimbo, 2001). In case of both color systems, to compute the distance between the color histograms of the query image and the target image, the intersection of the histograms is used (Smith, 1997). The studies have also shown that using this metric in content-based visual query gives very good results as quadratic distance between histograms that is more difficult to calculate (Smith, 1997, Stanescu et al, 2006).

3 EXPERIMENTS

The experiments were performed in the following conditions.

A database with 520 color images from the field of the digestive apparatus was created. The images are from patients with the following diagnosis: polyps, ulcer, esophagitis, ulcerous tumors and colitis. For each image there are several images with affected area captured from 3 or 4 viewing directions. For each image in the database there is another identical image, but having the illumination intensity changed.

A software tool that permits the processing of each image was created. The software tool executes the following steps:

1. the transformation of image from RGB color space to HSV color space and the quantization to 166 colors
2. the transformation of image from RGB color space to l1l2l3 color space and the quantization to 64 colors
3. calculation of the two color histograms with 166, respectively 64 values, that represent the characteristics vectors and storing them in the database

In order to make the query the procedure is:

- a query image is chosen
- the dissimilitude between the query image and every target image from the database is computed, for each two specified criteria (color histograms with 166 colors and the color histogram with 64 colors);
- the images are displayed on 2 columns corresponding to the 2 methods in ascending order of the computed distance.

For each query, the relevant images have been established. Each of the relevant images has become in turn a query image, and the final results for a query are an average of these individual results.

The experimental results are summarized in table 1. Method 1 represents the query using the HSV color space quantized at 166 colors and Method 2 represents the query on color using the l1l2l3 color space quantized at 64 colors. The values in the table represent the number of relevant images of the first 5 images retrieved for each query and each of the methods, as an average of the values obtained on each executed query.

Table 1: Experimental results.

Query	Method 1	Method 2
Polyps	3.6	3.2
Colitis	3.5	3.1
Ulcer	3.2	2.9
Ulcerous Tumor	3.5	3.1
Esophagitis	3.4	3.1

It must be mentioned that the queries were made for each of the 5 diagnostics in part. The notion of relevant image was strictly defined. The images from the same patient captured at different illumination intensity and from different points of view were considered relevant for a query, and not the ones with the same diagnosis. The quality of the content-based image query process was strictly analyzed. In figure 1 there is an example of content-based image query considering the two specified methods for images categorized as colitis. The first column contains 5 images retrieved by Method1 and the second contains the images retrieved using Method2. In the first case there are 5 relevant images and in the second case, 4 relevant images.

4 CONCLUSION

The paper presents the condition in which the quality of the content-based visual query process was studied, using a collection of medical images from digestive tract. The quality was measured

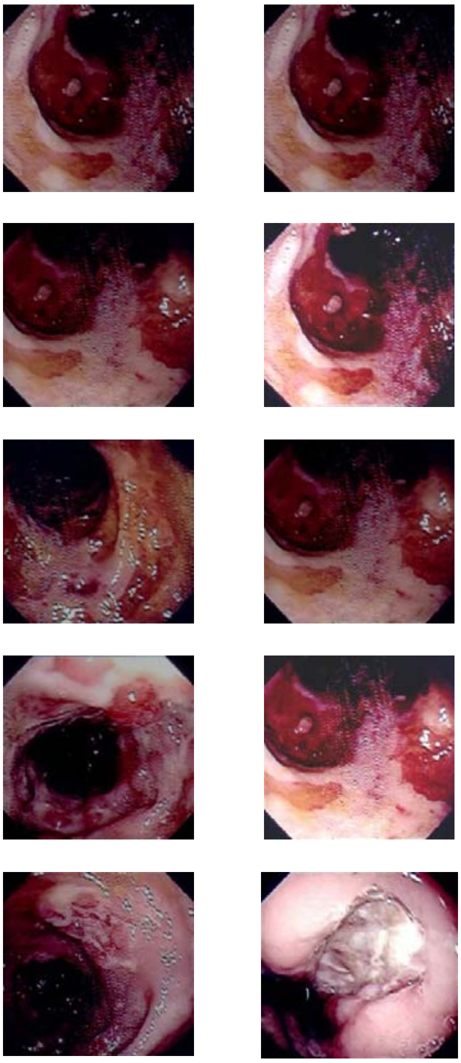


Figure 1: The retrieved images using the two specified methods.

calculating the precision and recall parameters. HSV system, quantized to 166 colors and l1l2l3 color system quantized to 64 colors were considered highlighting the way they influence the process of content-based visual query if some important parameters that often affects medical images are modified: viewing direction, direction and intensity of the illumination.

Several conclusions can be formulated after the experimental results were analyzed:

1. to find images representing the same ill area, that were captured by an endoscope from several viewing directions, the solution that uses HSV color system quantized to 166 colors gives the best results
2. for images representing the same ill area, captured to different illumination intensities, the solution that uses l1l2l3l color system quantized to 64 colors, gives the best results in querying process
3. globally, the solution that uses HSV color space gives most satisfying results, because the database includes both types of images

In general, for medical images, the first case, with images representing ill area captured from different angles is the most frequent case. So, that is why the use of HSV color space, quantized to 166 colors, is recommended. The situation in the database that was studied was the same, namely, the number of images captured from different angles was higher than the number of images where only the illumination intensity was different.

In the future the study will be extended by using a bigger database with much more images in order to see if this conclusion will be also confirmed. New experiments with images from other parts of the human body or images produced by other medical devices will be effectuated.

REFERENCES

Del Bimbo, A., 2001. *Visual Information Retrieval,* Morgan Kaufmann Publishers. San Francisco USA.

Gevers, T., Smeulders, W.M., 1999. Color-based object recognition. *Pattern Recognition*. 32, 453-464

Gevers, T., 2001. Color in Image Search Engines. In *Principles of Visual Information Retrieval*. Springer-Verlag, London.

Gevers, T., 2004. Image Search Engines: An Overview. In *Emerging Topics in Computer Vision*. Prentice Hall.

Gevers, T., Van de Weijer. J., Stokman, H., 2006. Color Feature Detection. In *Color Image Processing: Methods and Applications*. CRC Press.

Muller, H., Michoux, N., Bandon, D., Geissbuhler, A., 2004. A Review of Content_based Image Retrieval Systems in Medical Application – Clinical Benefits and Future Directions. *Int J Med Inform*. 73(1)

Sebe, N., Lew, M., 2001. Color-based retrieval. *Pattern Recognition Letters*. 22, 223-230

Smith, J.R., 1997. *Integrated Spatial and Feature Image Systems: Retrieval, Compression and Analysis, Ph.D. thesis*, Graduate School of Arts and Sciences. Columbia University.

Stanescu, L., Burdescu, D.D., Ion, A., Brezovan, M., 2006. Content-Based Image Query on Color Feature in the Image Databases Obtained from DICOM Files. In : *International Multi-Conference on Computing in the Global Information Technology*. Bucharest. Romania

COMPARISON OF TWO IDENTIFICATION TECHNIQUES: THEORY AND APPLICATION

Pierre-Olivier Vandanjon$^{(1)}$, Alexandre Janot$^{(2)}$
$^{(1)}$LCPC, Route de Bouaye BP 4129, 44341Bouguenais Cedex, France
$^{(2)}$CEA/LIST Interactive Robotic Unit, 18 route du Panorama, BP 6, 92265 Fontenay aux Roses, Cedex, France
pierre-olivier.vandanjon@lcpc.fr, alexandre.janot@cea.fr

Maxime Gautier$^{(3)}$, Flavia Khatounian$^{(2)}$
$^{(3)}$IRCCyN, Robotic team, 1, rue de la Noë - BP 92 101 - 44321 Nantes Cedex 03, France
Maxime.Gautier@irccyn.ec-nantes.fr

Keywords: Parameters identification, inverse model, least squares method, simple refined instrumental variable method.

Abstract: Parametric identification requires a good know-how and an accurate analysis. The most popular methods consist in using simply the least squares techniques because of their simplicity. However, these techniques are not intrinsically robust. An alternative consists in helping them with an appropriate data treatment. Another choice consists in applying a robust identification method. This paper focuses on a comparison of two techniques: a "helped" least squares technique and a robust method called "the simple refined instrumental variable method". These methods will be applied to a single degree of freedom haptic interface developed by the CEA Interactive Robotics Unit.

1 INTRODUCTION

In most of cases, parametric identification is a hard task and requires a good know-how. A popular method consists in using the least squares technique (LS) because of its simplicity. However, it has been proven that the LS are sensitive to outliers, leverage points and noise models (Hampel, 1971) and (Hueber, 1981). In other words, the LS are not intrinsically robust and we must "help" them in order to obtain reliable results.

In the last decade, the IRCCyN robotic team has designed a identification method based on LS technique, inverse model, base parameters, exciting trajectories and appropriate data treatment (Gautier and Khalil, 1990), (Gautier and Khalil, 1991) and (Gautier, 1997). This technique has been successfully applied to identify inertial parameters of industrial robots (Gautier, Khalil and Restrepo, 1995) and (Gautier, 1997). Recently, it was used for identifying electrical parameters of a synchronous machine (Khatounian et al, 2006), inertial parameters of a haptic device (Janot et al, 2006) and the parameters of a compactor (Lemaire et al, 2006). The obtained results were interesting and reliable.

At the same time, robust methods have been successfully used (Mili, Phaniraj and Rousseeuw, 1990), (Mili et al, 1996), (Young 2006) and (Gilson et al, 2006). An interesting robust method is the simple refined instrumental variable (SRIV) because of its simplicity (Young 1979). Indeed, no noise model identification is needed and this method is consistent even in the coloured noise situation (Gilson et al, 2006). This method is implemented in the MATLAB CONTSID toolbox developed by the CRAN team (Garnier, Gilson and Huselstein, 2003) and (Garnier, Gilson and Cervellin, 2006). To our knowledge, this method has not been used to identify inertial parameters of robots.

Hence, it seems interesting to apply the SRIV method to a single degree of freedom haptic interface and to compare it with the LS method.

The paper is organized as follows: second section presents the general inverse dynamic model of robots, the LS technique and the SRIV method; experimental results are presented in section 3; finally, section 4 introduces a discussion.

2 THEORY

2.1 General Inverse Dynamic Model

The inverse dynamic model (commonly called dynamic model) calculates the joint torques as a function of joint positions, velocities and accelerations. It is usually represented by the following equation:

$$\mathbf{\Gamma} = \mathbf{A(q)\ddot{q}} + \mathbf{H(q,\dot{q})} + \mathbf{F_v\dot{q}} + \mathbf{F_c sign(\dot{q})} \qquad (1)$$

Where, $\mathbf{\Gamma}$ is the torques vector, $\mathbf{q}$, $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ are respectively the joint positions, velocities and accelerations vector, $\mathbf{A(q)}$ is the inertia matrix, $\mathbf{H(q,\dot{q})}$ is the vector regrouping Coriolis, centrifugal and gravity torques, $\mathbf{F_v}$ and $\mathbf{F_C}$ are respectively the viscous and Coulomb friction matrices.

The classical parameters used in this model are the components XX_j, XY_j, XZ_j, YY_j, YZ_j, ZZ_j of the inertia tensor of link j denoted ${}^j\mathbf{J_j}$, the mass of the link j called M_j, the first moments vector of link j around the origin of frame j denoted ${}^j\mathbf{MS_j}=[M_{Xj}\ M_{Yj}\ M_{Zj}]^T$, and the friction coefficients F_{Vj}, F_{Cj}.

The kinetic and potential energies being linear with respect to the inertial parameters, so is the dynamic model (Gautier and Khalil, 1990). It can thus be written as:

$$\mathbf{\Gamma} = \mathbf{D}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}})\chi \qquad (2)$$

Where $\mathbf{D(q,\dot{q},\ddot{q})}$ is a linear regressor and χ is a vector composed of the inertial parameters.

The set of base parameters represents the minimum number of parameters from which the dynamic model can be calculated. They can be deduced from the classical parameters by eliminating those which have no effect on the dynamic model and by regrouping some others. In fact, they represent the only identifiable parameters.

Two main methods have been designed for calculating them: a direct and recursive method based on calculation of the energy (Gautier and Khalil, 1990) and a method based on QR numerical decomposition (Gautier, 1991). The numerical method is particularly useful for robots consisting of closed loops.

By considering only the b base parameters, (2) can be rewritten as follows:

$$\mathbf{\Gamma} = \mathbf{W}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}})\chi_{\mathbf{b}} \qquad (3)$$

Where $\mathbf{W(q,\dot{q},\ddot{q})}$ is the linear regressor and $\chi_\mathbf{b}$ is the vector composed of the base parameters.

2.2 LS Method

2.2.1 General Theory

Generally, ordinary LS technique is used to estimate the base parameters solving an over-determined linear system obtained from a sampling of the dynamic model, along a given trajectory $(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}})$, (Gautier, Khalil and Restrepo, 1995), (Gautier, 1997). $\mathbf{X}$ being the b minimum parameters vector to be identified, $\mathbf{Y}$ the measurements vector, $\mathbf{W}$ the observation matrix and ρ the vector of errors, the system is described as follows:

$$\mathbf{Y}(\mathbf{\Gamma}) = \mathbf{W}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}})\mathbf{X} + \boldsymbol{\rho} \qquad (4)$$

The L.S. solution $\hat{\mathbf{X}}$ minimizes the 2-norm of the vector of errors ρ. $\mathbf{W}$ is a r×b full rank and well conditioned matrix, obtained by tracking exciting trajectories and by considering the base parameters, r being the number of samplings along a trajectory. Hence, there is only one solution $\hat{\mathbf{X}}$ (Gautier, 1997). Standard deviations $\sigma_{\hat{X}_i}$ are estimated using classical and simple results from statistics. The matrix $\mathbf{W}$ is supposed deterministic, and $\boldsymbol{\rho}$**,** a zero-mean additive independent noise, with a standard deviation such as:

$$\mathbf{C}_{\boldsymbol{\rho}} = E\left(\boldsymbol{\rho}\boldsymbol{\rho}^{\mathbf{T}}\right) = \sigma_{\boldsymbol{\rho}}^2\mathbf{I}_{\mathbf{r}} \qquad (5)$$

where E is the expectation operator and $\mathbf{I_r}$, the r×r identity matrix. An unbiased estimation of σ_ρ is:

$$\sigma_{\boldsymbol{\rho}}^2 = \frac{\left\|\mathbf{Y} - \mathbf{W}\hat{\mathbf{X}}\right\|^2}{(r-b)} \qquad (6)$$

The covariance matrix of the standard deviation is calculated as follows:

$$\mathbf{C}_{\hat{\mathbf{X}}\hat{\mathbf{X}}} = \sigma_{\boldsymbol{\rho}}^2\left(\mathbf{W}^{\mathbf{T}}\mathbf{W}\right)^{-1} \qquad (7)$$

$\sigma_{\hat{X}i}^2 = C_{\hat{X}\hat{X}_{ii}}$ is the i^th^ diagonal coefficient of $\mathbf{C}_{\hat{\mathbf{X}}\hat{\mathbf{X}}}$. The relative standard deviation $\%\sigma_{\hat{X}_{ri}}$ is given by:

$$\%\sigma_{\hat{X}_{jr}} = 100\frac{\sigma_{\hat{X}_j}}{\left|X_j\right|} \qquad (8)$$

However, in practice, $\mathbf{W}$ is not deterministic. This problem can be solved by filtering the measurement matrix $\mathbf{Y}$ and the columns of the observation matrix $\mathbf{W}$.

2.2.2 Data Filtering

Vectors $\mathbf{Y}$ and $\mathbf{q}$ are samples measured during an experimental test. We know that the LS method is sensitive to outliers and leverage points. A median filter is applied to $\mathbf{Y}$ and $\mathbf{q}$ in order to eliminate them.

The derivatives $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ are obtained without phase shift using a centered finite difference of the vector $\mathbf{q}$. Knowing that $\mathbf{q}$ is perturbed by high frequency noises, which will be amplified by the numeric derivation, a lowpass filter, with an order greater than 2 is used. The choice of the cut-off frequency ω_f is very sensitive because the filtered data $(\mathbf{q}_f, \dot{\mathbf{q}}_f, \ddot{\mathbf{q}}_f)$ must be equal to the vector $(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ in the range [0, ω_f], in order to avoid distortion in the dynamic regressor. The filter must have a flat amplitude characteristic without phase shift in the range [0, ω_f], with the rule of thumb $\omega_f > 10 * \omega_{dyn}$, where ω_{dyn} represents the dynamic frequency of the system. Considering an off-line identification, it is easily achieved with a non-causal zero-phase digital filter by processing the input data through an IIR lowpass butterworth filter in both the forward and reverse direction. The measurement vector $\mathbf{Y}$ is also filtered, thus, a new filtered linear system is obtained:

$$\mathbf{Y}_f = \mathbf{W}_f \mathbf{X}_f + \boldsymbol{\rho}_f \tag{9}$$

Because there is no more signal in the range [ω_f, $\omega_s/2$], where ω_s is the sampling frequency, vector $\mathbf{Y_f}$ and matrix $\mathbf{W_f}$ are resampled at a lower rate after lowpass filtering, keeping one sample over n_d samples, in order to obtain the final system to be identified:

$$\mathbf{Y}_{fd} = \mathbf{W}_{fd} \mathbf{X}_{fd} + \boldsymbol{\rho}_{fd} \tag{10}$$

with:

$$n_d = 0.8\omega_s / 2\omega_f \tag{11}$$

2.2.3 Exciting Trajectories

Knowing the base parameters, exciting trajectories must be designed. In fact, they represent the trajectories which excite well the parameters. If the trajectories are enough exciting, then the conditioning number of $\mathbf{W}$, (denoted cond($\mathbf{W}$)) is close to 1. However, in practice, this conditioning number can reach 200 because of the high number of base parameters. Design and calculations of these trajectories can be found in (Gautier and Khalil, 1991).

If the trajectories are not enough exciting, then the results have no sense because the system is ill conditioned and some undesirable regroupings occur.

2.3 SRIV Method

From a theoretical point of view, the LS assumptions are violated in practical applications. In the equation (4), the observation matrix $\mathbf{W}$ is built from the joint positions $\mathbf{q}$ which are often measured and from $\dot{\mathbf{q}}$, $\ddot{\mathbf{q}}$ which are often computed numerically from $\mathbf{q}$. Therefore the observation matrix is noisy. Moreover identification process takes place when the robot is controlled by feedback. It is well known that these violations of assumption imply that the LS solution is biased. Indeed, from (4), it comes:

$$\mathbf{W}^T\mathbf{Y} = \mathbf{W}^T\mathbf{W}\mathbf{X} + \mathbf{W}^T\boldsymbol{\rho} \tag{12}$$

As $\boldsymbol{\rho}$ includes noises from the observation matrix $E(\mathbf{W}^T\boldsymbol{\rho}) \neq 0$.

The Refined Instrumental Variable approach deals with this problem of noisy observation matrix and can be statistically optimal (Young, 1979). It is the reason why we propose to enrich our identification methods by using concepts from this approach. In the following, we describe the application of this method in our field.

The Instrumental Variable Method proposes to remove the bias on the solution by building the instrument matrix $\mathbf{V}$ such as $E(\mathbf{V}^T\boldsymbol{\rho}) = 0$ and $\mathbf{V}^T\mathbf{W}$ is invertible.

The previous equation becomes:

$$\mathbf{V}^T\mathbf{Y} = \mathbf{V}^T\mathbf{W}\mathbf{X} + \mathbf{V}^T\boldsymbol{\rho} \tag{13}$$

The instrumental variable solution is given by

$$\hat{\mathbf{X}}_V = \left(\mathbf{V}^T\mathbf{W}\right)^{-1}\mathbf{V}^T\mathbf{Y} \tag{14}$$

The main problem is to find the instrument matrix $\mathbf{V}$. A classical solution is to build an observation matrix from simulated data instead of measured data. The following iterative algorithm describes this solution:

Step 0: a first set of parameters is given by using classical LS.

Step k: From a set of parameters $\hat{\mathbf{X}}_{V/k-1}$ given at the previous step, the following ordinary differential equation, describing the robot dynamic, is simulated:

$$\ddot{\mathbf{q}} = \mathbf{A}^{-1}\left(\boldsymbol{\Gamma} + \mathbf{H}(\mathbf{q}, \dot{\mathbf{q}}) + \boldsymbol{\Gamma}_f\right) \tag{15}$$

$\boldsymbol{\Gamma}$ is the motor vector which is generally given by a feedback built by comparison between the desired and the real movement.

$\mathbf{A}$, $\mathbf{H}$, $\boldsymbol{\Gamma}_f$ are respectively the inertia matrix, the Coriolis-centrifugal vector and the friction torques. They are computed with the parameters identified at the step k-1: $\hat{\mathbf{X}}_{V/k-1}$

By simulation of this differential equation, $\mathbf{q_k}, \dot{\mathbf{q}}_\mathbf{k}, \ddot{\mathbf{q}}_\mathbf{k}$ are obtained. Then the Instrument Matrix at the step k is computed: $\mathbf{V_k} = \mathbf{W}(\mathbf{q_k}, \dot{\mathbf{q}}_\mathbf{k}, \ddot{\mathbf{q}}_\mathbf{k})$. The equation (14) gives the set of parameters $\hat{\mathbf{X}}_{Vk}$.

This iterative process is conducted until the convergence of the parameters. Practically, 4 or 5 steps are enough to decrease dramatically the correlation between the instrument matrix and the noise on the system.

3 APPLICATION

3.1 Presentation and Modelling of the Interface

The interface to be identified is presented Figure 1. It consists of synchronous machine and a handle actuated thanks to a cable transmission. This type of transmission is a good comprise between friction, slippage and losses. The modelling and the identification are made under the rigid model assumption. Naturally, the authors have checked that this hypothesis is valid in the case of this study.

The modelling and the identification of the synchronous machine are given in (Khatounian et al, 2006).

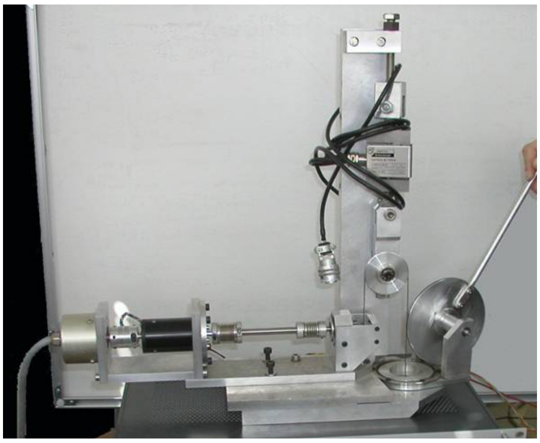


Figure 1: Presentation of the interface to be identified.

The frames defined to model the interface are presented Figure 2. The inverse dynamic model is given by the following equation:

$$\Gamma = J\ddot{q} - M_X g\cos(q) + M_Y g\sin(q) + \Gamma_f \qquad (16)$$

where q and $\ddot{q}$ are respectively the joint position and acceleration, J is the global inertia of the system.

The torque Γ is calculated through the current measurement. We have verified that we have $\Gamma = NK_cI$, where N is the gear ratio, K_c the torque constant and I the measured current. More details about the implemented control law can be found in (Khatounian et al, 2006).

The friction torque Γ_f was measured according to the method developed by (Spetch and Isermann, 1988), which consists in measuring the torque at different constant speeds and eliminating all transient modes. Figure 3 shows that the friction model can be considered at non-zero velocity as the following, where $\text{sign}(\dot{q})$ is the sign function of the velocity $\dot{q}$, and f_c, f_v denote the Coulomb and viscous friction coefficients.

$$\Gamma_f = f_v\dot{q} + f_c\text{sign}(\dot{q}) \qquad (17)$$

The dynamic model can be thus written as a sampled linear form:

$$\mathbf{\Gamma} = \mathbf{W}\,\mathbf{X} \qquad (18)$$

$$\mathbf{W} = \begin{bmatrix} \ddot{q}_1 & -g\cos(q_1) & g\sin(q_1) & \dot{q}_1 & \text{sign}(\dot{q}_1) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \ddot{q}_r & -g\cos(q_r) & g\sin(q_r) & \dot{q}_r & \text{sign}(\dot{q}_r) \end{bmatrix},$$

$$\mathbf{Y} = \begin{bmatrix} \Gamma_1 \\ \vdots \\ \Gamma_r \end{bmatrix} \text{ and } \mathbf{X} = \begin{bmatrix} J & M_X & M_Y & f_v & f_c \end{bmatrix}^T.$$

In this simple case, the parameters J, M_X, M_Y, f_v and f_c constitute the set of base parameters (QR decomposition confirms that).

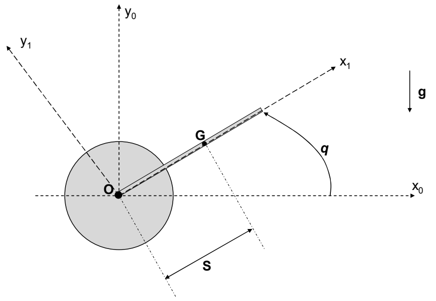


Figure 2: Notations and frames used for modelling the interface.

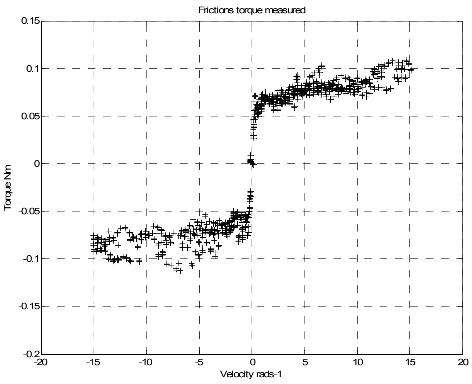


Figure 3: Friction torque measured.

3.2 Identification of the Base Parameters

3.2.1 LS Method

Current **I** and joint position **q** were measured, with a sampling period of 240µs. Exciting trajectory consists of trapezoidal speed reference. This trajectory was chosen because it is equivalent to a rectangular trajectory for the joint acceleration. It allows the estimation of the inertia term due to the change of the acceleration sign, of the gravitational and viscous friction terms because the velocity is constant over a period and of the Coulomb term due to the change of sign of the velocity. Vectors $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ were derived from the position vector **q**, and all data were filtered as described in section 2.2.2. Because of the friction model, the velocities close to zero are eliminated.

The cut-off frequency of the butterworth filter and of the decimate filter is close to 20Hz. This value can be calculated thanks to a spectral analysis of the data. In our case one has ω_{dyn} close to 2Hz. Thus with the rule of thumb $\omega_f > 10*\omega_{dyn}$, we retrieve the value given above.

The identified values of the mechanical parameters are summed up in Table 1.

The conditioning number of **W** is close to 30. Hence, the system is well conditioned and the designed trajectories are enough exciting.

Direct comparisons have been performed. These tests consist in comparing the measured and the estimated torques just after the identification procedure. An example is illustrated Figure 4. Notice that the estimated torque follows the measured torque closely.

Finally, we simulate the system by means of a SIMULINK model. We use the same speed references and the differential equation given by (15) is solved with the Euler's method (ODE 1). The step time integrator is of 240µs. Figure 5 shows the measured and the simulated states. They are close to each others.

Table 1: Estimated mechanical parameter values.

Mechanical parameters	Estimated values	Relative deviation %
J (kg.m²)	$1{,}45.10^{-3}$	0,5
M_X (kg.m)	$2{,}2.10^{-3}$	5,0
M_Y (kg.m)	$0{,}9.10^{-3}$	8,7
f_v (Nm.s.rad^{-1})	$2{,}4.10^{-3}$	2,6
f_c (Nm)	$5{,}9.10^{-2}$	0.8
Cond(**W**) = 30		

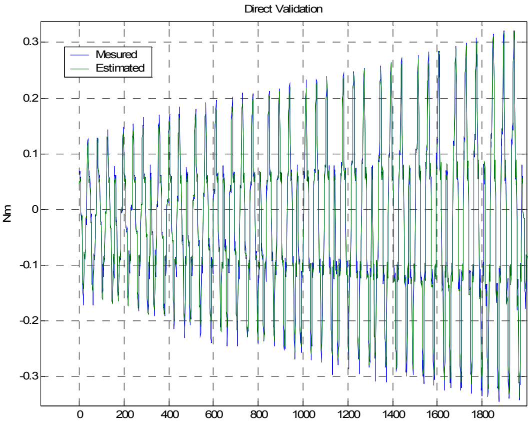


Figure 4: Comparison between the measured and estimated torque calculated through the LS technique.

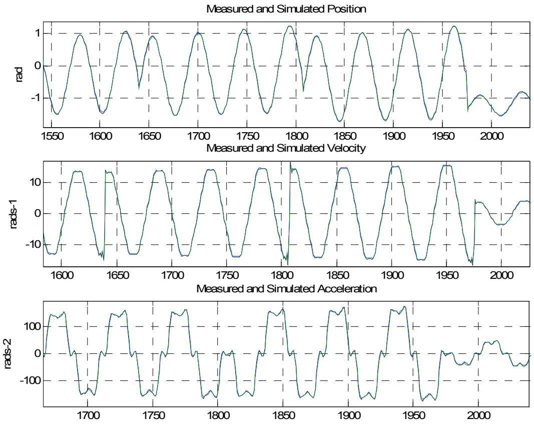


Figure 5: Comparison between the measured states (blue) and the simulated states (green).

It comes that the LS technique identification gives interesting and reliable results even if the assumptions on the noise model are violated.

3.2.2 SRIV Method

As explained section 2.3, the instrument matrix is built with the simulated states (see previous section for the simulation parameters) and the algorithm is initialized with the values given in Table 1. The parameters are estimated thanks to (14). These values do not vary consequently (only 2 steps are enough to decrease the correlation between the instrument matrix and the noise on the system). The results are summed up in Table 2.

We have also performed direct comparison (Figure 6). Once again, the estimated torque follows the measured torque closely. As an expected result, the SRIV method gives reliable and interesting results and can be used to identify inertial parameters of robots. The results given Table 2 are close to those exposed Table 1. That means that, in this case, the LS method detailed section 2.2 is as efficient as the SRIV technique detailed section 2.3.

Table 2: Estimated mechanical parameter values calculated through the SRIV method after 5 steps.

Mechanical parameters	Estimated values	Relative deviation %
J (kg.m²)	$1{,}45.10^{-3}$	0,4
M_X (kg.m)	$2{,}2.10^{-3}$	5,0
M_Y (kg.m)	$0{,}9.10^{-3}$	8,2
f_v (Nm.s.rad^{-1})	$1{,}9.10^{-3}$	3,5
f_c (Nm)	$6{,}3.10^{-2}$	0.8
Cond(**W**) = 30		

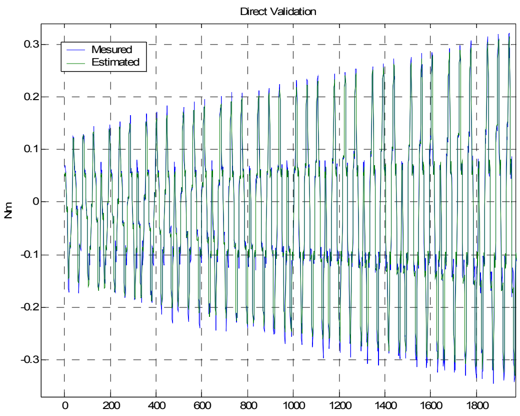


Figure 6: Comparison between the measured and estimated torque calculated through the SRIV.

4 DISCUSSION

In the case of this study, the experimental results tend to prove that the identification technique developed by the IRCCyN robotic team is as efficient and consistent as the SRIV method. This mainly due to the data treatment described section 2.2.2. Indeed, it helps the LS estimator: outliers and leverage points are eliminated while the derivatives of **q** are calculated without magnitude and phase shift. In addition, the data filtering is well designed.

It is interesting to mix both methods. Indeed, with a simple analysis, we can detect which parameters are sensitive to noise measurement, noise modelling, high frequency disturbances…

To do so, we introduce an estimation relative error given by (19):

$$\%e = 100\left|1-\frac{\hat{X}_{SRIVj}}{\hat{X}_{LSj}}\right| \tag{19}$$

Where $\hat{X}_{SRIVj}$ is the jth parameter estimated by means of the SRIV method and $\hat{X}_{LSj}$ is the jth parameter estimated thanks to the LS technique. The results are summed up in Table 3.

In our case, only the parameters of friction torque are quite sensitive. This is mainly due to the fact that they “see” all undesirable effects such as torque ripple, cable wear and imperfect contacts which are not modelled. Hence, that proves the interest of this approach and the use of the SRIV method.

Table 3: Estimation relative error.

Parameter	%e
J (kg.m²)	0,1 %
Mx (kg.m)	0,2 %
My (kg.m)	1,3 %
fv (Nm.s.rad^{-1})	20 %
fc (Nm)	7,3 %

During the experiments, it is appeared that the SRIV could be helpful when we are not familiar with the data filtering. We know that the LS estimators are sensitive to noises filtering and if the filtering is not well designed, the results could be controversial. These results will be published in a later publication.

5 CONCLUSION

In this paper, two identification methods have been tested and compared: the first based on the ordinary LS technique associated with an appropriate data treatment and the second based on the SRIV method. Both methods give interesting and reliable results. Hence, we can choose these techniques for a parametric identification.

In addition, the authors have introduced a simple calculation which enables us to know the parameters which are sensitive to noises and undesirable effects.

Future works concern the use of both techniques to identify a 6 degrees of freedom robot.

REFERENCES

Garnier H., Gilson M. and Huselstein E., 2003. “Developments for the MATLAB CONTSID toolbox”, In: *13th IFAC Symposium on System Identification, SYSID 2003*, Rotterdam Netherlands, August 2003, pp. 1007-1012

Garnier H., Gilson M. and Cervellin O., 2006. “Latest developments for the MATLAB CONTSID toolbox”, In: *14th IFAC Symposium on System Identification, SYSID 2006*, Newcastle Australia, March 2006, pp. 714-719

Gautier M. and Khalil W., 1990. "Direct calculation of minimum set of inertial parameters of serial robots", *IEEE Transactions on Robotics and Automation*, Vol. 6(3), June 1990

Gautier M. and Khalil W., 1991. “Exciting trajectories for the identification of base inertial parameters of robots”, *In: Proc. Of the 30th Conf. on Decision and Control*, Brigthon, England, December 1991, pp. 494-499

Gautier M., 1997. “Dynamic identification of robots with power model”, *Proc. IEEE Int. Conf. on Robotics and Automation*, Albuquerque, 1997, p. 1922-1927

Gautier M., Khalil W. and Restrepo P. P., 1995. “Identification of the dynamic parameters of a closed loop robot”, *Proc. IEEE on Int. Conf. on Robotics and Automation*, Nagoya, may 1995, p. 3045-3050

Gilson M., Garnier H., Young P.C. and Van den Hof P., 2006. “A refined IV method for closed loop system identification”, In: *14th IFAC Symposium on System Identification, SYSID 2006*, Newcastle Australia, March 2006, pp. 903-908

Hampel F.R., 1971. “A general qualitative definition of robustness”, *Annals of Mathematical Statistics*, 42, 1887-1896, 1971

Huber P.J., 1981. “Robust statistics”, *Wiley*, New-York, 1981

Janot A., Bidard C., Gautier M., Keller D. and Perrot Y., 2006. “Modélisation, Identification d’une Interface Médicale”, *Journées Identification et Modélisation Expérimentale 2006*, Poitiers, November 2006

Khatounian F., Moreau S., Monmasson E., Janot A. and Louveau F., 2006. “Simultaneous Identification of the Inertial Rotor Position and Electrical Parameters of a PMSM for a Haptic Interface”, *In: 12th EPE-PEMC Conference*, Portoroz, Slovenia, August September 2006, CD-ROM, ISBN 1-4244-0121-6

Lemaire Ch.E., Vandanjon P.O., Gautier M. and Lemaire Ch., 2006. “Dynamic Identification of a Vibratory Asphalt Compactor for Contact Efforts Estimation”, In: *14th IFAC Symposium on System Identification, SYSID 2006*, Newcastle Australia, March 2006,pp. 973-978

Mili L., Phaniraj V. and Rousseeuw P.J., 1990. “Robust estimation theory for bad data diagnostics in electrical power systems”, *In: C. T. Leondes editor, Control and Dynamic Systems:Advances in Theory and Applications*, pp. 271-325, San Diego 1990, Academic Press

Mili L., Cheniae M.G., Vichare N.S. and Rousseeuw P.J., 1996. “Robust state estimation based on projection statistics”, IEEE Trans. On Power Systems, 11(2), 1118-1127, 1996

Specht R., and Isermann R., 1988. “On-line identification of inertia, friction and gravitational forces applied to an industrial robot”, *Proc. IFAC Symp. on Robot Control, SYROCO’88*, 1988, p. 88.1-88.6

Young P.C. and Jakeman A.J., 1979. Refined instrumental variable methods of time-series analysis: Part 1, SISO systems, *International Journal of Control*, 29:1-30, 1979

Young P.C., 2006. “An instrumental variable approach to ARMA model identification and estimation”, In: *14th IFAC Symposium on System Identification, SYSID 2006*, Newcastle Australia, March 2006, pp. 410-415

SPECIAL SESSION ON INTELLIGENT VEHICLE CONTROL SYSTEMS

CHAIR:
OLEG GUSIKHIN

VEHICLE MODELS AND ESTIMATION OF CONTACT FORCES AND TIRE ROAD FRICTION

Nacer K. M'Sirdi
LSIS, CNRS UMR 6168. Dom. Univ. St Jrme, Av. Escadrille Normandie - Niemen 13397, Marseille Cedex 20, France
nacer.msirdi@lsis.org; www.nkms.free.fr

Abdelhamid Rabhi
C.R.E.A, 7 Rue du Moulin Neuf, 80000 Amiens, France

Aziz Naamane
LSIS, CNRS UMR 6168. Dom. Univ. St Jrme, Av. Escadrille Normandie - Niemen 13397, Marseille Cedex 20, France

Keywords: Vehicle dynamics, Sliding Modes observer, Robust nonlinear observers, tire forces estimation.

Abstract: In this paper a 16 DoF vehicle model is presented and discussed. Then some partial models are considered and justified for the design of robust estimators using sliding mode approach in order to identify the tire-road friction or input variables. The estimations produced are based on split system equations in as cascaded observers and estimators. The first produces estimations of vehicle states.

1 INTRODUCTION

In recent years, the increasing demand for the safety in car vehicles has promoted research and development of the technology of active safety. However more and more new active safety systems are developed and installed on vehicles for real-time monitoring and observers for controlling the dynamic stability (EBS, ABS, ESP).

Car accidents occur for several reasons which may involve the driver or vehicle components or environment. Such situations appears when the vehicle is driven beyond the adherence or stability limits. One of the important factors determining vehicle dynamics including safety is road friction and the tire forces (ground-vehicle interactions). In general partial and approximated models are used. They are not fully justified and their validity is often limited. In this work we try to highlight some of the approximations made and give some details allowing to evaluate what is really neglected.

Robust observers looking forward are based on the physics of interacting systems (the vehicle, the driver and the road). However, tire forces and road friction are difficult to measure directly and to represent precisely by some deterministic model equations. In the literature, their values are often deduced by some experimentally approximated models (Gustafsson). The knowledge of tire parameters and variables (forces, velocities, wheel and slip), tire forces is essential to advanced vehicle control systems such as anti-lock braking systems (ABS), traction control systems (TCS) and electronic stability program (Ackermann)(Msirdi04)(Canudas03). Recently, many analytical and experimental studies have been performed on estimation of the frictions and contact forces between tires and road.

We focus our work, as presented in this paper, first on modeling and second on on-line estimation of the tires forces (Ackermann)(Msirdi04). We estimate the vehicle state and identify tire forces ((Msirdi03)). The main contribution is the emphasize of the rational behind partial approximated models and the on-line estimation of the tire force needed for control.

Tire forces can be represented by the nonlinear (stochastic) functions of wheel slip. The deterministic tire models encountered are complex and depend on several factors (as load, tire pressure, environmental characteristics, etc.). This makes on line estimation of forces and parameters difficult for vehicle control applications and detection and diagnosis for driving monitoring and surveillance. In (Drakunov)(Canudas03), application of sliding mode control is proposed. Observers based on the sliding mode approach have been also used in (Rabhi04).

In (Ray) an estimation method is based on the least squares algorithm and combined with a Kalman filter to estimate the contact forces. The paper of (Gustafsson) presents an estimation of tire/road frictions by means of a Kalman filter. It gives relevant estimates of the slope of μ versus slip (λ). In (Carlson) estimations of longitudinal stiffness and wheel effective radius are proposed using vehicle sensors and a GPS for low values of the slip.

Robust observers with unknown inputs have been shown to be efficient for estimation of road profile (Imine) and for estimation of the contact forces (Msirdi04)(Rabhi04). Tracking and braking control reduce wheel slip. This can be done also by means of its regulation while using sliding mode approach for observation and control.(Msirdi04)(Rabhi04). This enhances the road safety leading better vehicle adherence and maneuvers ability. The vehicle controllability in its environment along the road admissible trajectories remain an important open research problem.

The proposed estimation procedure has to be robust enough to avoid model complexity. It can then be used to detect some critical driving situations in order to improve the security. This approach can be used also in several vehicle control systems such as Anti-look Brake Systems (ABS), traction control system (TCS), diagnosis systems, etc... The main characteristics of the vehicle longitudinal dynamics were taken into account in the developed model used to design robust observer and estimations. The estimations are produced using only the angular wheel position as measurement by the specially designed robust observer based on the super-twisting second-order sliding mode. The proposed estimation method is verified through simulation of one- wheel model (with a "Magic formula" as tire model). In a second step of validation we present some application results (on a Peugeot 406) showing an excellent reconstruction of the velocities, tire forces and estimation of wheel radius.

2 VEHICLE MODELING

2.1 Complete 16 DoF Model

In literature, many studies deal with vehicle modeling (Kien)(Ramirez)(Mendoza). This kind of systems are complex and nonlinear composed with many coupled subsystems: wheels, motor and system of braking, suspensions, steering, more and more inboard and imbedded electronics. Let us represent the vehicle (like eg a Peugeot 406) by the scheme of figure 1 and define the following notations.

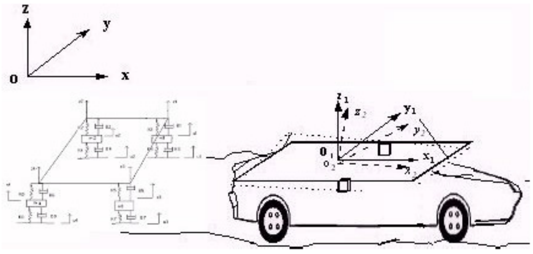


Figure 1: Vehicle dynamics and reference frames.

The vehicle body receives as excitations external forces and moments following the three axes: - Longitudinal, - Lateral, - Vertical. These come from interaction of the wheels and road, from perturbations (wind for example), gravity and vehicle drive line. Let us consider the basic reference fixed frame R. We can consider the vehicle as made of 5 sub-systems: chassis whit 6 DoF and then 4 wheels with their suspensions. Each of the rear wheels has 2 DoF. The front ones are driven wheels with 3 DoF each. Then we have 16 DoF. Let the generalized variables be in the vector $q \in R^{16}$, defined as

$$q^T = [x, y, z, \theta_z, \theta_y, \theta_x, z_1, z_2, z_3, z_4, \delta_3, \delta_4, \varphi_1, \varphi_2, \varphi_3, \varphi_4]$$

where x, y, et z represent displacements in longitudinal, lateral and vertical direction. angles of roll, pitch and yaw are θ_x, θ_y et θ_z respectively. The suspensions elongations are noted z_i: $(i = 1..4)$. δ_i: stands for the steering angles (for wheels numbered as $i = 3,4$), finally φ_i: are angles wheels rotations $(i = 1..4.)$. Vectors $\dot{q}, \ddot{q} \in R^{16}$ are respectively velocities and corresponding accelerations. $M(q)$ is the inertia matrix and $C(q,\dot{q})\dot{q}$ are coriolis and centrifugal forces. The gravity term is G . Suspensions forces are $V(q,\dot{q}) = K_v\dot{q} + K_p q$ with respectively damping and stiffness matrices K_v , K_p. We can define as dynamic equations of the vehicle by applying the principles fundamental of the dynamics (see (Beurier)):

$$\Gamma + J^T F = M\ddot{q} + C(q,\dot{q})\dot{q} + Kq + G \qquad (1)$$

with as parameters only to give an idea

$$M = \begin{bmatrix} \bar{M}_{1,1} & \bar{M}_{1,2} & \bar{M}_{1,3} & 0 & 0 \\ \bar{M}_{2,1} & \bar{M}_{2,2} & \bar{M}_{2,3} & \bar{M}_{2,4} & \bar{M}_{2,5} \\ \bar{M}_{3,1} & \bar{M}_{3,2} & \bar{M}_{3,3} & 0 & 0 \\ 0 & \bar{M}_{4,2} & 0 & \bar{M}_{4,4} & 0 \\ 0 & \bar{M}_{5,2} & 0 & 0 & \bar{M}_{5,5} \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & \bar{C}_{12} & \bar{C}_{13} & 0 & 0 \\ 0 & \bar{C}_{22} & \bar{C}_{23} & \bar{C}_{24} & \bar{C}_{25} \\ 0 & \bar{C}_{32} & \bar{C}_{33} & 0 & 0 \\ 0 & \bar{C}_{42} & 0 & 0 & 0 \\ 0 & \bar{C}_{52} & 0 & 0 & \bar{C}_{55} \end{bmatrix}$$

$$\text{and } J^T = \begin{bmatrix} J_{1,1} & J_{1,2} & J_{1,3} & J_{1,4} \\ J_{2,1} & J_{2,2} & J_{2,3} & J_{2,4} \\ J_{3,1} & J_{3,2} & J_{3,3} & 0 \\ 0 & 0 & 0 & J_{4,4} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

This is just to show that we can decompose our system as coupled subsystems. Let us say five coupled subsystems, that we have considered in our previous works. This has been computed using a symbolic computation software considering 16 generalized variables: 6 for position and orientation of body, 4 as suspensions ones, 2 for front wheels steering and 4 as wheels rotations. The matrices M, C and K are of dimensions 16×16. F is input forces vector acting on wheels, it has 12 components(3 forces (longitudinal, lateral and normal) × 4 wheels), Γ represent extra inputs for perturbations.. In the following application this model has been reduced and simplified assuming as nominal behavior a normal driving situation (Msirdi03).

2.2 Coupled Sub Models

We can split the previous model, without approximations, in five parts as follows. This leads us to the body's translations dynamics:

$$\begin{bmatrix} F_{LT} \\ F_{TT} \\ F_{NT} \end{bmatrix} = \bar{M}_{11} \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} + \bar{M}_{12} \begin{bmatrix} \ddot{q}_4 \\ \ddot{q}_5 \\ \ddot{q}_6 \end{bmatrix} + \qquad (2)$$

$$+\bar{M}_{13} \begin{bmatrix} \ddot{q}_{11} \\ \ddot{q}_{21} \\ \ddot{q}_{31} \\ \ddot{q}_{41} \end{bmatrix} + \bar{C}_{12} \begin{bmatrix} \dot{q}_4 \\ \dot{q}_5 \\ \dot{q}_6 \end{bmatrix} \qquad (3)$$

Rotations and orientation motions of the body:

$$\begin{bmatrix} J_{4F} \\ J_{5F} \\ J_{6F} \end{bmatrix} = \bar{M}_{12}^T \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix} + \bar{M}_{22} \begin{bmatrix} \ddot{q}_4 \\ \ddot{q}_5 \\ \ddot{q}_6 \end{bmatrix} +$$

$$+\bar{M}_{23} \begin{bmatrix} \ddot{q}_{11} \\ \ddot{q}_{21} \\ \ddot{q}_{31} \\ \ddot{q}_{41} \end{bmatrix} + \bar{M}_{24} \begin{bmatrix} \ddot{q}_{12} \\ \ddot{q}_{22} \\ \ddot{q}_{32} \end{bmatrix} + \bar{M}_{25} \begin{bmatrix} \ddot{q}_{42} \\ \ddot{q}_{33} \\ \ddot{q}_{43} \end{bmatrix} +$$

$$+\bar{C}_{22} \begin{bmatrix} \dot{q}_4 \\ \dot{q}_5 \\ \dot{q}_6 \end{bmatrix} + \bar{C}_{23} \begin{bmatrix} \dot{q}_{11} \\ \dot{q}_{21} \\ \dot{q}_{31} \\ \dot{q}_{41} \end{bmatrix} +$$

$$+\bar{C}_{24} \begin{bmatrix} \dot{q}_{12} \\ \dot{q}_{22} \\ \dot{q}_{32} \end{bmatrix} + \bar{C}_{25} \begin{bmatrix} \dot{q}_{42} \\ \dot{q}_{33} \\ \dot{q}_{43} \end{bmatrix}$$

Suspensions dynamics:

$$\begin{bmatrix} J_{7F} \\ J_{8F} \\ J_{9F} \\ J_{10F} \end{bmatrix} = \bar{M}_{1,3} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix} + \bar{M}_{2,3} \begin{bmatrix} \ddot{q}_4 \\ \ddot{q}_5 \\ \ddot{q}_6 \end{bmatrix} +$$

$$+\bar{M}_{3,3} \begin{bmatrix} \ddot{q}_{11} \\ \ddot{q}_{21} \\ \ddot{q}_{31} \\ \ddot{q}_{41} \end{bmatrix} + \bar{C}_{32} \begin{bmatrix} \dot{q}_4 \\ \dot{q}_5 \\ \dot{q}_6 \end{bmatrix} + \bar{C}_{33} \begin{bmatrix} \dot{q}_{11} \\ \dot{q}_{21} \\ \dot{q}_{31} \\ \dot{q}_{41} \end{bmatrix} +$$

$$+ \begin{bmatrix} R_{11} & & & \\ & R_{21} & & \\ & & R_{31} & \\ & & & R_{41} \end{bmatrix} \begin{bmatrix} q_{11} \\ q_{21} \\ q_{31} \\ q_{41} \end{bmatrix} + \begin{bmatrix} G_7 \\ G_8 \\ G_9 \\ G_{10} \end{bmatrix} \qquad (4)$$

The previous 16 DoF model is then equivalent to:

$$\begin{bmatrix} F_T \\ F_2 \\ F_3 \\ 0 \\ U_5 \end{bmatrix} = M \begin{bmatrix} \ddot{\xi}_1 \\ \ddot{\xi}_2 \\ \ddot{\xi}_3 \\ \ddot{\xi}_4 \\ \ddot{\xi}_5 \end{bmatrix} + C \begin{bmatrix} \dot{\xi}_1 \\ \dot{\xi}_2 \\ \dot{\xi}_3 \\ \dot{\xi}_4 \\ \dot{\xi}_5 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ R_{33}.\xi_3 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} G_1 \\ G_2 \\ G_3 \\ 0 \\ 0 \end{bmatrix}$$

In the last expression, we can remark that splitting the model can be realized and this model is helpful, when using reduced models, to identify what is neglected regard to our proposed nominal model with 16 DoF. The dynamic equations can be reduced, in case where we assume that motion is normal driving in a normal staith road, to translations and rotations of the body, and wheels plus suspension motions. For translations we find often in literature:

$$\begin{aligned} m\dot{v}_x &= \sum F_x \\ m\dot{v}_y &= \sum F_y \\ m\dot{v}_z &= \sum F_z \end{aligned} \qquad (5)$$

where m is the total mass of the vehicle and $v = [v_x, v_y, v_z]^T$ describe the vehicle velocities along x, y, z. In the left hand side of this approximate model are the forces $\sum F_x$, $\sum F_y$ and $\sum F_z$ applied in directions of x, y and z and the balance of the moments $(\sum M_x, \sum M_y, \sum M_z)$, give rotations following the three directions x, y and z, is given by:

$$J \begin{bmatrix} \ddot{\theta} \\ \ddot{\phi} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} \sum M_x \\ \sum M_y \\ \sum M_z \end{bmatrix} \qquad (6)$$

The wheel angular motions can be written:

$$\begin{aligned} \dot{\omega}_{fl} &= \frac{1}{I_\omega}(C_{fl} - R_\omega F_{xf1}) \qquad (7) \\ \dot{\omega}_{fr} &= \frac{1}{I_\omega}(C_{fr} - R_\omega F_{xf2}) \\ \dot{\omega}_{rl} &= \frac{1}{I_\omega}(C_{rl} - R_\omega F_{xr1}) \\ \dot{\omega}_{rr} &= \frac{1}{I_\omega}(C_{rr} - R_\omega F_{xr2}) \end{aligned}$$

with ω_f and ω_r are the rotation velocities of the front and rear wheel, C_{mi} is the motor couple applied at wheel i and T_i is the braking couple applied at wheel i. Let r_1 be the distance between the center of gravity and the front axis and r_2 the distance between the center of gravity and the rear axis.

2.3 Partial Models

The complete model is difficult to use in control applications. It involves several variables which are not available for measurement or not observable. The most part of applications deal with simplified and partial models. Let us consider, for our robust observer, the simplified motion dynamics of a quarter-vehicle model, capturing only nominal behavior (Msirdi04) (Msirdi03). This model retains the main characteristics useful for the longitudinal dynamic. For a global application, this method can be easily extended to the complete vehicle and involve the four coupled wheels. The amount of neglected parts in the modeling can be considered to evaluate robustness of proposed estimators.

Applying Newton's law to one isolated wheel gives:.

$$\begin{aligned} m\dot{v}_x &= F_x \\ J_r\dot{\omega} &= T - rF_x \end{aligned}$$

where m is the vehicle mass and J_r, r are the inertia and effective radius of the tire respectively. v_s is the linear velocity of the vehicle, ω is the angular velocity of the considered wheel. T is the accelerating (or braking) torque, and F_x is the tire/road friction force. The tractive (respectively braking) force, produced at the tire/road interface when a driving (braking) torque is applied to pneumatic tire, has opposite direction of relative motion between the tire and road surface. This relative motion exhibits the tire slip properties. The wheel-slip is due to deflection in the contact patch. The longitudinal wheel slip is generally called the slip ratio and is described by a kinematic relation as (Carlson).

$$\lambda = \frac{|v_r - v_x|}{\max(v_r, v_x)} \tag{8}$$

where v_r is the wheel velocity. Representing the adhesion coefficient as a function of the wheel slip yields the adhesion characteristic $\mu(\lambda)$, which depends on the road surfaces as shown in the following figure 2.

The figure 2 shows the relations between coefficient of road adhesion μ and longitudinal slip λ for different road surface conditions. It can be observed that all curves $\mu(\lambda)$ start at $\mu = 0$ for zero slip, which corresponds to the non-braking and non accelerating, free rolling wheel. With a linear increasing slip ratio from 3% to 20%. Beyond this maximum value the slope of the adhesion characteristic is maximum and then slope becomes negative. At a slip ratio of 100% the wheel is completely skidding, which corresponds to the locking of the wheel. The adhesion characteristic plays an essential role for both the design and the validation of ABS. Overall, to improve the performance of an ABS it is desirable to have some real-time information about the adhesion characteristic.

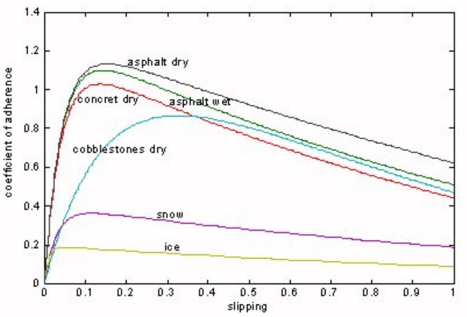


Figure 2: Wheel slip.

By assuming that the longitudinal forces are proportional to the transversal ones, we can expressed theses forces as follows, where F_z is the vertical force of the wheel.

$$F_x = \mu F_z \tag{9}$$

The vertical forces that we use in our model are function of the longitudinal acceleration and the height of the center of gravity. The vertical force can be represented as:

$$Fz = \frac{m}{2(l_f + l_r)}(gl_r - h\dot{v}_x) \tag{10}$$

where h is the height of the center of gravity, l_f is the distance between the center of gravity and the front axis center of gravity and l_r is the distance between the center of gravity and the rear axis center of gravity.

3 OBSERVER DESIGN

The sliding mode technique is an attractive approach (Davila). The primary characteristic of SMC is that the feedback signal is discontinuous, switching on one or several manifolds in the state-space. In what follows, we use a second order differentiator in order to derive robust estimates of the tire road friction.

3.1 High Order Sliding Mode Observer (HOSM)

In this part we will use a Robust Differentiation Estimator (RDE) to deduce our robust estimations. Consider a smooth dynamics function, $s(x) \in \mathbb{R}$. The system containing this variable may be closed by some possibly-dynamical discontinuous feedback where the control task may be to keep the output $s(x(t)) = 0$. Then provided that successive total time derivatives $s, \dot{s}, \ddot{s} ... s^{(r-1)}$ are continuous functions of the closed system state space variables, and the t-sliding point set is non-empty and consist locally of Filippov trajectories.

$$s = \dot{s} = \ddot{s} = ... = s^{(r-1)} = 0 \tag{11}$$

is non-empty and consists locally of Filippov trajectories. The motion on set (Filippov)(Utkin99) is called r-sliding mode (rth-order sliding mode) (Orlov)(Levant).

The HOSM dynamics converge toward the origin of surface coordinates in finite time always that the order of the sliding controller is equal or bigger than the sum of a relative degree of the plant and the actuator. To estimate the derivatives s_1 and s_2 without its direct calculations of derivatives, we will use the 2^{nd}-order exact robust differentiator of the form (Levant)

$$\begin{aligned}
\dot{z}_0 &= v_0 = z_1 - \lambda_0 |z_0 - s_\omega|^{\frac{2}{3}} sign(z_0 - s_\omega) \\
\dot{z}_1 &= v_1 = -\lambda_1 sign(z_1 - v_0)^{\frac{1}{2}} sign(z_1 - v_0) + z_2 \\
\dot{z}_2 &= -\lambda_2 sign(z_2 - v_1)
\end{aligned}$$

where z_0, z_1 and z_2 are the estimate of s_ω, s_1 and s_2, respectively, $\lambda_i > 0$, $i = 0, 1, 2$. Under condition $\lambda_0 > \lambda_1 > \lambda_2$ the third order sliding mode motion will be established in a finite time. The obtained estimates are $z_1 = s_1 = \dot{s}_\omega$ and $z_2 = s_2 = \ddot{s}_\omega$ then they can be used in the estimation of the state variables and also in the control.

3.2 Cascaded Observers - Estimators

In this section we use the previous approach to build an estimation scheme allowing to identify the tire road friction. The estimations will be produced in three steps, as cascaded observers and estimator, reconstruction of information and system states step by step. This approach allow us to avoid the observability problems dealing with inappropriate use of the complete modeling equations. For vehicle systems it is very hard to build up a complete and appropriate model for global observation of all the system states in one step. Thus in our work, we avoid this problem by means of use of simple and cascaded models suitable for robust observers design.

The first step produces estimations of velocities. The second one estimate the tire forces (vertical and longitudinal ones) and the last step reconstruct the friction coefficient.

The robust differentiation observer is used for estimation of the velocities and accelerations of the wheels. The wheels angular positions and the velocity of the vehicles body v_x, are assumed available for measurements. The previous Robust Estimator is useful for retrieval of the velocities and accelerations.

1^{st} Step:

$$\begin{aligned}
\dot{\widehat{\theta}} &= v_0 = \widehat{\omega} - \lambda_0 \left|\theta - \widehat{\theta}\right|^{\frac{2}{3}} sign(\theta - \widehat{\theta}) \\
\dot{\widehat{\omega}} &= v_1 = \dot{\widehat{\omega}} - \lambda_1 sign(\widehat{\omega} - v_0)^{\frac{1}{2}} sign(\widehat{\omega} - v_0) \\
\ddot{\widehat{\omega}} &= -\lambda_2 sign(\dot{\widehat{\omega}} - v_1)
\end{aligned}$$

The convergence of these estimates is guaranteed in finite time t_0.

2^{nd} Step: In the second step we can estimate the forces F_x and F_z. Then to estimate F_x we use the following equation,

$$J\dot{\widehat{\omega}} = T - R_{ef}\widehat{F_x} \tag{12}$$

In the simplest way, assuming the input torques known, we can reconstruct F_x as follows:

$$\widehat{F_x} = \frac{(T - J\dot{\widehat{\omega}})}{R_{ef}} \tag{13}$$

$\dot{\widehat{\omega}}$ is produced by the Robust Estimator (RE). Note that any estimator with output error can also be used to enhance robustness versus noise. In our work, in progress actually, the torque T will be also estimated by means of use of additional equation from engine behavior related to accelerating inputs.

After those estimations, their use in the same time with the system equations allow us to retrieve de vertical forces F_z as follows. To estimate F_z we use the following equation

$$\widehat{F_z} = \frac{m}{2(l_f + l_r)}(gl_r - h.\dot{\widehat{v}_x}) \tag{14}$$

$\widehat{v}_x$ is produced by the RE.

3^{rd} Step: At this step it only remains to estimate the adherence or friction coefficient. To this end we assume the vehicle rolling in a normal or steady state situation in order to be able to approximate this coefficient by the following formula

$$\widehat{\mu} = \frac{\widehat{F_x}}{\widehat{F_z}} \tag{15}$$

4 SIMULATION AND EXPERIMENTAL RESULTS

In this section, we give some realistic simulation results in order to test and validate our approach and the proposed observer. In simulation, the state and forces are generated by use of a car simulator called VeDyna (VEDYNA). In this simulator the model involved is more complex than the one of 16 DoF presented in the first part of the paper. Comparind the simplified model to the 16 DoF one, let us evaluate the robustness of estimation The VeDyna simulated brake torque is shown in figure 3.

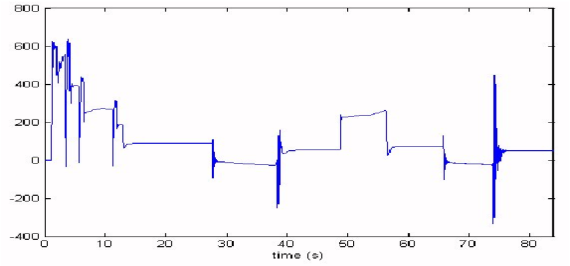


Figure 3: Braking torque.

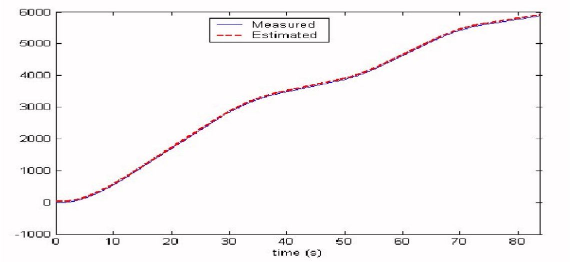


Figure 4: Angular displacements.

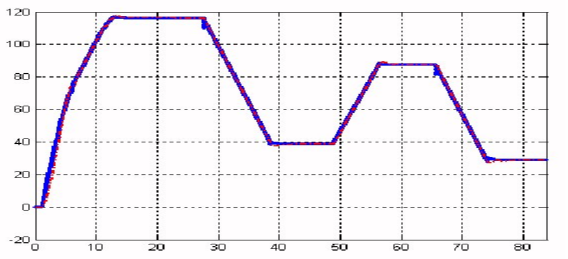


Figure 5: Angular Velocity.

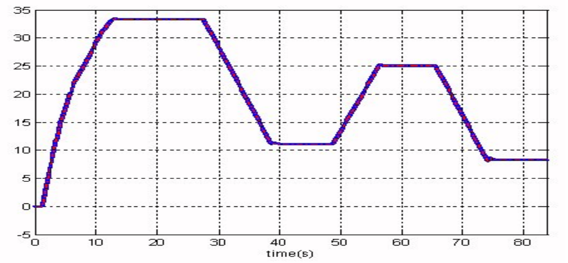


Figure 6: Vehicle velocity.

Figure 4 shows the measured and estimated wheel angular position. This signal is used to estimate velocities and accelerations. Figure 4 shows the estimated wheel angles. In the figure 6, we represent the estimation of vehicle velocity. The figure shows the good convergence to the actual vehicle velocity. Figure 7 shows the obtained vehicle acceleration. The observer allows a good estimation of angular velocity and acceleration. The last step gives us the estimated longitudinal forces F_x and normal forces F_z which are presented in figure 8 and 9. Finally road friction coefficient is deduced and presented in figure (10).

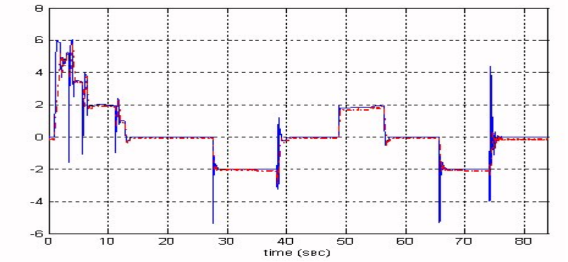


Figure 7: Estimated and measured acceleration.

5 CONCLUSION

In this work we have tried to highlight all approximations made in general when using simplified models and this paper gives some details allowing to evaluate what is really neglected. In second part od this paper, we have proposed an efficient and robust estimator baser on the second order sliding mode differentiator. This is used to build an estimation scheme allowing to identify the tire road frictions and input forces which are non observable when using the complete model

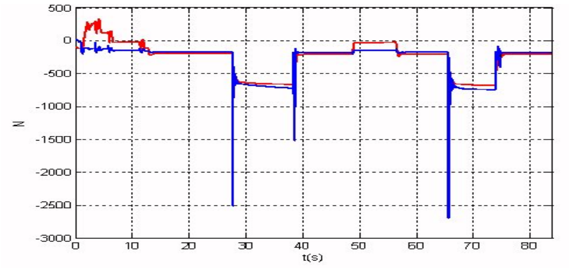


Figure 8: Longitudinal force.

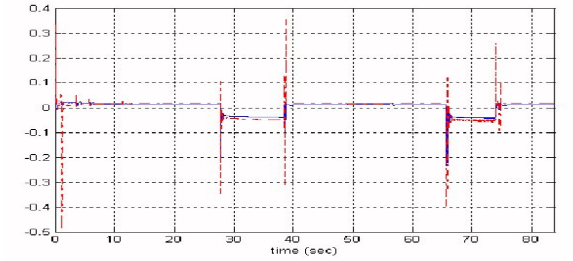


Figure 10: Road friction.

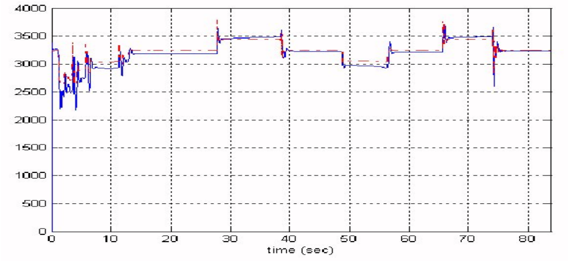


Figure 9: Normal force F_z.

and standard sensors. The estimations produced finite time converging measurements of model inputs, in three steps by cascaded observers and estimators. This method shows very good performances in simulations conducted using a more complex model (than the 16 DoF one) involved in VeDyna car simulator. Tire forces (vertical and longitudinal ones) are also estimated correctly. Simulation results are presented to illustrate the ability of this approach to give estimation of both vehicle states and tire forces. The robustness versus uncertainties on model parameters and neglected dynamics has also been emphasized in simulations. Application of this approach with inclusion of torque estimation using a simplified model for the engine behavior, is in progress.

ACKNOWLEDGEMENTS

This work has been done in a collaboration managed by members of the LSIS inside the GTAA (research group supported by the CNRS).

Thanks are addressed to the LCPC of Nantes for experimental data and the trials with their vehicle Peugeot 406.

REFERENCES

F. Gustafsson, ”Slip-based tire-road friction estimation”, Automatica, vol 33, no. 6, pp. 1087-1097, 1997.

J. Ackermann ”Robust control prevents car skidding. IEEE Control systems magazine, V17, N3, pp23-31, 1997

N. K. M’sirdi, A. Rabhi, N. Zbiri and Y. Delanne. VRIM: Vehicle Road Interaction Modelling for Estimation of Contact Forces. Accepted for TMVDA 04. 3rd Int. Tyre Colloquium Tyre Models For Vehicle Dynamics Analysis August 30-31, 2004 University of Technology Vienna, Austria

C. Canudas de Wit, P.Tsiotras, E.Velenis, M.Basset, G.Gissinger. Dynamic Friction Models for Road/Tire Longitudinal Interaction. Vehicle Syst. Dynamics 2003. V39, N3, pp 189-226.

Laura Ray. Nonlinear state and tire force estimation for advanced vehicle control. IEEE T on control systems technology, V3, N1, pp117-124, march 1995.

Nacer K. M’Sirdi. Observateurs robustes et estimateurs pour l’estimation de la dynamique des véhicules et du contact pneu - route. JAA. Bordeaux, 5-6 Nov 2003

S. Drakunov, U. Ozguner, P. Dix and B. Ashrafi. ABS control using optimum search via sliding modes. IEEE Trans. Control Systems Technology, V 3, pp 79-85, March 1995.

A. Rabhi, H. Imine, N. M’ Sirdi and Y. Delanne. Observers With Unknown Inputs to Estimate Contact Forces and Road Profile AVCS’04 International Conference on Advances in Vehicle Control and Safety Genova - Italy, October 28-31 2004

Christopher R. Carlson. Estimation With Applications for Automobile Dead Reckoning and Control. PhD thesis, University of STANDFOR 2003.

U. Kiencken, L. Nielsen. Automotive Control Systems. Springer, Berlin, 2000.

Mendoza, Sur la modèlisation et la commande des véhicules automobiles, Thèse, Juillet 1997.

H. Imine, N. M’Sirdi, L. Laval, Y. Delanne - Sliding Mode Observers for Systems with Unknown Inputs: Application to estimate the Road Profile. A paraître dans ASME, Journal of Dynamic Systems, Measurement and Control en mars 2003.

R. Ramirez-Mendoza. Sur la modélisation et la commande des véhicules automobiles. PHD thesis Institut National Polytechnique de Grenoble, Laboratoire d Automatique de Grenoble 1997.

G. Beurier. Modélisation et la commande de système. PHD thesis LRP. UPMC Paris 6, 1998.

J. Davila and L. Fridman. "Observation and Identification of Mechanical Systems via Second Order Sliding Modes", 8th. International Workshop on Variable Structure Systems,September 2004, Espana

A. Levant, "Robust exact differentiation via sliding mode technique", Automatica, vol. 34(3), 1998, pp 379-384.

A. F. Filippov, Differential Equations with Discontinuous Right-hand Sides, Dordrecht, The Netherlands:Kluwer Academic Publishers; 1988.

V. Utkin, J. Guldner, J. Shi, Sliding Mode Control in Electromechanical Systems, London, UK:Taylor & Francis; 1999.

J. Alvarez, Y. Orlov, and L. Acho, "An invariance principle for discontinuous dynamic systems with application to a coulomb friction oscillator", Journal of Dynamic Systems, Measurement, and Control, vol. 122, 2000, pp 687-690.

Levant, A. Higher-order sliding modes, differentiation and output-feedback control, International Journal of Control, 2003, Vol.76, pp.924-941

Simulator VE-DYNA. www.tesis.de/index.php.

APPENDIX

Definition of the matrices involved.in the model.

$$\bar{M}_{11}= \begin{bmatrix} M_{1,1} & 0 & 0 \\ 0 & M_{2,2} & 0 \\ 0 & 0 & M_{3,3} \end{bmatrix}$$

$$\bar{M}_{12} = \bar{M}^T_{21}= \begin{bmatrix} M_{1,4} & M_{1,5} & M_{1,6} \\ M_{2,4} & M_{3,5} & M_{2,6} \\ 0 & M_{3,5} & M_{3,6} \end{bmatrix};$$

$$\bar{M}_{13} = \bar{M}^T_{31} = \begin{bmatrix} M_{1,7} & M_{1,8} & M_{1,9} & M_{1,10} \\ M_{2,7} & M_{2,8} & M_{2,9} & M_{2,10} \\ M_{3,7} & M_{3,8} & M_{3,9} & M_{3,10} \end{bmatrix};$$

$$\bar{M}_{23} = \bar{M}^t_{32} = \begin{bmatrix} M_{4,7} & M_{4,8} & M_{4,9} & M_{4,10} \\ M_{5,7} & M_{5,8} & M_{5,9} & M_{5,10} \\ M_{6,7} & M_{6,8} & M_{6,9} & M_{6,10} \end{bmatrix}$$

$$\bar{M}_{24} = \bar{M}^T_{42} = \begin{bmatrix} M_{4,11} & M_{4,12} & M_{4,13} \\ M_{5,11} & M_{5,12} & M_{5,13} \\ 0 & 0 & 0 \end{bmatrix}$$

$$\bar{M}_{2,5} = \bar{M}^T_{52} \begin{bmatrix} M_{4,14} & M_{4,15} & M_{4,16} \\ M_{5,14} & M_{5,15} & M_{5,16} \\ 0 & M_{6,15} & M_{6,16} \end{bmatrix}$$

$$\bar{M}_{2,2} = \begin{bmatrix} M_{4,4} & M_{4,5} & M_{4,6} \\ M_{5,4} & M_{5,5} & M_{5,6} \\ M_{6,4} & M_{6,5} & M_{6,6} \end{bmatrix};$$

$$\bar{M}_{3,3} = \begin{bmatrix} M_{7,7} & 0 & 0 & 0 \\ 0 & M_{8,8} & 0 & 0 \\ 0 & 0 & M_{9,9} & 0 \\ 0 & 0 & 0 & M_{10,10} \end{bmatrix};$$

$$\bar{M}_{4,4} = \begin{bmatrix} M_{11,11} & 0 & 0 \\ 0 & M_{12,12} & 0 \\ 0 & 0 & M_{13,13} \end{bmatrix}$$

$$\bar{M}_{5,5} = \begin{bmatrix} M_{14,14} & 0 & 0 \\ 0 & M_{15,15} & 0 \\ 0 & 0 & M_{16,16} \end{bmatrix}$$

BENCHMARKING HAAR AND HISTOGRAMS OF ORIENTED GRADIENTS FEATURES APPLIED TO VEHICLE DETECTION

Pablo Negri, Xavier Clady and Lionel Prevost
Université Pierre et Marie Curie-Paris 6, ISIR, CNRS FRE 2507
3 rue Galilée, 94200 Ivry sur Seine, France
pablo . negri [a] lisif.jussieu.fr

Keywords: Intelligent vehicle, vehicle detection, AdaBoost, Haar filter, Histogram of oriented gradient.

Abstract: This paper provides a comparison between two of the most used visual descriptors (image features) nowadays in the field of object detection. The investigated image features involved the Haar filters and the Histogram of Oriented Gradients (HoG) applied for the on road vehicle detection. Tests are very encouraging with a average detection of 96% on realistic on-road vehicle images.

1 INTRODUCTION

On road vehicle detection is an essential part of the Intelligent Vehicles Systems and has many applications including platooning (i.e. vehicles travelling in high speed and close distances in highways), Stop&Go (similar that precedent situation, but at low speeds), and autonomous driving.

Most of the detecting methods distinguish two basic steps: Hypothesis Generation (HG) and Hypothesis Verification (HV) (Sun et al., 2006). HG approaches are simple low level algorithm used to locate potential vehicle locations and can be classified in three categories:

- knowledge-based: symmetry (Bensrhair et al., 2001), colour (Xiong and Debrunner, 2004; Guo et al., 2000), shadows (van Leeuwen and Groen, 2001), edges (Dellaert, 1997), corners (Bertozzi et al., 1997), texture (Bucher et al., 2003), etc.,
- stereo-based: disparity map (Franke, 2000), inverse perspective mapping (Bertozzi and Broggi, 1997), etc,
- and motion-based (Demonceaux et al., 2004).

HV approaches perform the validation of the Regions of Interest generated by the HG step. They can be classified in two categories: template-based and appearance-based. Template-based methods perform a correlation between a predefined pattern of the vehicle class and the input image: horizontal and vertical edges (Srinivasa, 2002), regions, deformable patterns (Collado et al., 2004) and rigid patterns (Yang et al., 2001). Appearance-based methods learn the characteristics of the classes vehicle and non-vehicle from a set of training images. Each training image is represented by a set of local or global descriptors (features) (Agarwal et al., 2004). Then, classification algorithms can estimate the decision boundary between the two classes.

One of the drawbacks of optical sensors are the considerable time processing and the average robustness. In that way, Viola & Jones (Viola and Jones, 2001) developed simple an appearance-based system obtaining amazing results in real time. Their appearance based method uses Haar-based representation, combined with an AdaBoost algorithm (Freund and Schapire, 1996). They also introduce the concept of a cascade of classifiers which reaches high detection results while reducing computation time.

The present article compares the Haar-based features with the Histograms of Oriented Gradient (HoG) based features using the same cascade architecture.

The next section describes briefly the Haar and the HoG features. Section two introduces the learning classification algorithms based on AdaBoost. We finish the article with the results and conclusions.

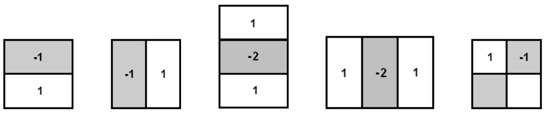


Figure 1: 2D Wavelet set.

2 FEATURES

The common reasons why features are choosen instead of pixels values are that features can code high level object information (segments, texture, ...) while intensity pixel values based system operates slower than a feature based system. This section describes the features used to train the Adaboost cascade.

2.1 Haar Filters

Each wavelet coefficient describes the relationship between the average intensities of two neigh-boring regions. Papageorgiou *et al.* (Papageorgiou and Poggio, 1999) employed an over-complete set of 2D wavelets to detect vehicles in static images.

Figure 1 shows basic Haar filters: *two*, *three* and *four* rectangle features, where the sum of the pixels which lie within the white rectangles are subtracted from the sum of pixels in the grey rectangles. We conserve the *two* and *three* rectangle features since vehicles have rectangular shape: diagonal features (*four* rectangle template) doesn't give extra information for this type of pattern.

Viola & Jones (Viola and Jones, 2001) have introduced the Integral Image, an intermediate representation for the input image. The sum of the rectangular region in the image can be calculated in four Integral Image references. Then, the difference between two adjacent rectangles, can be computed with only six references, eight in the case of the three rectangle feature.

The Haar feature set is composed of the resulting value of the rectangular filters at various scales in a image.

In figure 2 we can see the results of two rectangular filters (vertical and horizontal) at two scales: 2x2 and 4x4 pixels. Lightness pixels mean important subtraction values (the result is always calculated in modulus). The complete set of Haar's features utilizing the three rectangular filters (see fig. 1) in a 32x32 pixel image at {2,4,8,16} scales is 11378. Every single feature j in the set could be defined as: $f_j = (x_j, y_j, s_j, r_j)$, where r_j is the rectangular filter type, s_j the scale and (x_j, y_j) are the position over the 32x32 image.

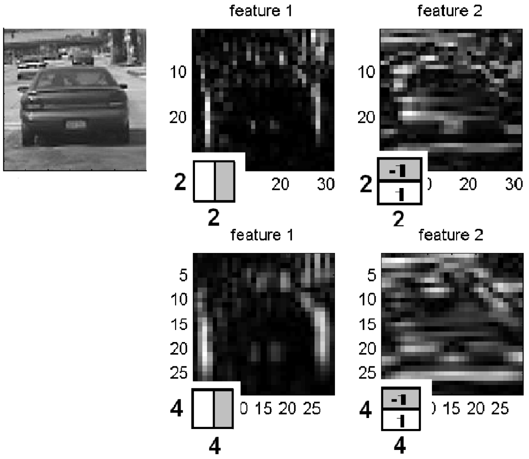


Figure 2: 2D Haar Wavelet example on a vehicle image.

2.2 Histogram of Oriented Gradient

The Histograms of Oriented Gradient (HoG) is another way to encode an input image to obtain a vector of visual descriptors. This local descriptor, based on Scale Invariant Feature Transform (SIFT) (Lowe, 1999), uses the gradient magnitude and orientation around a keypoint location to construct an histogram. Orientations are quantized by the number of bins in the histogram (four orientations are sufficient). For each histogram bin, we compute the sum in the region of all the magnitudes having that particular orientation. The histogram values are then normalised by the total energy of all orientations to obtain values between 0 and 1.

Gepperth (Gepperth et al., 2005) train a neural network classifier using these features for a two class problem: vehicle, non-vehicle. First, a ROI is subdivided into a fixed number of regions called *receptive fields*. From each *receptive field*, they obtain an oriented histogram feature.

The HoG features set is composed of histograms calculated inside a rectangular region on the original image. We evaluate the the gradient of the image using the Sobel filters to obtain the gradient magnitude and orientation.

There are three types of rectangle regions: r_1 square $l{*}l$, r_2 vertical rectangle $l{*}2l$, r_3 horizontal rectangle $2l{*}l$. Considering $l : \{2, 4, 8, 16\}$ scales, we have a total of 4678 features. A single histogram j in the set could be defined as: $h_j = (x_j, y_j, s_j, r_j)$, where r_j is the rectangular filter type, s_j the scale and (x_j, y_j) are the position over the 32x32 image.

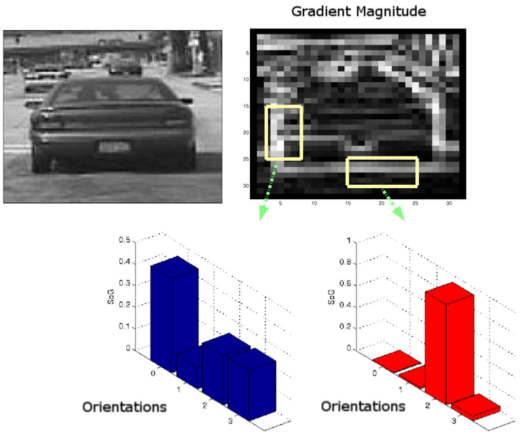


Figure 3: HoG example on a vehicle image.

3 ADABOOST

As we saw in previous sections, Haar and HoG representations are used to obtain a vector of visual descriptors describing an image. The size of these vectors is clearly bigger than the number of pixel in the image. Using the total number of features to carry out a classification is inadequate from the computing time point of view of the and the robustness, since many of these features do not contain important information (noise). Different methods: statistics (Schneiderman and Kanade, 2000), PCA, genetic algorithms (Sun et al., 2004), etc. can be used to select a limited quantity of representative features.

Among these methods, the Boosting (Freund and Schapire, 1996) classification method improves the performance of any algorithm. It finds precise hypothesis by combining several *weak classifiers* which, on average, have a moderate precision. The *weak classifiers* are then combined to create a *strong classifier*:

$$G = \begin{cases} 1 & \sum_{n=1}^{N} \alpha_n g_n \geq \frac{1}{2} \sum_{n=1}^{N} \alpha_n = T \\ 0 & \text{otherwise} \end{cases} \qquad (1)$$

Where G and g are the strong and weak classifiers respectively, and α is a coefficient wheighting each feature result. T is the strong classifier threshold.

Different variants of boosting are known such as Discrete Adaboost (Viola and Jones, 2001), Real AdaBoost (Friedman et al., 2000), Gentle AdaBoost, etc. The procedures (pseudo-code) of any of this variants are widely developed in the literature.

We need, however, to study the construction of the weak classifier for both cases: Haar and HoG features.

3.1 Haar Weak Classifier

We define the weak classifier as a binary function g:

$$g \begin{cases} 1 & \text{if } p_j f_j < p_j \theta_j \\ 0 & \text{otherwise} \end{cases} \qquad (2)$$

where f_j is the feature value, θ_j the feature threshold and p_j the threshold parity.

3.2 Hog Weak Classifier

This time, instead of evaluate a feature value, we estimate the distance between an histogram h_j of the input image and a model histogram m_j. The model is calculated like the mean histogram between all the training positive examples. For each histogram h_j of the feature set, we have the corresponding m_j. A vehicle model is then constructed and AdaBoost will found the most representative m_j which best separate the vehicle class from the non-vehicle class.

We define the weak classifier like a function g:

$$g \begin{cases} 1 & \text{if } d(h_j, m_j) > \theta_j \\ 0 & \text{otherwise} \end{cases} \qquad (3)$$

where $d(h_j(x), m_j)$ is the Bhattacharyya distance (Cha and Srihari, 2002) between the feature h_j and m_j and θ_j is the distance feature threshold.

4 TEST AND RESULTS

4.1 Dataset

The images used in our experiments were collected in France using a prototype vehicle. To ensure data variety, 557 images where captured during different time, and on different highways.

The training set contains 745 vehicle sub-images of typical cars, sport-utility vehicles (SUV) and minivan types. We duplicate this quantity flipping the sub-images around y-axis, obtaining 1490 examples. We split this new set keeping 1000 of the examples for training and the others for validation: the training set (TS) contains 1000 sub-images aligned to a resolution of 32 by 32 pixels, the validation set (VS) contains 490 vehicle sub-images with the same resolution. The negative examples come from 3196 images without vehicles.

The test set contains 200 vehicles in 81 images.

4.2 Single Stage Detector

First experiments were carried out with a strong classifier constructed with 100, 150 and 200 Haar or HoG features using the Discrete Adaboost algorithm (Viola and Jones, 2001).

We used the TS for the positive examples. The non-vehicle (negatives) examples were collected by selecting randomly 5000 sub-windows from a set of 250 non-vehicle images at different scales.

To evaluate the performance of the classifiers, the average detection rate (DR) and the number of false positives (FP) were recorded using a three-fold cross validation procedure. Specifically, we obtain three sets of non-vehicle sub-windows to train three strong classifiers. Then, we test these classifiers on the test set.

4.3 Multi Stage Detector

This section shows the test realised using a cascade of strong classifiers (Viola and Jones, 2001). The multi stage detector increases detection accuracy and reduces the computation time. Simpler classifiers (having a reduced number of features) reject the majority of the false positives before more complex classifiers (having more features) are used to reject difficult sub-windows.

Stages in the cascade are constructed with the Adaboost algorithm, training a strong classifier which achieves a minimum detection rate ($d_{min} = 0.995$) and a maximum false positive rate ($f_{max} = 0.40$). The training set is composed of the TS positive examples and the non-vehicle images separated in 12 different folders (the maximum number of stages). Subsequent classifiers are trained using those non-vehicle images of the corresponding folder which pass through all the previous stages.

An overall false positive rate is defined to stop the cascade training process ($F = 43 * 10^{-7}$) within the maximum number of stages.

This time, the average accuracy (AA) and false positives (FP) where calculated using a five-fold cross validation procedure. We obtain five detectors from five differents TS and VS randomly obtained.

4.4 Results

Table 1 shows the detection rate of the single stage detector trained either on Haar features or on HoG features with respectively 100, 150 and 200 features. These results are very interesting though quite predictible. As seen before, HoG classifiers computes a distance from the test sample to a ”vehicle model” (the mean histograms). These are generating classifiers. When the number of considered features increases, the model is refined and the detection rate increases while the number of false positives keeps stable. On the other hand, Haar classifiers are discriminative classifier evaluating a fronteer between positive and negative samples. Now, the fronteer is refined - and the number of false positives decreases - when the number of features increases. Figure 4 presents the ROC curves for each detector. As told before, for a given detection rate, the number of false positives is lower for Haar classifiers than for HoG classifiers.

Table 1: Single stage detection rates (Haar and HoG classifiers).

Classifier	DR (%)	FP	Time
HoG - 100 fts	69.0	1289	3,52
HoG - 150 fts	72.5	1218	4,20
HoG - 200 fts	83.1	1228	5,02
Haar - 100 fts	96.5	1443	2,61
Haar - 150 fts	95.7	1278	3,93
Haar - 200 fts	95.8	1062	5,25

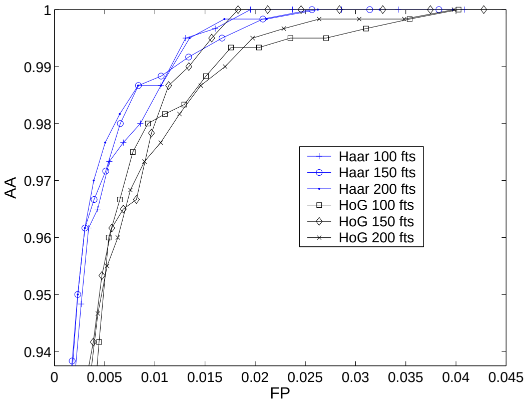


Figure 4: ROC curves for Haar and HoG Single Stage detectors.

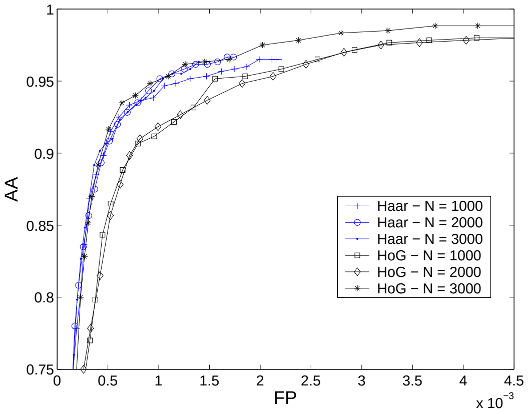
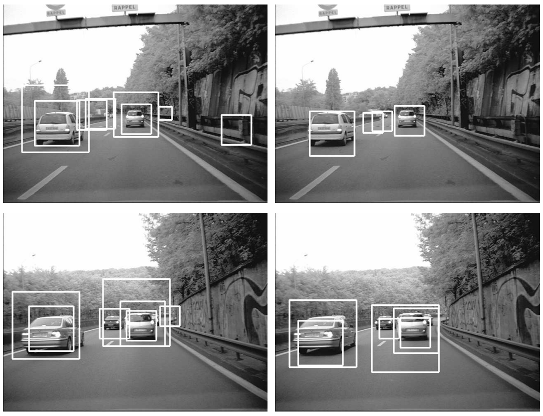


Figure 5: ROC curves for Haar and HoG Multi Stage detectors.

Table 2: Multi stage detection rate (Haar and HoG classifiers).

Classifier	Stages	# Fts	# Neg	DR (%)	FP	t (seg)
Haar	12	306	1000	94.3	598	0.75
Haar	12	332	2000	94	490	0.71
Haar	12	386	3000	93,5	445	0.59
HoG	12	147	1000	96.5	935	0.51
HoG	12	176	2000	96.1	963	0.59
HoG	11	192	3000	96.6	954	0.55



(a) (b)

Figure 6: Detection results for (a) HoG and (b) Haar Multi Stage detectors.

Table 2 shows results of cascade detectors using Haar and HoG based features. We also tested the effect of increasing the size of the negative set in each training stage. The behavior of each detector is the same as described before. HoG detector try to construct a finer vehicle model to take into account the new negatives. The number of features used increases as the model refines. But the detection rate and the number of false positives does not change significantly. Haar detector refines the fronteer using somemore features and the number of false positives decreases while the detection keeps quite stable. Figure 5 shows the ROC curves for each detector applied for the last stage in the cascade. For a given detection rate, these curves show a similar behavior as the single stage detector, where the number of false positives is lower for the Haar classifiers than for the HoG classifiers; except for the HoG detector trained with 3000 negatives, which has a similar behavior with a half quantity of features (see table 2). Figure 6 presents some detection results and false alarms.

5 CONCLUSION

This communication deals with a benchmark comparing Haar-like features and Histograms of Oriented Gradients features applied to vehicle detection. These features are used in a classification algorithm based on Adaboost. Two strategies are implemented: a single stage detector and a multi-stage detector. The tests - applied on realistic on-road images - show two different results: for the HoG (generative) features, when the number of considered features increases, the detection rate increases while the number of false positives keeps stable; for the Haar-like (discriminative) features, the number of false positives decreases. Future works will be oriented to combined these behaviors. An approach could be build using simultaneously both feature types. We should also select relevant features.

ACKNOWLEDGEMENTS

This research was supported by PSA Peugeot Citroën. The authors would liko to thank Fabien Hernandez from PCA *Direction de la Recherche et de l'Innovation Automobile* for their help with the data collection.

REFERENCES

Agarwal, S., Awan, A., and Roth, D. (2004). Learning to detect objects in images via a sparse, part-based representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11):1475–1490.

Bensrhair, A., Bertozzi, M., Broggi, A., Miche, P., Mousset, S., and Toulminet, G. (2001). A cooperative approach to vision-based vehicle detection. In *Proceedings on Intelligent Transportation Systems*, pages 207–212.

Bertozzi, M. and Broggi, A. (1997). Vision-based vehicle guidance. *Computer*, 30(7):49–55.

Bertozzi, M., Broggi, A., and Castelluccio, S. (1997). A real-time oriented system for vehicle detection. *Journal of Systems Architecture*, 43(1-5):317–325.

Bucher, T., Curio, C., Edelbrunner, J., Igel, C., Kastrup, D., Leefken, I., Lorenz, G., Steinhage, A., and von Seelen, W. (2003). Image processing and behavior planning for intelligent vehicles. *IEEE Transactions on Industrial Electronics*, 50(1):62–75.

Cha, S. and Srihari, S. N. (2002). On measuring the distance between histograms. *Pattern Recognition*, 35(6):1355–1370.

Collado, J., Hilario, C., de la Escalera, A., and Armingol, J. (2004). Model based vehicle detection for intelligent vehicles. In *International Symposium on Intelligent Vehicles*, pages 572–577.

Dellaert, F. (1997). Canss: A candidate selection and search algorithm to initialize car tracking. Technical report, Robotics Institute, Carnegie Mellon University.

Demonceaux, C., Potelle, A., and Kachi-Akkouche, D. (2004). Obstacle detection in a road scene based on motion analysis. *IEEE Transactions on Vehicular Technology*, 53(6):1649 – 1656.

Franke, U. (2000). Real-time stereo vision for urban traffic scene understanding. In *Proceedings IEEE Intelligent Vehicles Symposium 2000*, pages 273–278, Detroit, USA.

Freund, Y. and Schapire, R. (1996). Experiments with a new boosting algorithm. In *International Conference on Machine Learning*, pages 148–156.

Friedman, J., Hastie, T., and Tibshirani, R. (2000). Additive logistic regression: a statistical view of boosting. *The Annals of Statistics*, 28(2):337–374.

Gepperth, A., Edelbrunner, J., and Bocher, T. (2005). Real-time detection and classification of cars in video sequences. In *Intelligent Vehicles Symposium*, pages 625–631.

Guo, D., Fraichard, T., Xie, M., and Laugier, C. (2000). Color modeling by spherical influence field in sensing drivingenvironment. In IEEE, editor, *Intelligent Vehicles Symposium*, pages 249–254, Dearborn, MI, USA.

Lowe, D. (1999). Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision*, pages 1150–1157.

Papageorgiou, C. and Poggio, T. (1999). A trainable object detection system: Car detection in static images. Technical report, MIT AI Memo 1673 (CBCL Memo 180).

Schneiderman, H. and Kanade, T. (2000). A statistical method for 3d object detection applied to faces and cars. In *ICCVPR*, pages 746–751.

Srinivasa, N. (2002). Vision-based vehicle detection and tracking method for forward collision warning in automobiles. In *IEEE Intelligent Vehicle Symposium*, volume 2, pages 626–631.

Sun, Z., Bebis, G., and Miller, R. (2004). Object detection using feature subset selection. *Pattern Recognition*, 37(11):2165–2176.

Sun, Z., Bebis, G., and Miller, R. (2006). On-road vehicle detection: A review. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(5):694–711.

van Leeuwen, M. and Groen, F. (2001). Vehicle detection with a mobile camera. Technical report, Computer Science Institute, University of amsterdam, The Netherlands.

Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Conference on Computer Vision and Pattern Recognition*, pages 511–518.

Xiong, T. and Debrunner, C. (2004). Stochastic car tracking with line- and color-based features. *Advances and Trends in Research and Development of Intelligent Transportation Systems: An Introduction to the Special Issue*, 5(4):324–328.

Yang, H., Lou, J., Sun, H., Hu, W., and Tan, T. (2001). Efficient and robust vehicle localization. In *International Conference on Image Processing*, volume 2, pages 355–358, Thessaloniki, Greece.

MULTIMODAL COMMUNICATION ERROR DETECTION FOR DRIVER-CAR INTERACTION

Sy Bor Wang, David Demirdjian, Trevor Darrell
Massachusetts Institute of Technology, 32 Vassar Street, Cambridge MA 02139, USA
sybor@csail.mit.edu, demirdjian@csail.mit.edu

Hedvig Kjellström
KTH (Royal Institute of Technology), CSC, SE-100 44 Stockholm, Sweden
hedvig@csail.mit.edu

Keywords: Audio-Visual Recognition, System Error Identification, Conversational systems.

Abstract: Speech recognition systems are now used in a wide variety of domains. They have recently been introduced in cars for hand-free control of radio, cell-phone and navigation applications. However, due to the ambient noise in the car recognition errors are relatively frequent. This paper tackles the problem of detecting when such recognition errors occur from the driver's reaction. Automatic detection of communication errors in dialogue-based systems has been explored extensively in the speech community. The detection is most often based on prosody cues such as intensity and pitch. However, recent perceptual studies indicate that the detection can be improved significantly if both acoustic and visual modalities are taken into account. To this end, we present a framework for automatic audio-visual detection of communication errors.

1 INTRODUCTION

In recent years, there has been an increased interest in more intelligent and emotional car interfaces. This has been motivated by the need to make driver-car interactions easier and more natural and to reduce the cognitive load of the driver, who is now confronted with multiple device, e.g. phone, radio, navigation system. To this extent, speech recognition technology has recently been introduced in the car. However, due to the difficulty of the environment (e.g. noise produced by the engine, vibrations), speech recognition is still brittle and satisfying recognition rates can be obtained only for relatively small vocabularies, limiting the extent of the driver-car interaction. A speech recognition system that can automatically detect recognition errors would allow for smoother interaction.

Many spoken dialogue systems have difficulty determining whether the communication is going well or has problems (e.g. due to poor speech recognition). Various researchers have shown that human users change their speaking style when the system misrecognizes their speech as compared to when the system correctly recognized their speech (Hirschberg et al., 2001; Litman et al., 2001; Oviatt and VanGent, 1998). For example, users tend to speak slower or louder when speech misrecognition occurs. Such a change in speaking style usually leads to worse recognition results since standard speech recognizers are trained on normal, non-hyperarticulated speech (Oviatt and VanGent, 1998). These problems motivated the monitoring of prosodic aspects of a speaker's utterances, and several studies have shown that using automatically extracted prosodic features helps in error detection (Litman et al., 2001). However, the level of effectiveness of these prosodic features differs across studies and the analysis of prosodic features are done only on user utterances and not on audio cues of users while they are listening to the system response. Such limitations hint at the possible use of additional modalities or other types of features (e.g. visual features) to improve error detection.

The co-occurrence of audio and visual modalities has been widely explored for emotion recognition. Recent work in multimodal recognition of emotions has shown that a combination of prosodic features and facial expressions improves affect recognition (Zeng et al., 2004). The primary aim of our work in this paper is to find an automatic system to detect communication errors in a conversational system. Using both visual and audio features, we compare the performance of different classifiers in the unimodal stream and different audio-visual fusion strategies for identification, taking into account asynchrony between acoustic and visual user reactions, using audio and video data of user interactions with a dialogue system in a natural setting.

2 RELATED WORK

There has been limited literature on the use of low-level audio cues and visual features in automatically detecting dialogue-based system errors in an authentic environment. A perceptual study conducted by Barkhuysen et al. (Barkhuysen et al., 2004) showed that audio and visual cues are useful in detecting communication errors. The study also showed that using visual cues were very effective for detecting system errors when the user is listening in silence to the reply from the dialog manager. In this study though, subjects were specifically instructed to face a camera embedded in a cellphone while speaking to it. Knowledge of this camera could bias the subject's behavior. As shown by Sebe et al. (Sebe et al., 2004), this knowledge bias was significant for learning facial expressions. In this work, subjects were viewing movie clips in a kiosk, without any knowledge of a camera capturing their facial expressions. However, no prosody or audio cues of the subjects were collected.

Recent work done in emotion or affect recognition has explored the combined use of prosody and facial features (Zeng et al., 2004; Chen et al., 1998). Zeng et al. (Zeng et al., 2004) used a voting method to combine the facial feature and prosody classifiers to improve affect recognition. Although this work addressed the difficult task of classifying eleven emotional states, it suffers from the use of a database where subjects generated emotions upon request, which may not be the genuine expressions in an authentic environment.

In the domain of detecting communication errors, also known as system errors, audio cues have been explored widely. Oviatt (Oviatt and VanGent, 1998) showed that there is a pattern of hyper-articulation when there are system errors, which leads to worse recognition results. Litman et al. (Litman et al., 2001) and Hirschberg et al. (Hirschberg et al., 2001) automatically extracted prosodic features of a speaker's utterances and showed that these features have been useful in error detection, although the extent to which prosody is beneficial differs across studies. This implies that the accuracy of error detection can be improved by the addition of other features, e.g. visual cues, either as a combination with audio cues or simply as a separate stream.

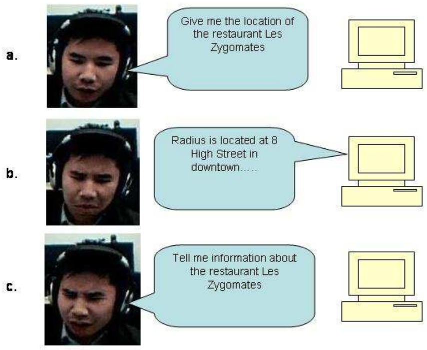

a.
Give me the location of the restaurant Les Zygomates
b.
Radius is located at 8 High Street in downtown.....
c.
Tell me information about the restaurant Les Zygomates

Figure 1: Illustration of communication errors. In a., the subject is making a query of a restaurant the first time. In b., the subject is listening to the response of the system. And in c., the subject repeats his query. The facial expression of the user in b. and c., as well as the tone of the user's voice in c. are the cues our algorithm exploits for the detection of communication errors.

In this paper we propose to detect communication errors multimodally by using visual and audio features. We focus on an environment where the user is interacting with a conversational information query system similar to the ones present in car navigation systems. We define a communication error as the instance where the system misrecognizes the driver speech, and makes an erroneous reply. This is illustrated in Figure 1. We detect the presence of this error when the user is speaking or when the user is listening to the system.

The rest of the paper is described as follows. First, we describe the features we extract from audio and visual streams to measure confusion. Then we give a description of our classification models and late fusion strategies, followed by our experiment setup and procedure. In the last section, we show comparative results of the different classifiers.

3 MULTIMODAL INPUT

3.1 Visual Component

In this section we describe our algorithm for estimating head pose and facial motion features from monocular image sequences. In our framework, head motion is decomposed into two distinct components. The first component consists of the 3D rigid motion of the head. The second component consists of the local motion generated by the non-rigid parts of the face (e.g. mouth, lips, eyes).

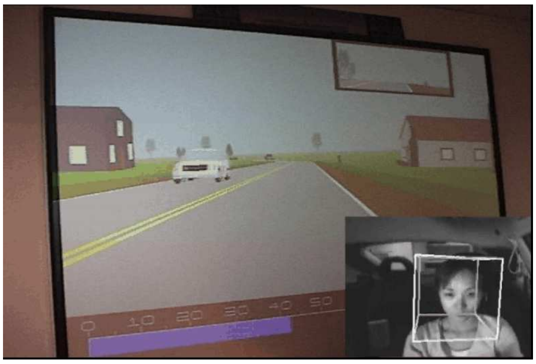


Figure 2: Audio-visual platform installed in a car simulator. The face of the driver is tracked using a camera installed near the rear view mirror: the white cube surrounding the driver's face corresponds to Π, the location and orientation of the pose estimate.

The head motion estimation algorithm consists of first estimating the rigid motion component of the head by using a robust algorithm. This rigid motion then is used to compensate the global motion of the head and to estimate the local motion of the non-rigid parts.

3.1.1 Head Motion and Pose Estimation

The algorithm for head motion and pose estimation is an implementation of the drift-free tracking technique described in (Morency et al., 2003). In contrast to the original work, which required stereo inputs, our implementation estimates head pose in monocular image sequences. In addition motion estimation is based on the robust estimator RANSAC (Fischler and Bolles, 1981) which allows large noise in the data (e.g. temporary occlusions of the face). The face tracking algorithm is initialized in a standard fashion using a frontal-view face detector (Viola and Jones, 2004).

Our algorithm provides the head pose in a 3D coordinate system and is represented by a 6-vector Π:

$$\Pi = (t_x, t_y, t_z, \phi_\alpha, \phi_\beta, \phi_\gamma) \qquad (1)$$

where (t_x, t_y, t_z) is the location and $(\phi_\alpha, \phi_\beta, \phi_\gamma)$ the orientation (pan, tilt and swing angles) of the head.

3.1.2 Facial Motion Features

Let δ be the rigid motion between the last two frames $t-1$ and t. The facial motion features are defined as the head motion-compensated optical flow, i.e. the optical flow between the images I_{t-1} and I_t from which the motion δ has been 'subtracted'. The facial motion features correspond to the local non-rigid motion generated by the muscles of the face only (e.g. lips, jaw, eyes), independent from the global head motion. In our framework, the vision-based features are defined as:

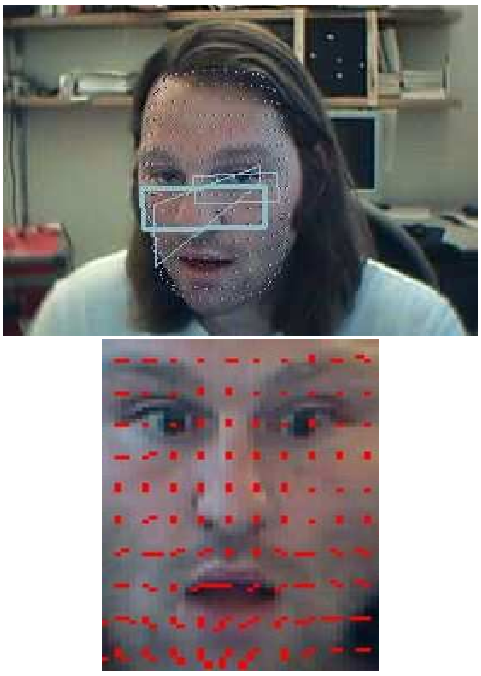


Figure 3: Our head motion estimation algorithm consists in estimating the rigid motion component (left image) and compensated optical flow on the face surface (right image).

$$V_f = (\bar{u}(m_1), \ldots, \bar{u}(m_N)) \qquad (2)$$

where $\bar{u}(m_k)$ is the head motion-compensated optical flow for a point m_k of the face.

3.2 Audio Component

We use three kinds of prosody features: the intensity E, the pitch, F_0, and the first formant frequency, F_1. The prosody feature vector A_f is then defined as:

$$A_f = (E, F_0, F_1) \qquad (3)$$

These features are computed at every 10 msecs using the speech analysis software, PRAAT (Boersma, 2001). The intensity E is computed as: $E = log(\sum_{i=1}^{N}(x[i]-\bar{x})^2)$ where N is the window length and $x[i]$ is the ith sample in that window and $\bar{x}$ is the local average signal value. In our computation (and for the rest of this section) we used a window length of 40 samples. The pitch F_0 is estimated as the reciprocal of the fundamental period as described in (Boersma, 1993). In our experiments, we set the search range of the pitch to be 75 - 1000 Hz. As for the computation of the first formant frequency, F_1, a segment of N samples is extracted for every time step of 1 msec. This segment is multiplied by a Gaussian-like window and the LPC coefficients are computed. This first formant is then extracted using these coefficients by the Burg algorithm described in (Childers, 1978).

In previous work (Sebe et al., 2004) syllable rate was used as a prosody feature. However, in our work, our audio data consists of spoken as well as non-spoken words, e.g. exclamations, gasps or humming, which we want to model for automatic problem detection. and our speech recognizer had a lot of difficulty computing an accurate syllable rate. Of the 219 utterances processed by the speech recognizer, 97 utterances have an incorrect number of hypothesized vowel phones. On average, these incorrectly recognized utterances have 2.73 syllables more than the hypothesized ones.

4 MULTIMODAL DETECTION OF SYSTEM ERRORS

We explore different techniques to detect communication errors from sequences of audio-visual features estimated in Section 3.2. First, we describe unimodal classification models followed by the multimodal fusion strategies we tested.

4.1 Unimodal Classification Methods

We want to map an observation sequence $\mathbf{x}$ to class labels $y \in \mathcal{Y}$, where $\mathbf{x}$ is a vector of t consecutive observations, $\mathbf{x} = \{x_1, x_2, \ldots x_t\}$. In our case, the local observation x_t can be an audio feature A_f, or a visual feature, V_f.

To detect communication errors, learning the sequential dynamics of these observations is important. Hidden Markov Models (HMMs) (Rabiner, 1989) are well known generative probabilistic sequence models that capture sequence dynamics; Hidden Conditional Random Fields (HCRFs) (Quattoni et al., 2004; Wang et al., 2006) are discriminative analogs that have been recently introduced for gesture recognition. We compare both techniques in our experiments below; experiments with classifiers taking a single observation as input previously demonstrated poor results, and were not included in our experiments.

Hidden Markov Models (HMM) - We trained a HMM model for each communication state. During evaluation, test sequences were passed through each of these models and the model with the highest likelihood was selected as the recognized communication state. This is a generative, sequential model with hidden states. More details of this model are described in (Rabiner, 1989).

Hidden Conditional Random Fields (HCRF) - The HCRF is a model that has recently been introduced for the recognition of observation sequences (Quattoni et al., 2004). Here we describe the HCRF model briefly:

A HCRF models the conditional probability of a class label given an observation sequence by:

$$P(y \mid \mathbf{x}, \theta) = \sum_{\mathbf{s}} P(y, \mathbf{s} \mid \mathbf{x}, \theta) = \frac{\sum_{\mathbf{s}} e^{\Psi(y,\mathbf{s},\mathbf{x};\theta)}}{\sum_{y' \in \mathcal{Y}, \mathbf{s} \in S^m} e^{\Psi(y',\mathbf{s},\mathbf{x};\theta)}} \tag{4}$$

where $\mathbf{s} = \{s_1, s_2, \ldots, s_m\}$, each $s_i \in S$ captures certain underlying structure of each class and S is the set of hidden states in the model. If we assume that $\mathbf{s}$ is observed and that there is a single class label y then the conditional probability of $\mathbf{s}$ given $\mathbf{x}$ becomes a regular CRF. The potential function $\Psi(y, \mathbf{s}, \mathbf{x}; \theta) \in \Re$, parameterized by θ, measures the compatibility between a label, the observation sequence, and the configuration of the hidden states.

In our paper, the local observations are the visual features, V_f, or the audio features, A_f. We trained a single two-class HCRF. Test sequences were run with this model and the communication state class with the highest probability was selected as the recognized error state.

For the HMM model, the number of Gaussian mixtures and states was set by minimizing the error on training features. For the HCRF model, the number of hidden states was set in a similar fashion.

4.2 Multimodal Fusion Strategies

We have a choice between early or late fusion when combining the audio and visual modalities. In early fusion, we can model the audio and visual features in a single joint feature space, and use the joint feature for training a single classifier. In late fusion, we can train a classifier on each modality separately and merge the outputs of the classifiers. As illustrated in Figure 1, our communication error detection has two different modes: in b. we use visual features only for error detection and in c. we use both audio and visual features. The single mode in b. requires us to train a classifier using a single input stream. In addition, training classifiers based on individual streams is a simpler process. As such, we choose late fusion techniques, i.e. fusing the outputs of two classifiers. We use two common late-fusion strategies as described in (Kittler et al., 1998).

Let the feature input to the j-th classifier, $j = 1, \ldots, R$ be x_j, and the winning label be h. A uniform prior across all classes is assumed.

PRODUCT rule: $h = \arg\max_k \prod_{j=1}^{R} P(w_k|x_j)$.

With the product rule, we multiply the probabilities of the visual feature classifier and the audio feature classifier, and select the winning class based on the highest scoring multiplication.

SUM rule: $h = \arg\max_k \sum_{j=1}^{R} P(w_k|x_j)$.

With the sum rule, we add the probabilities of the visual feature classifier and the audio feature classifier, and select the winning class based on the highest scoring sum.

5 EXPERIMENTS AND RESULTS

5.1 Data Collection

To evaluate the performance of the different classifiers and fusion strategies, we collected an audio-visual database where the facial expressions and the audio cues would correspond to the actual conversational state of the subject. There were several design issues we had to consider to minimize bias of our data collection experiment. These issues were approached in a similar fashion to the database collected for natural facial expressions (Sebe et al., 2004). In particular, the subjects could not know that they were being tested for their communication state [1]. Such knowledge could influence their communication state and invalidate the results.

We set up a conversational kiosk with a hidden camera and microphone array. This kiosk contained a web-based navigation query speech interface consisting of a display showing a Google map with restaurant icons. Subjects did not know that they were involved in an experiment about communication error detection: they were told their task was to test the navigation system and report what they liked or disliked about it at the end of the experiment. They were given a list of restaurants to query for information. The subjects had to make the queries in sequential order, and repeat the query in any way they wished (e.g. repeating the same question, or using a different phrase) when the system did not respond correctly. They could only proceed to the next query when the system displayed the correct restaurant information on the display. The purpose of this sequential query was to create a need to solve the communication error when it occurs. The audio and video of the user were recorded throughout the whole experiment under brightly lit and low noise conditions. The video was recorded at 15 Hz and the audio at 44kHz. A total of six subjects performed the study. All six subjects were male and were between 20-30 years of age.

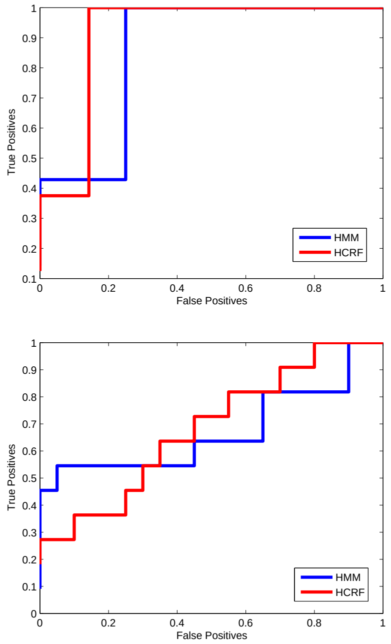


Figure 4: (top) ROC curves showing the performance of the different classifiers of visual features. (bottom) ROC curves showing the performance of the different classifiers using prosody features. From this figure, HMMS and HCRFs do not perform well on the prosody features, while HCRFs perform much better than HMMs on visual features.

From the database described above, all the sequences were manually labeled and segmented for training our classifiers described in Section 4.1. A total of 227 error sequences and 84 error-free ones were collected. For each human subject, 90% of his/her data were picked at random and used for training, while the remaining ones were used for testing.

5.2 Testing and Evaluation

5.2.1 Visual Features Classification

Facial motion features, V_f, described in Section 3.1 are used as observations for training and testing. Figure 4 shows the results of the classifiers described in Section 4.1 using visual features. From this figure, HCRF performs better than HMMs for visual feature classification.

[1] At the end of the experiment, we procured agreement for the use of the audio-visual footage from the subjects for our experiments.

5.2.2 Prosody Features Classification

Using prosody features, A_f, from Section 3.2 as observations, we trained two classifiers described in Section 4.1. Figure 4 shows the ROC curves for the different classifiers. From this figure, both HCRFs and HMMs perform poorly for prosody feature classification. This is due to the use of only three acoustic characteristics as our prosody features and shows that such features are not very indicative of communication problems.

5.2.3 Audio-Visual Classification

We compared the performance of HMMs and HCRFs in the late fusion experiments. Figure 5 shows the ROC curve of the combining the various classifiers using the SUM and PRODUCT rule. The classifiers show a significant improvement using the late fusion strategies, despite a poor performance when only prosody features are used. Interestingly, this concurs with findings in the area of audio-visual speech perception (Massaro, 1987; Summerfield, 1987), indicating that humans fuse information from different modalities at a relatively late stage. In addition, fusion of HCRF classifiers performed better than fusion of HMM classifiers. Using the SUM rule to combine the HCRF classifiers produced the best result.

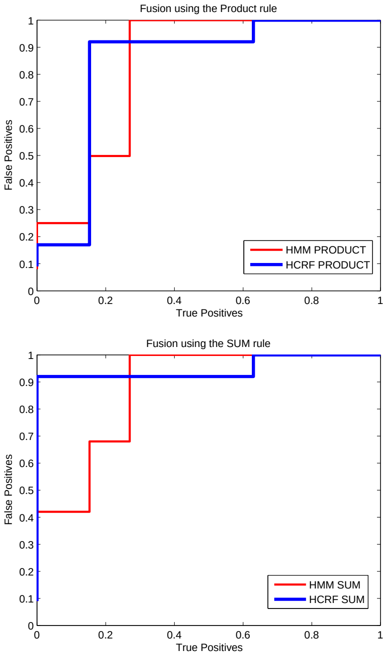


Figure 5: (top) ROC curve showing the performance of different classifiers using PRODUCT rule for fusion.(bottom) ROC curve showing the performance of different classifiers using SUM rule for fusion. Both curves show that despite poor performance from the audio stream, late fusion improved the performance significantly. In addition, HCRFs perform better than HMMs after fusion.

6 CONCLUSIONS

In this paper, we presented experiments evaluating different classification and fusion methods for detecting communication errors in a conversational system. Authentic audio-visual data of human-dialogue interactions with the conversational system was collected and labeled according to the presence of communication errors, and used to train and test the automatic system.

Features extracted from the audio included different prosody characteristics, such as pitch and intensity and the first formant frequency. From the visual channel, the user's global head motion and local face motion were extracted. Different strategies for classification from these cues were evaluated, as well as strategies for fusion of the two modalities. Despite the problems associated with asynchrony of audio and visual features, we used simple late fusion strategies of our HCRF and HMM classifiers and showed that they have improved error detection. To summarize, we find that communication errors in a dialogue-based system can be detected with a better accuracy using a HCRF with audio-visual input and a fusion strategy using the SUM rule.

REFERENCES

Barkhuysen, P., Krahmer, E., and Swerts, M. (2004). Audiovisual perception of communication problems. In *Speech Prosody*.

Boersma, P. (1993). Accurate short-term analysis of the fundamental frequency and the harmonis-to-noise ratio of a sampled sound. In *IFA*.

Boersma, P. (2001). Praat, a system for doing phonetics by computer. *Glot International*, 5(9/10):341–345.

Chen, L., Huang, T. S., Miyasato, T., and Nakatsu, R. (1998). Multimodal human emotion/expression recognition. In *International Conference on Face and Gesture Recognition*.

Childers, D. G. (1978). *Modern Spectrum Analysis*. IEEE Press.

Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395.

Hirschberg, J., Litman, D., and Swerts, M. (2001). Identifying user corrections automatically in spoken dialogue systems. In *2nd Meeting of the North American Chapter of the Association for Computational Linguistics on Language Technologies*.

Kittler, J., Hatef, M., Duin, R., and Matas, J. (1998). On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239.

Litman, D., Hirschberg, J., and Swerts, M. (2001). Predicting user reactions to system error. In *ACL*.

Massaro, D. (1987). *Speech Perception By Ear and Eye*. Lawrence Erlbaum Associates, Hillsdale, NJ, USA.

Morency, L., Rahimi, A., and Darrell, T. (2003). Adaptive view-based appearance models. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 803–810.

Oviatt, S. L. and VanGent, R. (1998). Error resolution during multimodal human-computer interaction. In *SpeechCommunication*.

Quattoni, A., Collins, M., and Darrell, T. (2004). Conditional random fields for object recognition. In *Neural Information Processing Systems*.

Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

Sebe, N., Lew, M., Cohen, I., Sun, Y., Gevers, T., and Huang, T. S. (2004). Authentic facial expression analysis. In *International Conference on Automatic Face and Gesture Recognition*.

Summerfield, Q. (1987). Some preliminaries to a comprehensive account of audio-visual speech perception. In Dodd, B. and Campbell, R., editors, *Hearing by Eye*, pages 3–51. Lawrence Erlbaum Associates, Hillsdale, NJ, USA.

Viola, P. and Jones, M. J. (2004). Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154.

Wang, S., Quattoni., A., Morency, L.-P., Demirdjian, D., and Darrell, T. (2006). Hidden conditional random fields for gesture recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Zeng, Z., Tu, J., Liu, M., Zhang, T., Rizzolo, N., Zhang, Z., Huang, T. S., Roth, D., and Levinson, S. (2004). Bimodal hci-related affect recognition. In *International Conference on Multimodal Interfaces*.

PRIUS CONTROL WITH A HYBRID METHOD

Danil Prokhorov
Toyota Technical Center
Toyota Motor Engineering and Manufacturing North America (TEMA)
2350 Green Rd., TRD, Ann Arbor, MI 48105, U.S.A.
dvprokhorov@gmail.com

Keywords: HEV powertrain, neurocontroller, extended Kalman filter.

Abstract: We describe an application of a computational intelligence method for superior control of the Toyota Prius hybrid electric vehicle. We are interested in improvement of fuel efficiency while reducing emissions. We describe our approach which is based on recurrent neural networks. The proposed approach is quite general and applicable to other complex real-world systems.

1 INTRODUCTION

Hybrid powertrains have been gaining popularity due to their potential to improve significantly fuel economy and reduce undesirable emissions. Control strategies of the hybrid electric vehicle (HEV) are more complex than those of the internal combustion engine-only vehicle because they have to deal with multiple power sources in sophisticated configurations. The main function of any control strategy is power management. It typically implements a high-level control algorithm which determines the appropriate power split between the electric motor and the engine to minimize fuel consumption and emissions, while staying within specified constraints on drivability, reliability, battery charge sustenance, etc.

Computational intelligence techniques have previously been applied to HEV power management by various authors. A rule-based control was employed in (Baumann et al., 2000). Fuel economy improvement with a fuzzy controller was demonstrated in (Schouten et al., 2002), relative to other strategies which maximized only the engine efficiency. An intelligent controller combining neural networks and fuzzy logic which could adapt to different drivers and drive cycles (profiles of the required vehicle speed over time) was studied in (Baumann et al., 1998). Recently a neurocontroller was employed in a hybrid electric propulsion system of a small unmanned aerial vehicle which resulted in significant energy savings (Harmon et al., 2005).

The references cited above indicate a significant potential for improving HEV performance through more efficient power management based on application of computational intelligence (CI) techniques. To the best of our knowledge, there has been no work on improving HEV control by CI methods for the Toyota Prius. Though the Prius efficiency is quite high already, there is a significant potential for further improvement, as will hopefully become apparent from this paper.

Unlike traditional hybrid powertrain schemes, series or parallel, the Prius hybrid implements what is called the power split scheme. This scheme is quite innovative and has not been studied extensively yet. The Prius powertrain uses a planetary gear mechanism to connect an internal combustion engine, an electric motor and a generator. A highly efficient engine can simultaneously charge the battery through the generator and propel the vehicle (Figure 1). It is important to be able to set the engine operating point to the highest efficiency possible and at sufficiently low emission levels of undesirable exhaust gases such as hydrocarbons, nitrogen oxides and carbon monoxide. The motor is physically attached to the ring gear. It can move the vehicle through the fixed gear ratio and either assist the engine or propel the vehicle on its own for low speeds. The motor can also return some energy to the battery by working as another generator in the regenerative braking mode.

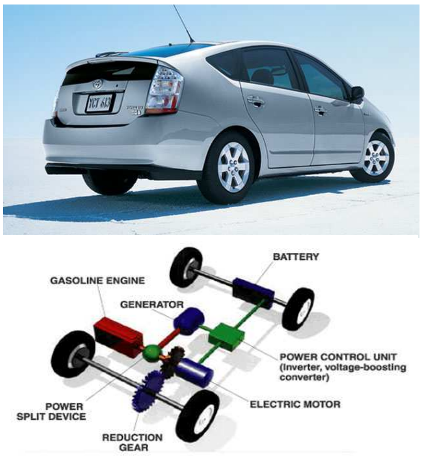


Figure 1: The Prius car and the main components of the Toyota hybrid system.

As in our previous work (Prokhorov et al., 2001), we employ recurrent neural networks (RNN) as controllers and train them for robustness to parametric and signal uncertainties (known bounded variations of physical parameters, reference trajectories, measurement noise, etc.). We intend to deploy the trained neurocontroller with fixed weights.

This paper is structured as follows. In the next section we describe the main elements of our off-line training. The approach permits us to create an RNN controller which is ready for deployment with fixed weights. We then describe our experiments in Section 3, followed by our conclusions and directions for future work.

2 OFF-LINE TRAINING

We adopt the approach of indirect or model based control development for off-line training. The Prius simulator is a highly complex, distributed software which makes training a neurocontroller directly in the simulator difficult. We implemented a hybrid approach in which the most essential elements of the simulator are approximated sufficiently accurately by a neural network model. The NN model is used to train a neurocontroller by effectively replacing the simulator. The trained neurocontroller performance is then verified in the simulator.

The use of differentiable NN for both model and controller makes possible application of the industrially proven training method which employs the backpropagation through time (BPTT) and the extended Kalman filter (EKF) for training NN. We refer the reader to other references for its more comprehensive account (Prokhorov et al., 2001), (Puskorius et al., 1996).

3 EXPERIMENTS

We first train a NN model to enable off-line training the neurocontroller as discussed in Section 2. To do supervised training of the NN model in Figure 2, we gather the input-output pairs from 20 diverse drive cycles generated in the Prius simulator. We trained a 25-node structured RNN for 3000 epochs using the multi-stream EKF (Prokhorov et al., 2001) and attained the training RMSE of $5 \cdot 10^{-4}$ (the largest generalization RMSE was within 20% of the training RMSE).

The closed-loop control system for training the NN controller is shown in Figure 2. The converter determines the required values of the speed ω_r^d and the torque T_r^d at the ring gear of the planetary mechanism to achieve the desired vehicle speed specified in the drive cycle. This is done on the basis of the Prius model of motion. The constraint verifier assures satisfaction of various constraints which must hold for the engine, the motor and the generator speeds and torques in the planetary gear mechanism, i.e., ω_e and T_e, ω_m and T_m, ω_g and T_g, respectively.

Our first control goal is to minimize the average fuel consumed by the engine. However, fuel minimization only is not feasible. The Prius nickel-metal hydride battery is the most delicate nonlinear component of the system with long-term dynamics due to discharging, charging and temperature variations. It is important to avoid rapid and deep discharges of the battery which can drastically reduce its life, requiring costly repairs or even battery replacement. Thus, the second goal of the HEV control is to maintain the battery State Of Charge (SOC) throughout the drive cycle in the safe zone. The SOC can vary between 0.0 (fully discharged) and 1.0 (fully charged), but the safe zone is typically above 0.4.

We combine the two control goals to obtain $cost(t) = \lambda_1 sf^2(t) + \lambda_2(t)(SOC^d(t) - SOC(t))^2$, where $sf(t)$ stands for specific fuel or fuel rate consumed by the engine at time t, $\lambda_1 = 1$, and $\lambda_2(t) \in [10, 50]$ due to about one order of magnitude difference between values of sf and those of SOC. The desired $SOC^d(t)$ is constant in our experiments for simplicity. We encourage our controller to pay approximately the same level of attention to both sf and SOC, although the optimal balance between λ_1 and λ_2 is yet to be determined. We also penalize reductions of the SOC below SOC^d five times heavier than its increases to discourage the controller from staying in the low-SOC region for long. Thus, $\lambda_2(t) = 10$ if $SOC(t) \geq SOC^d$, and $\lambda_2(t) = 50$ if $SOC(t) < SOC^d$.

Ultimately, we would also like to minimize emissions of the harmful gases. In this study we attempt to reduce emissions indirectly through reducing the fuel consumption because they are often correlated.

Our RNN controller has 5-5R-2 architecture, i.e., five inputs, five recurrent nodes in the fully recurrent hidden layer, and two bipolar sigmoids as output nodes. The RNN receives as inputs the required output drive speed ω_r^d and torque T_r^d, the current engine fuel rate sf, the current SOC and the desired SOC SOC^d (see Figure 2; the desired fuel rate is implicit, and it is set to zero). The RNN produces two control signals in the range of ± 1. The first output is the engine torque τ_e, and the second output is the engine speed w_e which become T_e and ω_e, respectively, after passing through the constraint verifier.

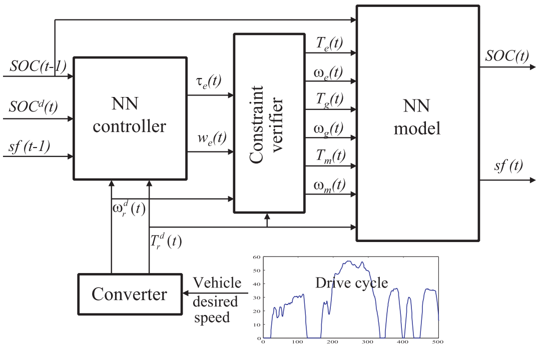


Figure 2: Block diagram of the closed-loop system for training the NN controller. The converter determines the required values of speed ω_r^d and torque T_r^d at the ring gear of the planetary mechanism to achieve the desired vehicle speed profile. The constraint verifier makes sure not only that the torques and speeds are within their specified physical limits but also that they are consistent with constraints of the planetary gear mechanism. The trained NN model takes care of the remaining complicated dynamics of the plant. The feedback loop is closed via SOC and the fuel rate sf, but the required ω_r^d and T_r^d are guaranteed to be achieved through the appropriate design of the constraint verifier.

Our RNN controller is trained off-line using the multi-stream EKF algorithm described in Section 2. When training of our NN controller from Figure 2 is finished, we can deploy it inside the high-fidelity simulator which approximates well behavior of the real Prius and all its powertrain components. As expected, we observed some differences between the neurocontroller performance in the closed loop with the NN model and its performance in the high-fidelity simulator because the NN model and the verifier only approximate the simulator's behavior. Our results below pertain to the simulator, rather than its NN approximation.

The basic idea of the current Prius HEV control logic is discussed in (Hermance, 1999). When the power demand is low and the battery SOC is sufficiently high, the motor powers the vehicle. As the power demand and vehicle speed increase, or the SOC reduces below a threshold, the engine is started (the generator may help the motor start the engine). The engine power is split between propelling the vehicle and charging the battery through the generator. As the power demand continues to grow, the engine might not be able to stay within its efficiency limits. In those cases the motor can provide power assist by driving the wheels to keep the engine efficiency reasonably high, as long as the battery can supply the required power. During decelerations the motor is commanded to operate as a generator to recharge the battery, thereby implementing regenerative braking.

It is hard to make this rule-based strategy optimal for such a complex powertrain. Significant averaging over drive cycles with quite different behavior compromising the best achievable performance is unavoidable. We believe that a strategy based on a data-driven learning system should be able to beat the rule-based strategy because of its ability to discern differences in driving patterns and take advantage of them for improved performance.

We compare our RNN controller trained for robustness with the rule-based control strategy of the Prius on 20 drive cycles including both standard cycles (required by government agencies) and non-standard cycles (e.g., random driving patterns). Our RNN controller is better by 15% on average than the rule-based controller in terms of fuel efficiency, and it appears to be slightly better than the rule-based controller in terms of its emissions on long drive cycles. It also reduces variance of the SOC over the drive cycle by at least 20%.

Figure 3 shows an example of our results. It is a fragment of a long drive cycle (the total length is $12,700$ seconds). Our advantage appears to be in the more efficient usage of the engine. The engine efficiency is 32% vs. 29% for the rule-based controller. We also achieved a big improvement in the generator efficiency: 77% vs. 32%, with other component efficiencies remaining basically unchanged.

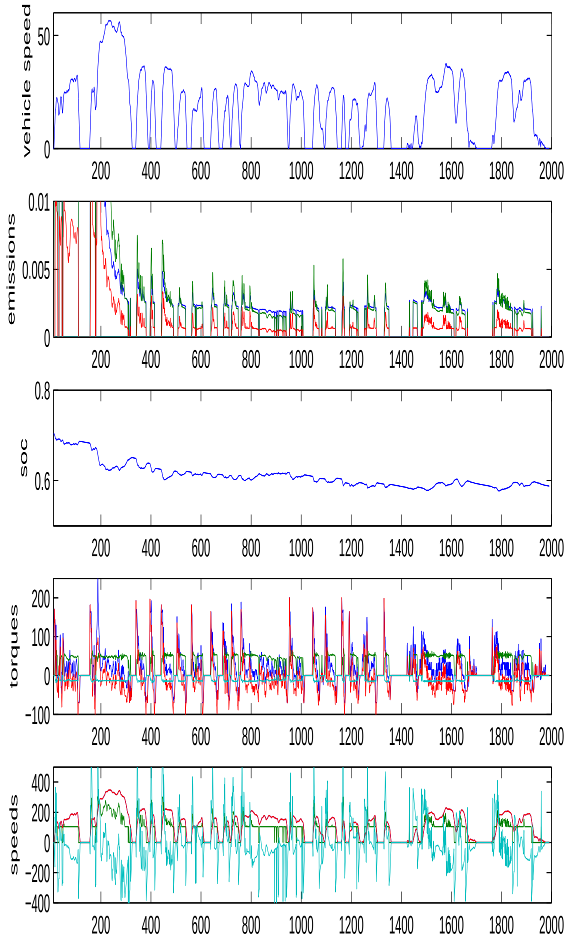


Figure 3: A 2000-second fragment of a long city drive cycle illustrating the typical performance of our RNN controller. The initial segment (from 0 to $\sim$ 300 seconds) has significant unavoidable emissions due to the engine cold start. The speed ω_r^d and the torque T_r^d are the blue lines, ω_e and T_e are the green lines, the ω_m and T_m are the red lines, and ω_g and T_g are the cyan lines. Note that $\omega_m = \omega_r^d$ due to the system design constraint.

4 CONCLUSION

We illustrate our approach to training neurcontrollers on the Toyota Prius HEV through a high-fidelity simulation, which is done for the first time by methods of computational intelligence and with results improved over those of the existing controller. Our approach is applicable to many real-world control problems.

REFERENCES

Baumann, B., Rizzoni, G., and Washington, G. N. (1998). Intelligent control of hybrid vehicles using neural networks and fuzzy logic. In *SAE Technical Paper 981061, SAE Int. Cong. and Exposition*. SAE Press.

Baumann, B. M., Washington, G. N., Glenn, B. C., and Rizzoni, G. (2000). Mechatronic design and control of hybrid electric vehicles. *IEEE/ASME Transactions on Mechatronics*, 5(1):58–72.

Harmon, F., Frank, A., and Joshi, S. (2005). The control of a parallel hybrid-electric propulsion system for a small unmanned arial vehicle using a cmac neural network. *Neural Networks*, 18(5/6):772–780.

Hermance, D. (1999). Toyota hybrid system. In *SAE TOPTEC Conference Proc., Albany, NY*. SAE Press.

Prokhorov, D. V., Puskorius, G. V., and Feldkamp, L. A. (2001). Dynamical neural networks for control. In *A Field Guide to Dynamical Recurrent Networks, Edited by J. Kolen and S. Kremer*. IEEE Press.

Puskorius, G. V., Feldkamp, L. A., and L. I. Davis, J. (1996). Dynamic neural network methods applied to on-vehicle idle speed control. *Proceedings of the IEEE*, 84(10):1407–1420.

Schouten, N., Salman, M., and Kheir, N. (2002). Fuzzy logic control for parallel hybrid vehicles. *IEEE Transactions on Control Systems Technology*, 10(3):460–468.

AUTHOR INDEX

AUTHOR INDEX (CONT.)

Proceedings of ICINCO 2007
Fourth International Conference on
Informatics in Control, Automation and Robotics
ISBN: 978-972-8865-83-2
http://www.icinco.org