ICINCO 2007

FOURTH INTERNATIONAL CONFERENCE ON
INFORMATICS IN CONTROL, AUTOMATION AND ROBOTICS

Proceedings

Robotics and Automation - Vol. 2

ANGERS, FRANCE · MAY 9-12, 2007

ORGANIZED BY

INSTICC

IN COOPERATION WITH

aaai

CO-SPONSORED BY

IFAC

GdR MACS

CNRS

EEA

INSTICC PRESS

ICINCO 2007

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

Volume RA-2

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

SHORT PAPERS

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

SHORT PAPERS

VEHICULAR ELECTRONIC DEVICES CONNECTED BY ONBOARD FIELDBUS TECHNOLOGIES

Miguel A. Domínguez, Perfecto Mariño, Francisco Poza and Santiago Otero
Electronic Technology Department
University of Vigo
E.T.S.I. Industriales, Campus Lagoas-Marcosende, 36280, Spain
mdgomez@uvigo.es, pmarino@uvigo.es, fpoza@uvigo.es, jsotero@uvigo.es

Keywords: Vehicular Control, Fieldbuses, CAN, City Public Transport Buses, Multiplexed Solutions.

Abstract: The electrical circuits and their Electronic Control Units (ECUs) in buses and coaches are essential for their good working. Drive, braking, suspension, opening door, security and communication devices must be integrated in a reliable and real time information system. The industrial communication networks or fieldbuses are a good solution to implement networked control systems for the onboard electronics in the public transport buses and coaches. The authors are working in the design of multiplexed solutions based on fieldbuses to integrate the body and chassis functions of city public transport buses. An example for the EURO5 model of the Scania manufacturer is reported in this paper. The authors are also working in the implementation of new modules based on FPGAs (Field Programmable Gate Arrays) that can be used in these networked control systems.

1 INTRODUCTION

Nowadays, the commercial buses and coaches are more and more equipped with electronic devices that make it easier to drive the vehicle and improve their security and comfort. These electronic devices are applied to functions such as Electronic Stability Program (ESP), braking help system (ABS), gear control, light control, climate control, opening door control, navigation and guide (based on GPS: Global Positioning System and GIS: Geographic Information System), etc. These functions require the use of reliable and real time exchange of information between the different control systems and the sensors and actuators. Thus, it becomes necessary the use of an industrial communication network or fieldbus (Marsh, 1999).

At the beginnings of 1980s the engineers of the automobile manufacturers assessed the existing fieldbuses for their using in vehicles. They came to the conclusion that any of these protocols fulfilled their requirements. It supposes the development of new fieldbus protocols (Mariño, 2003).

Each manufacturer has bet for a particular solution. For example, Bosch developed the CAN protocol, Volkswagen implemented the A-BUS, Renault and the PSA Consortium used the VAN protocol, BMW tried it with the M-BUS and Honda with the DLCS. The majority of these manufacturers evolved and adopted for the general purpose communication the CAN standard (Bosch, 1991) (ISO 11898, 1992) (ISO 11519-2, 1995).

For other functionalities, such as low speed smart sensors, multimedia, high speed and safety applications, the manufacturers are adopting other protocols in the last years. For example, the Firewire (IEEE 1394), MOST (Estevez, 2004), D2B optical and D2B Smartwirex are used for high speed multimedia applications; TTP, Byteflight and FlexRay for high speed and safety applications; and LIN for low speed smart sensor communication (Marsh, 2005) (Bender, 2004).

2 MULTIPLEXED SOLUTIONS

The buses and coaches used in the city public transport have a lot of onboard electronic systems, which must be integrated in an efficient way. These systems must be connected to a single industrial communication network with a Central Electronic Control (CEC) node that manages, in real time, all the information transmitted from the control modules installed in the vehicle.

There are several chassis manufacturers (Man, Volvo, Scania, Iveco, Renault, Mercedes, etc.) and every one proposed a different multiplexed solution. But the use of the CAN protocol and other fieldbus protocols based on CAN (for example SAE J1939) (SAE, 2005) is a common point in these solutions.

The multiplexed networks installed in the city-buses and coaches get an important reduction of the wiring that involves a reduction in costs, less breakdown risks and a simple scalability. The maintenance is easier and the global management of the technological systems is improved. The transmission of the information is integrated with a very low error rate. The control of the systems is in real time supporting very high elemental information traffic with command messages for actuators, data from sensors and alarm events.

2.1 Structure of the Network

A generic structure of communication used in the multiplexed solutions implemented currently in city public transport buses is presented in this section.

The main problem in the public transport buses is the increase of the requirements imposed to the electronic and electrical systems. It is getting more and more difficult to find adequate places for plugging the components with the corresponding wiring. The installation of circuits based on relays, diodes and resistors is very complicate.

Other topic to take into account is the satisfaction of the customer needs in a flexible and fast way. It is difficult and expensive. A manual process according to the documentation must implement the relay switching. The electromechanical circuits have a limited lifetime. The use of electronic switches in power circuits is a good solution because achieve the life time of the vehicle, increasing their reliability and reducing the time out of service.

Due to the problems related before, the CAN data bus has become into the automobile technology. Thus, an example of a multiplexed solution based on the CAN protocol for data communication in a city public transport bus is shown in figure 1. Independent data fieldbuses are used for the different areas in the system.

The bus manufacturers have trend towards the use of the VDV (Verband Deutscher Verkehrsunternehmen) recommendation 234 (VDV, 1996) for the onboard information system integrated in the dashboard. Therefore, the central unit must be equipped with a VDV interface module.

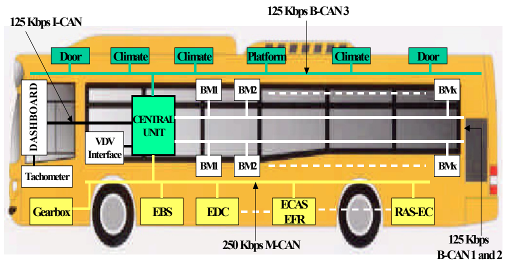


Figure 1: An example of multiplexed solution.

The central unit must execute the classic functions for the vehicle control and manage the data traffic from the different CAN fieldbuses. Thus, the central unit can report a full onboard diagnosis.

A typical multiplexed solution must consider the integration of a video surveillance system and the information for the passengers. This integration can be implemented using a CAN network or other fieldbuses more adequate for multimedia information (J1708, WorldFIP, IEEE 1394/Firewire, etc.). An example of the levels in a complete communication system is shown in figure 2.

2.2 Reliability

The chassis and body electronic reliability is very important in the public transport buses and coaches. The central unit should not connect and disconnect any power circuit. Based on city-bus manufacturers knowledge, under this rule, the central unit will only be implicated in one of each one thousand failures.

Three body CAN buses are contemplated to communicate several body modules with the aim to reduce the consequences of a breakdown. These body modules should be protected against a global overload with two security devices. Therefore, the effects in case of failures are minimized.

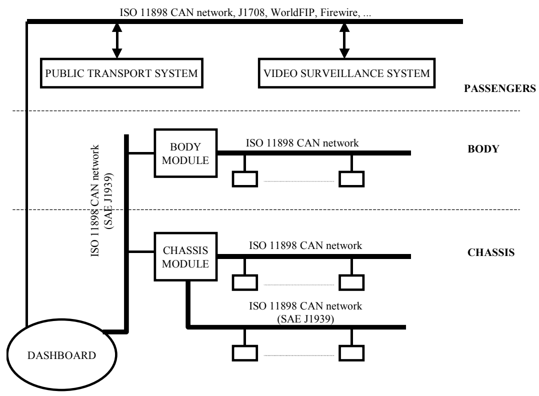


Figure 2: Levels in a typical city public transport bus.

The modules control their outputs according to the programmed emergency function when failures are detected in the data fieldbus. In this way, the outputs can be permanent connected, permanent disconnected, intermittent connected or maintaining the last state.

Other important application to get a good reliability is that the body and chassis electronic system supports a wide diagnostic. The central unit should manage all the CAN networks and implement their own diagnostic of the body and chassis electronic system. Every output of the body modules should be revised periodically to check that there are not short circuits, overloads and interruptions. The inputs should be controlled according to the technical possibilities of the connected sensors.

2.3 Challenges

Every chassis manufacturer has developed its own multiplexed solution based on CAN. The great inversions made from manufacturers imply that every one wants to impose its solution. The integration problem is for the coachbuilder that works with chassis from different manufacturers (ten or more chassis). Each chassis has a different multiplexed solution and only the modules chosen by the manufacturer can be used. It involves that the coachbuilder cannot have their own CAN system (an unified management for the electrical part of the body and chassis) and must use the modules chosen by the manufacturer with a non-competitive price.

The SAE Truck and Bus Control and Communications Sub-committee has developed the J1939 standard. The J1939 standard is necessary to sort the codifications that each manufacturer has used to specify the same peripheral unit (device, sensor) or the set of the units connected to the CAN network in the city-buses and coaches. Besides of the codification of the terminals, the standard defines several common parameters and data rates. Thus, the J1939 standard enables to the coachbuilder connecting different modules to the CAN system independently of the manufacturer.

3 DESIGN OF A NETWORKED CONTROL SYSTEM

The authors are working in the design of a networked control system that is a multiplexed solution to integrate the onboard electronic devices present in a typical city public transport bus. This system should integrate all the sensors and actuators present in the vehicle in an optimum way. Besides, it must resolve those particular functions that actually are not integrated in the multiplexed solutions of the chassis manufacturers (Domínguez, 2006).

The paper will describe an example of design for a vehicle of the Sweden chassis manufacturer Scania. The features of the system, used modules, software and requirements that must be taken into account will be commented in the next paragraphs.

3.1 Implementation of the System

The aim of the project exposed in this paper is to implement an onboard network in a city public transport bus. This network allows the control of the ECUs simplifying the wiring, removing electrical components, improving the reliability and the diagnostic and getting an open system.

There are different types of modules that are necessary in the system: dashboard and chassis and body modules (figure 2). The dashboard is where the information about the devices of the bus is displayed to the driver and also allows to the driver the control of the system through switch packs. The dashboard is usually made up of an Information Control Unit (ICU) and a Screen Control Unit (SCU). It includes a LCD or TFT VGA display where the information is shown to the driver. The dashboard also has multifrequency audio devices and LEDs to give information about state of the devices and alarms.

The French firm ACTIA supplies the chassis and body modules used in this design. There are master modules or CAMUs (Central Management Units) and slave modules or IOUs (Input/Output Units) and are shown in figure 3. These modules have several inputs and outputs of different types and implement the CAN V2.0 B protocol. These modules must be ready to work in the conditions of a vehicle in motion. Thus, the protection level must be high (IP65) and must endure the vibrations and fulfil the electromagnetic compatibility standards.

CAMU IOU

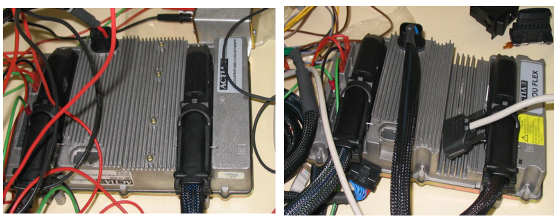


Figure 3: Chassis and body modules.

3.2 Inputs and Outputs

The first step in the design is the identification of the electrical signals of the bus that must be connected to the networked control system. These signals must be described in detail with their location in the bus. Some of the signals included in the control system in the Scania bus of the project are listed in table 1.

The signals are named according to a specified format. The first part of the name defines some characteristics of the signal. Some prefixes used in Table I are: IB (Input Binary) and OB (Output Binary). For example, the signal IB_STOPREQ is a binary (B) input (I) that is activated when a stop is requested. The column type shows additional information about the signals. For inputs, VBAT or GND active indicates whether the signal when activated is connected to the power supply or to the ground. For outputs there can be types like: signal (digital signals not intended for powering devices), lights, inductive (protection needed against power peaks), valve, etc.

Once the signals have been defined, the next step is to choose the number of required CAMUs and IOUs, their location in the bus and where the electrical signals will be connected (the connector pin of a specific CAMU or IOU). The designer must bear in mind the different types of inputs and outputs of the modules. The CAMUs and IOUs have the following types of inputs and outputs:

Table 1: Some electrical z in the SCANIA city-bus.

Name	Type	I/O	Power
IB_MSDOOR1	VBAT ACTIVE	Input	-
IB_MSDOOR2	VBAT ACTIVE	Input	-
IB_MSDOOR3	VBAT ACTIVE	Input	-
IB_STOPREQ	VBAT ACTIVE	Input	-
IB_RAMPREQ	VBAT ACTIVE	Input	-
IB_HANDBRAKE	GND ACTIVE	Input	-
IB_ALTERNATOR	VBAT ACTIVE	Input	-
IB_SMCSEC	CAN	Input	-
OB_REARMDOOR1	VALVE	Output	25 W
OB_DOOR1	SIGNAL	Output	-
OB_DOOR2	SIGNAL	Output	-
OB_DOOR3	SIGNAL	Output	-
OB_POSLIGHTS1	LIGHTS	Output	25 W
OB_ENGINEON	INDUCTIVE	Output	240 W
OB_HANDBRAKEBUZZ	INDUCTIVE	Output	3 W

- Wake up inputs:
  The signals that should wake up the system are connected in these inputs.
- Logic inputs:
  The detection of a “0” logic is from 0 V up to 1.8 V, and a “1” logic from 7 V up to 32 V.
- Analog inputs to ground:
  The voltage in the input of the microprocessor is directly proportional to the value of the resistive load.
- Analog inputs to positive voltage:
  These inputs consist in a resistive divisor.
- Frequency inputs:
  These inputs can be used for PWM signals, as for example the measurement of speed.
- High state and low state outputs:
  The maximum direct current of these outputs can be 9 A, 7 A, 5 A, 3.2 A, 2 A and 1.5 A.
- Free wheel diode output:
  It is dedicated to inductive loads (relays, electro valves) as for example wiper fast.
- Switched and unswitched outputs:
  The power supply of the output interface of high state signals can be unswitched (connected directly to the battery) or switched (connected after the master relay).
- PWM (Pulse Wide Modulation) outputs:
  There are several outputs that can be used for PWM (0-100% with 10% step) or frequency (50-500 Hz with 50 Hz step) outputs.
- Bridge outputs:
  They are used for bidirectional motors (electrical windows, external wing mirror, etc.) or for current measurements.

The number of CAMus and IOUs must be choice taking into account the distribution of the signals in the bus and also the power consumption, because the modules define the maximum dissipated power by group of outputs, the maximum total dissipated power and the maximum permanent current. Therefore, 1 CAMU and 4 IOUs have been required in this design. The location of the modules and the distribution of the signals are shown in figure 4.

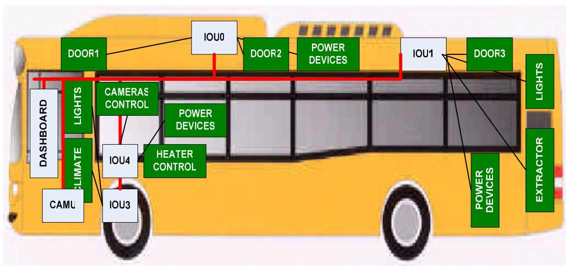


Figure 4: Structure of the networked control system onboard the bus.

3.3 Software Tools

There are several software tools used for the coding of the modules, validation and their installation: ActiGRAF (the wiring definition and inputs/outputs assignment), ISaGRAF (the functions specification), Multitool (diagnostic of the ECUs and CAN nodes) and Telemux (programming the ECUs).

For the design of a networked control system, ActiGRAF is the project manager and ISaGRAF is the programming environment. Thus, ISaGRAF is a tool used for ActiGRAF for developing the embedded software in the CAN modules of the control system. ISaGRAF supports the whole programming languages of the IEC 61131 standard.

An example of how the specification and implementation of functionality is made using Function Block Diagram (FBD) programming language is shown in figure 5. The user indicates the activation condition of the outputs depending on the state of the inputs with typical function blocks. Figure 5 shows a programming example where the aisle lights of the bus are switched on if there is battery voltage higher than 20 V (the master relay has been activated), the parking lights are turned on and the switch of the aisle lights is activated.

When the user designs the networked control system of a bus, the ActiGRAF tool is used to specify the central network of the vehicle (all the master modules or CAMUs), the intra-system network (all the slave modules or IOUs), the inter-system network and other networks that can be necessaries according to the requirements for the electrical architecture of the vehicle. After, the opening of the communication ports should be made (enable the CAN and J1939 drivers) and the input and output signals should be declared. The ISaGRAF tool is used to define the whole functionalities of the chassis and body functions of the bus using the more adequate programming language of the above mentioned.

Once the whole functionality and the wiring are defined, it is necessary to build the application that will be executed inside the modules. ActiGRAF uses a C compiler for compiling the source code generated by ISaGRAF. The compiler used for these tasks is from the Keil Company. The used version is for the C166 platform along with the RTX166. The binary code generated by the compiler is loaded using the Telemux application. Only the CAMU is reprogrammed with the binary code. As soon as the system is restarted, the CAMU automatically sends the new application and wiring data to all the IOUS of the system.

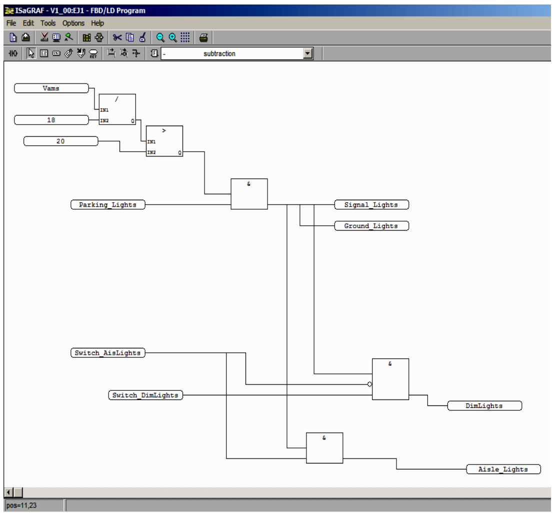


Figure 5: Example of the specification of a function in a bus using the ISaGRAF tool.

4 FUTURE WORKS

The authors are working in the design and implementation of new communication modules to improve the multiplexed networks used currently in city public transport buses. These modules should integrate all the sensors and actuators in an optimum way. Besides, these modules must resolve those particular functions that are not integrated in the multiplexed solutions of the chassis manufacturers.

These new communication modules will be designed using reconfigurable circuits technology as for example FPGAs (Lías, 2000). The implementation of communication processors for fieldbuses using FPGAs has a lot of advantages on account of their reconfiguration ability (Valdés, 2004).

The modules will be designed to enable the integration of the whole functions existent in the public transport buses and to add other new ones that are interesting for the coachbuilder such as closed-circuit TV, infrared control for doors, modules with outdoor connection, etc.

All these new functions must be integrated in the driver display. Thus, it is interesting the development of a system to integrate in the dashboard of the city-buses a PDA with the VDV protocol and that complements the information of the display included in the dashboard. The PDA could be integrated in the CAN networked control system using a VDV node that transmits the information to the PDA using a Bluetooth link. The VDV module can be designed with a FPGA that implements the CAN protocol in accordance to the J1939 standard and that manage the communication of a simple Bluetooth device (Flooks, 2005).

Other important improvement to be taking into account in the design of these modules is the possibility of integration of the localization and fleet management systems by GPRS, GSM, radio, etc. The integration of these systems enables the localization of the vehicles from the head office, the automation of the displaying systems for driver and passengers, the notification of next stop, estimated time to arrive to the bus stop, etc.

5 CONCLUSIONS

The main objectives of the work exposed in this paper is the improvement of the control system onboard the public transport buses. The authors design a networked control system based on modules with CAN communication. Thus, the advantages of this system used to integrate the electronic devices in a real time and reliable information system are:

- Development of a networked control system that satisfies the maximum demands of any public transport enterprise.
- Reduction of cables and number of electrical components (relays, fuses, etc.).
- Unification of the electronic equipment.
- The system has a central memory for the registration of alarms and maintenance.
- Autodiagnostic of the system.
- Improvements in the vehicle working control and the maintenance management.
- Improvements in the comfort.
- Best reliability of the components.
- Less maintenance costs.
- A flexible and modular system is obtained.

The design of modules based on FPGAs and fulfilling the J1939 standard and the VDV recommendation 234 is a very interesting solution for the coachbuilders. Accordingly, they can have their own CAN networked control systems and install in the public transport buses their own compatible devices to control the different chassis and body functionalities.

ACKNOWLEDGEMENTS

This work has been sponsored by an R&D project from the Autonomous Government (Galicia, Spain), Ref.PGIDIT05TIC011E. This work has been made in collaboration with the coachbuilder enterprise Castrosua S.A. and the Actia S.A. Company.

REFERENCES

Bender, M., September 2004. Introducing the MLX4: a microcontroller for LIN. *EDN Europe*, pp. 22-26.

Bosch, September 1991. *CAN specification Version 2.0*, Robert Bosch GmbH.

Domínguez, M.A., Mariño, P., Poza, F., Otero, S., 7-10 November 2006. Industrial communication system in technology control for public transport vehicles. In *Proceedings of 32nd Annual Conference of IEEE Industrial Electronics Society (IECON´06)*, ISBN 1-4244-0136-4, pp. 585-590.

Estevez, M., October 2004. MOST and MPEG: a perfect relationship?. *Embedded Systems Europe*, pp. 36-38.

Flooks, S., June 2005. Putting EDR to the test. *Electronic Design Europe*, pp. 8-9.

ISO 11898, 1992. *Road Vehicles – Interchange of digital information – Controller Area Network for high-speed communication*, ISO.

ISO 11519-2, 1995. *Road Vehicles – Low-speed serial data communication – Part 2: Low-speed Controller Area Network (CAN)*, ISO.

Lías, G., Valdés, M.D., Domínguez, M.A., Moure, M.J., September 2000. Implementing a fieldbus interface using a FPGA. *LNCS 1896*, Springer-Verlag, pp. 175-180.

Mariño, P., 2003. *Enterprise communications: Standards, networks and services*, Ed. RA-MA. Madrid, second edition.

Marsh, D., September 1999. Automotive design sets RTOS cost performance challenges. *EDN*, pp. 32-42.

Marsh, D. (editor), July 2005. Engines of change, *EDN Europe*, pp. 58-73.

SAE J1939, Revised January 2005. *Surface Vehicle Recommended Practice*, SAE.

Valdés, M.D., Domínguez, M.A., Moure, M.J., Quintáns, C., September 2004. Field Programmable Logic and Applications: A reconfigurable communication processor compatible with different industrial fieldbuses. *Lecture Notes in Computer Science 3203*, Springer-Verlag, pp. 1011-1016.

Verband Deutscher Verkehrsunternehmen (VDV), June 1996. *Driver's Workplace in the Low-Floor Line-Service Bus – Recommendation 234*, ISO/TC 22/SC 13/WG 3 N226.

PC-SLIDING FOR VEHICLES PATH PLANNING AND CONTROL

Design and Evaluation of Robustness to Parameters Change and Measurement Uncertainty

Mariolino De Cecco, Enrico Bertolazzi, Giordano Miori, Roberto Oboe
Dep. Mech Struct Eng, University of Trento, Via Mesiano 77, Trento, Italy
mariolino.dececco@ing.unitn.it

Luca Baglivo
Dept. Mech. Eng., Universit of Padova, via Venezia 1, Padova, Italy

Keywords: Wheeled Mobile Robot, WMR, path generation, path control, nonholonomic, clothoid.

Abstract: A novel technique called PC-Sliding for path planning and control of non-holonomic vehicles is presented and its performances analysed in terms of robustness. The path following is based upon a polynomial curvature planning and a control strategy that replans iteratively to force the vehicle to correct for deviations while sliding over the desired path. Advantages of the proposed method are its logical simplicity, compatibility with respect to kinematics and partially to dynamics. Chained form transformations are not involved. Resulting trajectories are convenient to manipulate and execute in vehicle controllers while computed with a straightforward numerical procedure in real-time. The performances of the method that embody a planner, a controller and a sensor fusion strategy is verified by Monte Carlo method to assess its robustness to parameters changes and measurement uncertainties.

1 INTRODUCTION

Path generation and control is the problem of determining a feasible set of commands that will permit a vehicle to move from an initial state to a final state following a desired geometrical figure in space while correcting for deviations in real time. While this problem can be solved for manipulators by means of inverting nonlinear kinematics, the common inverse problem for mobile robots is that of inverting nonlinear differential equations.

A basic method is therefore to plan a geometric path in the surface of motion, generally a 2D space, and conceive a suitable control strategy to force the vehicle to follow it. If the path is feasible its tracking will be accurate, otherwise there will be non negligible differences between the planned and the executed path.

If one plan a continuous curvature path, than can be sure of its compatibility with respect to kinematics and partially to dynamics if the maximum rate of curvature variation is taken into consideration. This for a huge variety of vehicles. As a matter of fact differential drive, car-like and all-wheel steering vehicles have constraints in curvature variation while moving.

Various methods have been employed to plan smooth trajectories (Rodrigues 2003). Some of them use splines (Labakhua 2006, Howard 2006, Solea 2006), other employ clothoids, generally in its linear curvature representation, to concatenate straight line segments with circumference arcs (Nagy 2000, Labakhua 2006). To cope with more complex representation of curvature, a method for trajectory planning based upon parametric trajectory representations have been developed (Kelly 2002). The method employs a polynomial representation of curvature. This is still a research field, obviously not for the geometric representations in itself (Dubins 1957), for the definition of numerical algorithms and control strategies efficiently employable in Real Time and for the systematic investigation of their robustness to parameters changes and measurement uncertainties.

Starting from the method of Kelly, we optimised the search strategy in order to extend the converging solutions. We also added a control algorithm that is perfectly integrated with the planning method.

The result is a Polynomial Curvature Sliding control, PC-Sliding, a novel RT procedure for planning and control that can be summarised as follows. The steering commands are designed by means of the polynomial curvature model applying a two-point boundary value problem driven by the differential posture (pose plus curvature). While following the path the vehicle replans iteratively the path with a repetition rate that must not necessarily be deterministic. To the actual curvilinear coordinate it is added a piece forward, than computed the corresponding posture in the original planned path, finally replanned the differential path steering the vehicle from the actual posture to the one just computed. The result is to force the vehicle to correct for deviations while sliding over the desired path. Those little pieces of correcting path have the property of fast convergence, thanks also to an optimised mathematical formulation, allowing a Real Time implementation of the strategy.

Advantages of the proposed method are its essentiality thanks to the use of the same strategy both for planning and control. Controlling vehicles in curvature assures compatibility with respect to kinematics and partially to dynamics if the maximum rate of curvature variation is taken into consideration. The method doesn't need chained form transformations and therefore is suitable also for systems that cannot be transformable like for example non-zero hinged trailers vehicles (Lucibello 2001). Controls are searched over a set of admissible trajectories resulting in corrections that are compatible with kinematics and dynamics, thus more robustness and accuracy in path following. Resulting trajectories are convenient to manipulate and execute in vehicle controllers and they can be computed with a straightforward numerical procedure in real-time. Disadvantages could be the low degrees of freedom to plan obstacle-free path (Baglivo 2005), but the method can readily be integrated with Reactive Simulation methods (De Cecco 2007), or the degree of the polynomial representing curvature can be increased to cope with those situations (Kelly 2002, Howard 2006).

Parametric trajectory representations limit computation because they reduce the search space for solutions but this at the cost of potentially introducing suboptimality. The advantage is to convert the optimal control formulation into an equivalent nonlinear programming problem faster in terms of computational load. Obviously the dynamic model is not explicitly considered thus missing effectiveness in terms of optimisation (Biral 2001).

By means of this iterative planning sliding control strategy we obtain a time-varying control in feedback which produces convergence to the desired path, guaranteeing at the same time robustness. In particular, small non-persistent perturbations are rejected, while ultimate boundedness is achieved in the presence of persistent perturbations.

Verification of stability convergence and robustness can be achieved analytically or statistically. The first way has the merit to synthesise the results in a general and compact fashion. By means of its analytic representation it is mostly easy to isolate the influence parameters and quantify its effect. The second way has the merit to cope easily with complex models where interact different effects. In the present paper we aim at verifying the performances of the proposed method that embody a planner, a controller and a sensor fusion strategy for the vehicle pose estimation that takes into account an iterative measurement system and an environment referred one (De Cecco 2003, De Cecco 2007). The fusion technique takes into account also systematic effects. This last part interacts with the control strategy injecting step inputs of different entity at each fusion step. For the above reasons we decided to take the second way of verification employing a Monte Carlo method.

Generally research focuses on control or measurement goal separately. Seldom the deep interaction between them is taken into consideration. In this work the whole system robustness is investigated by simulation.

2 PATH PLANNING AND CONTROL PROBLEMS

The main problem with Wheeled Mobile Robots path planning and control tasks is well known and it's strictly related to the nonholonomic constraints on velocity. These constraints limit possible instantaneous movements of the robot and cause the number of controlled states to be less than the control inputs one. WMR state equations constitute a non linear differential system that cannot be solved in closed form in order to find the control inputs that steer the robot from an initial to a goal posture (position, heading and curvature). As a consequence also the control task is non standard with respect to the case of holonomic systems like the most part of manipulators. A suitable control law for precise and fast mobile robots path following should compute feasible inputs that generate correcting paths compliant with nonholonomic constraints.

2.1 Generalized Clothoids

Since the robot velocity vector (and therefore its forward axis) has to be aligned with the path tangent, it is natural and convenient to include the curvature as a state while describing the robot state equations. In fact the curvature is directly related to the steering actuator input, therefore its continuity is an important issue to prevent path following deviations due to the planning phase.

A system state model that can be used for a car-like or differential drive robot is:

$$\begin{aligned} \dot{x}(s) &= \cos\delta(s) \\ \dot{y}(s) &= \sin\delta(s) \\ \dot{\delta}(s) &= k(s) \\ \dot{k}(s) &= u(s) \end{aligned} \tag{1}$$

where the state vector (posture) consists of the position coordinates x, y, heading δ and curvature k. Assuming, without losing generality, that $k(s)$ is a control input, the last equation in (1) can be omitted. The derivatives are expressed with respect to the arc length s rather than time, considering the robot velocity as an independent input that doesn't affect the path geometry except for actuators dynamics limitation.

Choosing for curvature a third order polynomial in arc length allows to steer the robot from a given starting to a final posture along a feasible smooth path. In other words, the problem is that of generating a continuous curvature path connecting the two end-postures :

$$\begin{aligned} \underline{x}(s_0) &= [0,0,0,k_0] \\ \underline{x}(s_f) &= [x_f, y_f, \delta_f, k_f] \end{aligned} \tag{2}$$

The first constraint in equation (2) is anyway general if the reference system is placed with its origin at the initial position and oriented aligned with the initial heading. In this way the first three constraints are automatically satisfied and problem is reduced to satisfy the remaining five constraints, that is initial curvature and final posture.

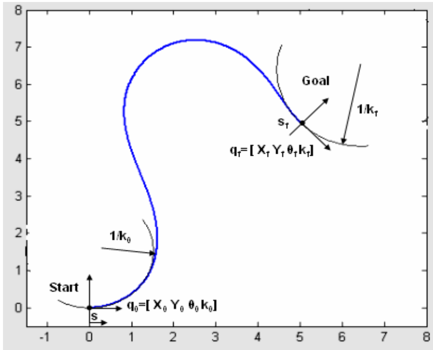


Figure 1: Continuous cubic curvature path planning. A solution example.

The problem requires the solution of a nonlinear differential equations system. In fact, while the heading is obtained as a simple integration of the curvature polynomial, the calculation of the cartesian position [x, y] requires the numeric integration of generalised Fresnel integrals. An effort to solve the inverse two points boundary problem (Figure 1) is worth thanks to the many advantages that this formulation leads to. The aim is to calculate five parameters, that in this case are the four coefficients of the curvature polynomial, and the total arc length s_f. Once the calculation algorithm has been designed and optimised, one could have, in a few parameters, a representation of the planned path that is nonholonomic compliant and can be generated in real time. Examples can be an autonomous path planning of a fork lift which have to reach a detected pallet or to control a car in a fully automated car parking. Besides, part of the solving algorithm is exploited to generate the forward integration of the curvature polynomial and the input curvature can be applied to a large variety of WMR, according to the kinematic model and dynamic constraints, without changing the planning and control algorithm.

3 THE PROPOSED METHOD

Starting from the method of (Kelly 2002), we optimised the search strategy in order to extend the converging solutions upon a large range of possible final configurations (Figure 2).

The control algorithm we designed has revealed to be efficient and very well integrated with the planning method. It uses the same planning algorithm to calculate feedback corrections, asymptotically reducing servo errors to the reference path.

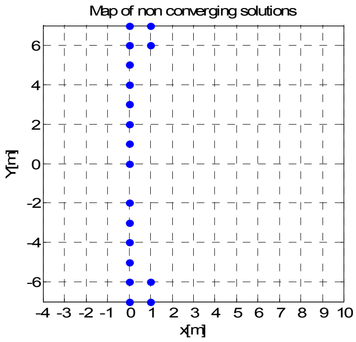


Figure 2: Validation map of end postures without convergence.

Table 1: Validation range.

Parameter	Min	Max	Step
x	0	10	1 m
y	-7	7	1 m
δ	-π/2	π/2	0.1 rad
k	-0.1	0.1	0.02 m^{-1}

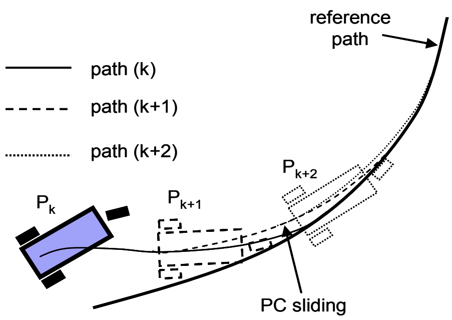


Figure 3: Schematic representation of the control method. At $k \cdot T_{PC}$ time-spaced instants a PC-sliding path is computed to reach a sliding subtarget thus forcing the robot to keep the desired planned PC path.

The whole algorithm can be summarized as follows:

[*Initialisation*]

a) a reference PC path is planned, mapping the arc length interval $[0, s_f]$ into the postures $[x(s), y(s), \delta(s), k(s)]$, that describe the robot posture evolution from an initial posture defined $P_0 = [x_0, y_0, \delta_0, k_0]$ to the final desired posture $P_f = [x_f, y_f, \delta_f, k_f]$ within defined tolerances;

b) at initial condition the robot is placed in any initial posture also different from P_0;

c) if initial planning was successful (all constraints satisfied), start moving at constant velocity V.

[*Loop at T_{PC} cycle time*]

d) get actual position estimate and compute the minimum distance position on the main reference path and its corresponding arc length s. Add to s an additional defined length, Δs, proportional to velocity. If $s + \Delta s < s_f$ $s_g = s + \Delta s$, otherwise $s_g = s_f$;

e) compute the correcting PC path by applying point b) to plan a path between actual posture and that mapped by s_g on the reference PC path $[x(s_g), y(s_g), \delta(s_g), k(s_g)]$, see Figure 3, *path(k)*;

f) If $s + \Delta s > s_f$. Reduce velocity, set the steering input according to the input curvature k;

g) set the steering input according to the input curvature k;

h) if final boundary condition is satisfied then STOP;

i) GO TO point d)

4 IMPLEMENTATION

Real time feasibility was analyzed taking in mind possible applications of industrial AGV transpallet. The simulation tests, reported in §5.1, showed good results in terms of fastness, low overall tracking error and robustness. The RT implementation is at the phase of Real Time cycle time verification and architecture design.

4.1 Simulation

Simulations were carried out involving a three wheeled robot (De Cecco 2003): 500 mm wheelbase and a 50 mm front wheel radius. The path control model incorporates the path planning and control. The actuators dynamics is taken into account. A model of a sensor fusion technique closes the loop by feeding the current posture to the path controller (see Figure 4). The sensor fusion algorithm combines odometric and triangulating laser pose estimates also taking into account systematic and random effects (De Cecco 2007).

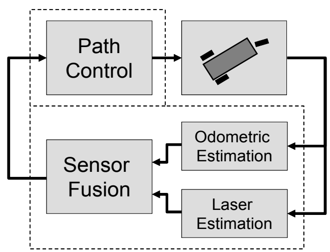


Figure 4: Logic scheme of the simulation model.

The simulated vehicle has two control inputs, velocity and steering angle of the front wheel. The actuators dynamics are simulated by means of first order model with a time constant for the steering mechanism of 0.1 s and that of the driving motor of 0.5 s. PC sliding method is updated with a refresh cycle time T_{PC} of 0.015 seconds. A value of $s_f/20$ for Δs was used for driving velocity of 1.5 m/s.

4.2 RT Implementation

The PC-sliding algorithm was implemented on a National Instruments 333 MHz PXI with embedded real-time operating system (Pharlap).

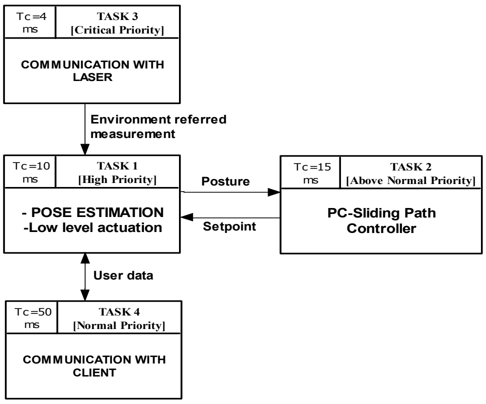


Figure 5: architecture for real time implementation.

The designed and preliminary verified software architecture has four main tasks (see Figure 5):

• TASK 1 – Pose estimation & Low level actuation: computes the best pose estimate starting from odometers and laser data and computes the reference commands to the drivers according to the actual planned path.

• TASK 2 – PC-Sliding Path Controller: computes the corrective control action based upon its current pose and the target one;

• TASK 3 – Communication with Laser: acquires the new pose measurement from laser when ready and makes it available for task 1

• TASK 4 – Client: communicates with a reference station to manage the missions start-stop, acquire and store data, etc.

The priority (static-priority) of the tasks was assigned according to rate monotonic algorithm which assign the priority of each task according to its period, so that the shorter the period the higher the priority.

Mean calculation times were measured for the PC-sliding planning algorithm. After optimisation a large number of paths were computed. The iteration termination condition was triggered when a weighted residual norm defined as:

$$r = \sqrt{(w_x \Delta x_f)^2 + (w_y \Delta y_f)^2 + (w_\delta \Delta \delta_f)^2 + (w_k \Delta k_f)^2} \qquad (4)$$

is under a threshold of 0.001, and the weights are computed in such a way that a fixed error upon final x_f or y_f or δ_f or k_f, alone exceed the threshold. First two weights in equation (4) were chosen equal to 1 m^{-1}, last two weights were chosen to be equal to the root square of ten. The computation time over all the iterations showed in Figure 2 (only paths converging under threshold) spanned from a minimum of 0.0003 to a maximum of 0.005 seconds. The termination condition was thought to obtain a feasible path for an industrial transpallet that has to lift correctly a pallet.

5 VERIFICATION

In this work the whole system robustness is investigated by simulation. We decided to evaluate statistically rather than analytically the convergence and the stability of the proposed method. The main motivation is the aim to investigate not only the influence of system delays and parameters bias on the control, but also the interaction between a near-reality measurement system used as the source of feedback to the path controller. Monte Carlo analysis is a powerful tool for this kind of tasks.

5.1 Simulation Results

Simulations were aimed at verifying control robustness toward different aspects related to:

A. approximation of forward integration;
B. actuators delay and inaccuracy
C. non ideal initial conditions
D. control model parameters uncertainty
E. pose measurement noise

For all the tests the maximum following absolute residual *eps_path* and final position residual *eps_fin* were computed in meters.

A. The planned path is not an exact solution because of generalized Fresnel integrals cannot be solved in closed form and a reasonable computation time is required for real time implementation. Therefore a planning solution is accepted if the termination condition is satisfied. First simulation test was about verifying the vehicle model implementation in ideal conditions, that is without actuator dynamics, using nominal model parameters and starting from ideal initial boundary conditions. The path in Figure 6 has been chosen as the representative path for the analysis. Boundary conditions are:

$$\underline{x}(0) = [0,0,0,0] \qquad \underline{x}(s_f) = [-6,3,\pi,0] \tag{5}$$

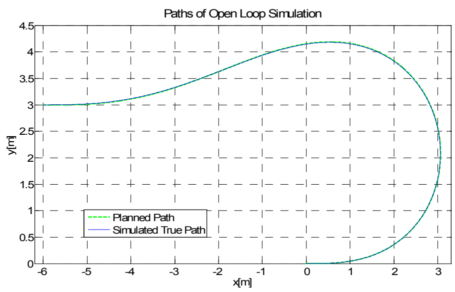


Figure 6: open loop ideal path, only approximation affect the following and final residual in this case. *eps_path*= 0.007 m, *eps_fin* = 10^{-4} m .

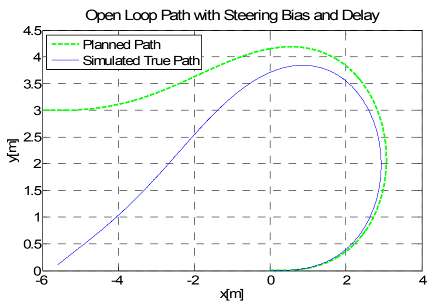


Figure 7: Open loop path with steering actuator biased and delayed.

B. Without changing any condition with respect to case A. except for the introduction of a dynamic steering actuator model with a time constant of 0.1 seconds and a steering actuator bias of 1 degree the resulting path is the one in Figure 7. To note that the effect of the steering actuator is the most significant, while the actuator delay effect is negligible. A proof is that the simulated true path closes the bend more than required.

Benefit of the proposed feedback controller can be seen just making a comparison between the open loop 1 degree steering angle biased case in Figure 7 and that in Figure 8 which is a result of close loop applying in the case of 5 degree biased steering actuation angle. The last could be considered a worst case due to heavy mechanical skew or simply to bad steering servo actuation. In this simulation the residuals were *eps_path* = 0.026 m, *eps_fin* = $4{\cdot}10^{-4}$ m.

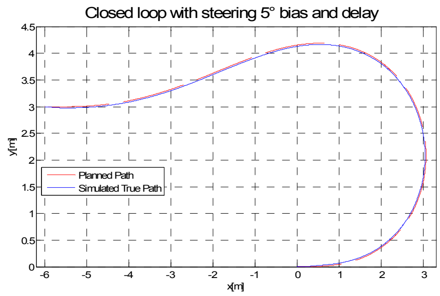


Figure 8: Close loop path with biased steering input angle

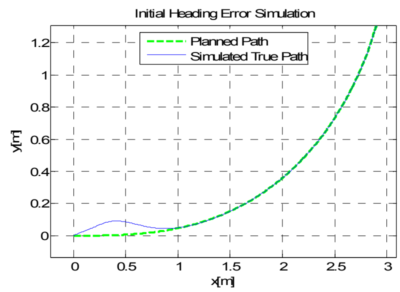


Figure 9: PC-sliding convergence to the reference path starting from an initial heading of 15 degrees.

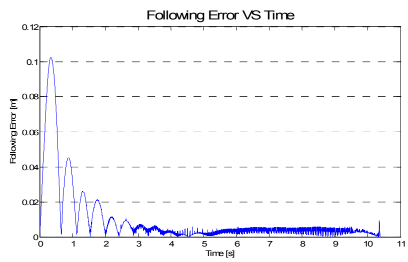


Figure 10: Residuals of the path following of the previous figure.

C. Another significant simulation is the one concerning with a non ideal initial condition: an initial heading difference of 15 degree from the aligned condition (Figure 9). In this case steering delay is accounted too.

In Figure 10 it is showed that the following absolute residual decreases asymptotically remaining always bounded within reasonable values.

D. A set of simulations concerning the robustness with respect to control parameters uncertainties has been achieved. A Monte Carlo approach was employed to analyse control performances when an iterated randomized set of control parameters is used to carry out a PC-sliding path following task. More precisely, at each iteration step the parameters b (wheelbase) and α_0 (steering angle for a theoretical straight path) are randomly drawn from a normal unbiased distribution with standard deviation σ_b and σ_α and then a complete simulation, with same boundary constraints of the previously presented cases, is done. Setting σ_b = 0.002 m and σ_α = 1 deg, the performance results are those shown in Figure 11 and Figure 12.

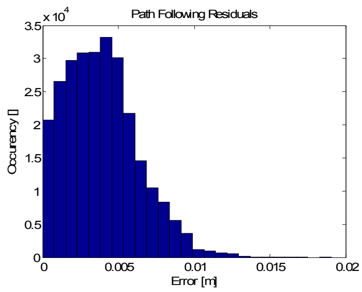


Figure 11: Position residuals along each trial path computed for each replan points at T_{PC} rate.

In this simulation the residuals were *eps_path* = 0.019 m, *eps_fin* = 10^{-4} m.

E. Last simulation tests are related to the analysis of control robustness with respect to pose measurement uncertainty. The first testing was made by feeding simulated fused pose measurements to the PC-sliding control algorithm. The measurement simulation model involves a sensor fusion algorithm that combines an odometric pose with a triangulating laser estimate (see Figure 4). While the odometric path estimate is smooth but affected by increasing systematic errors with time, the laser sensor furnishes unbiased but noisy pose measurements.

The sensor fusion algorithm compounds better characteristic of the two measurement systems, but the fused pose remains anyway affected by a certain bias and by a certain noise. The parameters used for the odometric model are the wheel radius R, the wheelbase b and the steering angle offset α_0. It was set σ_b = 0.002 m , σ_R = 0.0005 m and σ_α = 0.1 deg for kinematic model parameters uncertainties. For the laser pose measurement is reasonable to set the standard deviation $\sigma_x = \sigma_y =$ 0.015 m and that of robot attitude as σ_δ = 0.002 rad. All the parameters are ideal. Only the laser estimate is affected by noise influencing the fused pose proportionally to odometric uncertainty. Simulation residuals are reported in Figure 13 and Figure 14. In this simulation the residuals were *eps_path* = 0.024 m, *eps_fin* = 0.013 m.

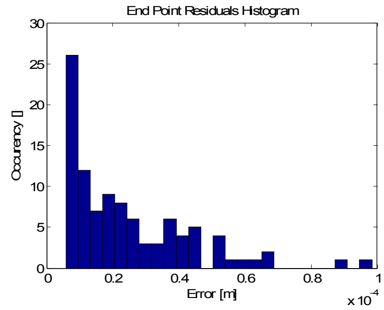


Figure 12: End-point residuals, one for each trial path.

Finally we carried out a simulation set where all the influencing factors were combined together. In this case the odometric model was given random parameters bias that were drawn randomly according to those of the control model except for an augmented steering actuation error, as such is expected to be in reality. The parameters used by odometric model were drawn from the same normal distributions which were supposed to be in the previous simulations set (σ_b = 0.002 m , σ_R = 0.0005 m and σ_α = 0.1 deg) while the control model is affected by the same wheelbase bias and by a 1 degree constant actuation error. In this simulation the residuals were *eps_path* = 0.064 m, *eps_fin* = 0.028 m. In Figure 15 the worst case in term of maximum following residual.

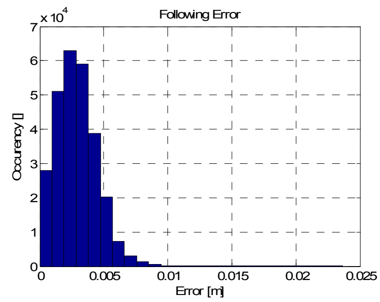


Figure 13: Position residuals along each path and for each trial path in the case of measurement noise influence.

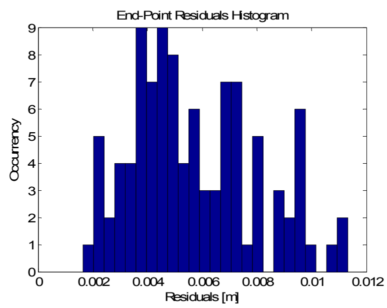


Figure 14: End-point residuals, one for each trial path.

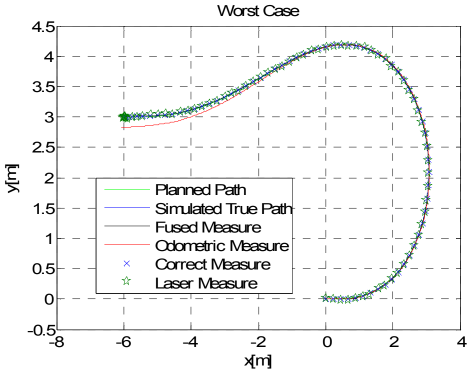


Figure 15: Worst case in term of maximum following residual in the combined influence factors simulations.

We would like to underline that in this preliminary verification it was carried out a limited number of iteration for the Monte Carlo analysis, $N = 100$.

6 FUTURE WORK

Future work envisage an experimental verification that will give an important verification of the method effectiveness. Nevertheless simulation results can be considered reliable from an in-principle point of view (Baglivo 2005).

A second track of research foresee an increase of the polynomial degree to achieve flexibility with respect to obstacles constraints and minimum curvature.

7 CONCLUSIONS

A novel technique for path planning and control of non-holonomic vehicles is presented and its performances verified. The performances of the method that embody a planner, a controller and a sensor fusion strategy was verified by Monte Carlo simulation to assess its robustness to parameters changes and measurement uncertainties.

The control algorithm showed high effectiveness in path following also in presence of high parameters deviations and measurement noise. The overall performances are certainly compatible with the operations of an autonomous transpallet for industrial applications. Just to recall an example, significant is the ability to compensate for a steering error of 5° over a path of 180° attitude variation and about 7 meters translation leading to a final deviation of only 0.5 mm in simulation.

REFERENCES

Baglivo L., De Cecco M., Angrilli F., Tecchio F., Pivato A., 2005 *An integrated hardware/software platform for both Simulation and Real-Time Autonomus Guided Vehicles Navigation*, ESM, Riga, Latvia, June 1st - 4th.

Biral F., Da Lio M., 2001 *Modelling drivers with the Optimal Manoeuvre method*, ATA Congress, firenze 2001, 01A1029, 23-25 May.

De Cecco M., Baglivo L., Angrilli F., 2007 *Real-Time Uncertainty Estimation of Autonomous Guided Vehicles Trajectory taking into account Correlated and Uncorrelated Effects*, IEEE Transactions on Instrumentation and Measurement.

De Cecco M., Marcuzzi E., Baglivo L., Zaccariotto M., 2007 *Reactive Simulation for Real-Time Obstacle Avoidance*, Informatics in Control Automation and Robotics III, Springer.

De Cecco M., 2003 *Sensor fusion of inertial-odometric navigation as a function of the actual manoeuvres of autonomous guided vehicles*, Measurement Science and Technology, vol. 14, pp 643-653.

Dubins L. E., 1957 *On Curves of Minimal Length with a Constraint on Average Curvature and with Prescribed Initial and Terminal Positions and Tangents*, American Journal of Mathematics, 79:497- 516.

Howard T., Kelly A., 2006 *Continuous Control Primitive Trajectory Generation and Optimal Motion Splines for All-Wheel Steering Mobile Robots*, IROS.

Kelly A., Nagy B., 2002. *Reactive Nonholonomic Trajectory Generation via Parametric Optimal Control*, International Journal of Robotics Research.

Labakhua L., Nunes U., Rodrigues R., Leite F. 2006. *Smooth Trajectory planning for fully automated passengers vehicles*, Third International Conference on Informatics in Control, Automation and Robotics, ICINCO.

Leao D., Pereira T., Lima P., Custodio L. 2002 *Trajectory planning using continuous curvature paths,* DETUA Journal, vol 3, n° 6.

Lucibello P., Oriolo G., 2001. *Robust stabilization via iterative state steering with an application to chained-form systems*, Automatica vol. 37, pp 71-79.

Nagy, M. and T. Vendel , 2000. *Generating curves and swept surfaces by blended circles*. Computer Aided Geometric Design, Vol. 17, 197-206.

Solea, R., Nunes U., 2006. *Trajectory planning with embedded velocity planner for fully-automated passenger vehicles*. IEEE 9th Intelligent Transportation Systems Conference. Toronto, Canada.

Rodrigues, R., F. Leite, and S. Rosa, 2003. *On the generation of a trigonometric interpolating curve in* $\Re^3$, 11th Int. Conference on Advanced Robotics, Coimbra, Portugal.

TASK PLANNER FOR HUMAN-ROBOT INTERACTION INSIDE A COOPERATIVE DISASSEMBLY ROBOTIC SYSTEM

Carolina Díaz, Santiago Puente and Fernando Torres
University of Alicante
Crta San Vicente del Raspeig s/n
PO Box 03080, San Vicente del Raspeig- Spain
cdiaz@dfists.ua.es, Santiago.Puente@ua.es, Fernando.Torres@ua.es

Keywords: Human-robot interaction, cooperative, disassembly, task planner.

Abstract: This paper develops a task planner that allows including a human operator which works cooperatively with robots inside an automatic disassembling cell. This method gives the necessary information to the system and the steps to be followed by the manipulator and the human, in order to obtain an optimal disassembly and a free-shock task assignation that guarantees the safety of the human operator.

1 INTRODUCTION

Disassembly is defined as the process of separating pieces that compose an object (Torres and Puente, 2006). In this process it is very useful to consider the advantages of cooperative tasks, in which two or more robots take part, or tasks in which the intervention of a human being is required (Adams and Skubic, 2005). Some of those advantages are: making tasks that a single robot can not do; sharing information and resources; greater tolerance to failures; and attending between manipulators and humans for different tasks. Working in a coordinated way also provides the system a faster and an effective disassembly, which allows a consequent saving of money to the industries that apply it.

The value of a group of entities collaborating among them, working in group as a team has been proven many times in many domains. For example, in nature a group of animals working cooperatively as a team, can manage to hunt a stronger and bigger animal. Also in the military service a group of men with limited resources and specific abilities are united to create groups with an incredible capacity. These examples illustrate that a group of entities with similar or different abilities joined to work in a team, can produces a work unit with abilities and capacities greater than the sum of its parts (Navarro-Serment, *et al.*, 2002). Including two or more agents working in a cooperative way increases the performance of the disassembly system, because of the synergy produces a group of units working together as a team.

Two groups can be distinguished in cooperative robots work field:

- Two or more robots working cooperatively to solve different tasks. This group is called robot-robot application for forward examples (Tinós and Terra, 2002; Fonseca and Tenreiro, 2003).
- Cooperative tasks in which robots manipulators and humans interact, named in this paper robot-human application (Kumar *et al.*, 2000; Hägele *et al.*, 2002).

The remarkable issue that differences these two groups is that when humans and robots interact, the system must consider more external and internal sensors in order to avoid humans suffering any physical damage.

It is important to highlight that this work tries to use the intervention of a human in task in which the person has more abilities and general comprehension than a robot. Robot manipulators transform in intelligent agents that assist humans in all kind of task and activities, taking advantages of the resources and characteristic of each agent and minimizing the negative properties collaborating between them.

In the present paper it is observed the advantages that bring to include a person working in a cooperative way inside a disassembling cell. Until recently in most of industrial environments the robot manipulator was isolated through securities fences, avoiding any possible contact or interaction with human operators. These were the methods for guaranteeing the safety of operators inside these environments (Corke, 1999; Kulic and Croft, 2005; Ikuta and Nokata, 2003). The present paper set up a task planner that allows human-robot interaction taking into account safety aspect.

This article is organized as follows: after the introduction in Section 2 the process' architecture is described. Then, in Section 3, the cooperative task planner is developed. In Section 4 an application example is explained. And finally conclusions and future works are presented.

2 PROCESS' ARCHITECTURE

The process' architecture used here is the same developed in a previous work (Díaz *et al.*, 2006); it is shown in Figure 1.

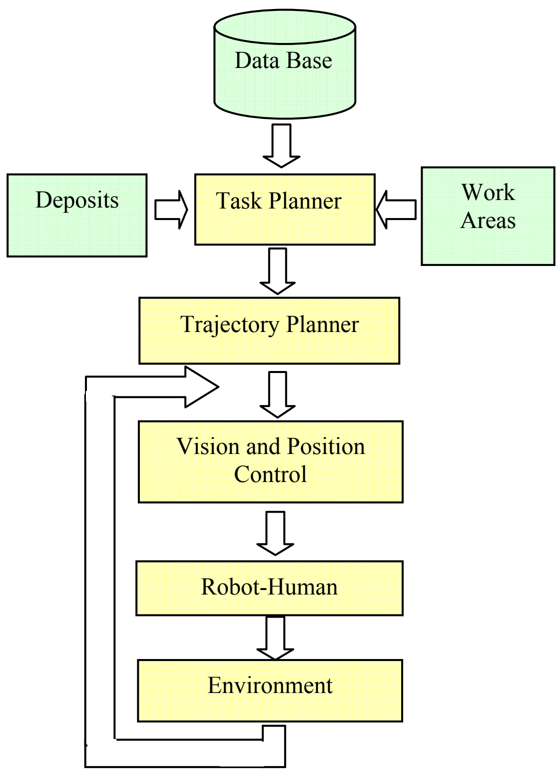


Figure 1: Process' Architecture.

In this scheme the Data Base contains a list of tasks for disassembling products, through a relational model graph developed in (Torres *et al.*, 2003). The Task Planner determines which action corresponds to each agent. Then a position and a vision control are applied to avoid collisions in real time between robots and humans, and also collisions of these with the environment. This grants the system the possibility of doing on-line corrections. This control is not developed in this paper.

The Task Planner has all the information of the layout of the cell, the storage deposits position, and the location of each agent work area and their intersection (Fig. 2). This information is very important in order to avoid collision, between robot and human and with the environment.

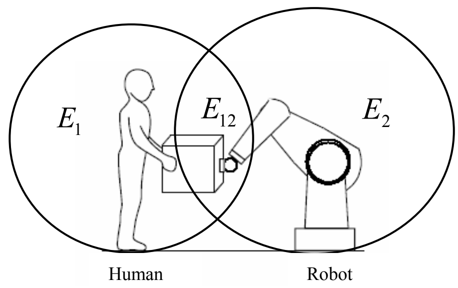


Figure 2: Scheme human robot working areas.

The Task Planner is the one that determines the sequence to be followed by the manipulator and the human who take part in the disassembly task; looking to obtain the maximum advantage of all the resources, and reducing the total disassembly time.

Is important to highlight the Vision and Position Control that is the one on charge to detection and avoid collision, is not develop in this paper. What is looking for is obtaining an optimal cooperative task planning that avoids possible collision in the intersection area, in normal condition. In case, for example, when accident takes place the Position and Vision control is the one that have to take the correct decision. In this project, it is also working with a special environment (Corrales *et al.*, 2006) that allow to monitoring the location of the human operator in real time.

3 TASK PLANNER

The Task Planner developed in this paper for robot human interaction is based on (Diaz *et al.*, 2006) for robot cooperative works; it is important to remark that only a few modifications have been necessary to adapt this Task Planner for human-robot interaction. This brings out the flexibility of the proposed method. The major modifications have to be done in another block of the system's architecture, like in the Vision and Position Control to allow the system monitoring the human and robot movement inside the cell.

To reduce risk factors inside an industrial cell in which robots and humans works Burke *et al.*, 2003 proposed three criteria:

- Redesigning the working cell looking for the way that the danger is eliminated.
- Control the danger thought sensor or physical limits.
- Warn and train the human operator which work in the cell.

In this paper the last two items of these criteria are considered; controlling the danger through a vision and position control and educating the person about the dangers of working cooperatively with robot manipulators in a disassembly system. The dangers are reduced considerably making a suitable task plan. The planner is the system that determines to which agent correspond each action execution and the precise moment to be executed, to obtain a successful disassembly free of collisions.

Given the relational model graph described on (Torres *et al.*, 2003), a hierarchical graph that represent the structure that sets up the product to be disassembled is obtained. This graph also contains all the actions to disassemble a product and gives much useful information, like the precedence and the parallelism between tasks. Crossing this graph the rules that specify the sequence to disassemble a product are obtained. In Figure 3 the component of a PC's mouse are shown and in Figure 4 the relational model graph for disassembly this products is observed.

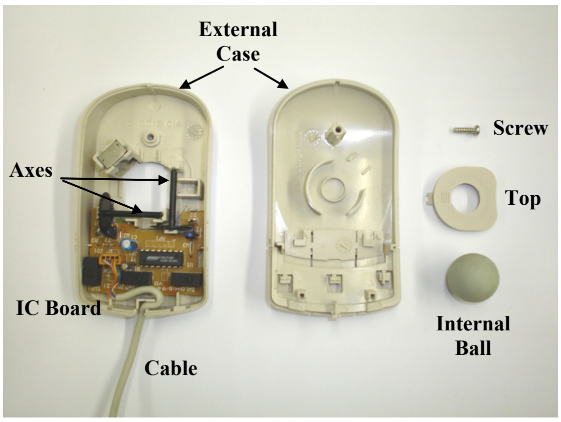


Figure 3: Components of a Pc's mouse.

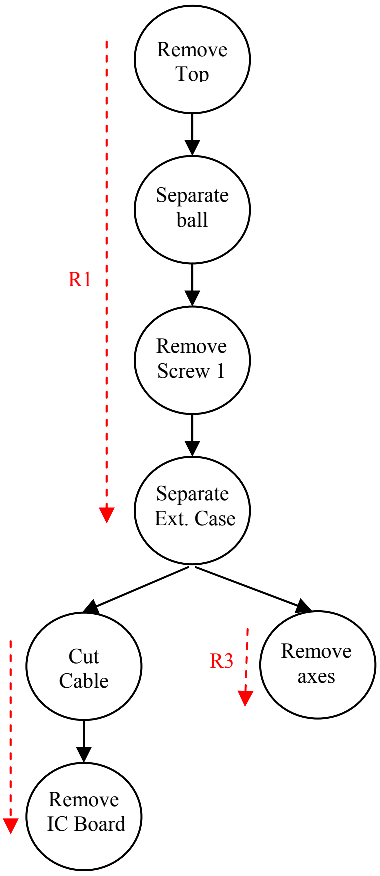


Figure 4: Hierarchical Graph to disassemble a PC's mouse.

In Figure 4 the product considered corresponds with a PC's mouse. In this case the rules for this object are:

Rule 1= Remove Top + Separate internal boll + Remove Screw 1+ Separate external case.

Rule 2 = Cut Cable + Remove CI Board.

Rule 3 = Remove axes.

The Task Planner based on these rules to constructs the decision trees that allocate the different tasks between robot and human, to obtain a cooperative and successful disassembly of a product.

In a working cooperative environment and taking into account the workspace intersection, two types of task are defined:

- Common Tasks: are those in which is required two or more entities working in the same specific object. For example the extraction of a CD player.
- Parallel Tasks: those in which each entity does a specific task. The presence of only one entity is required. It can be executed in simultaneous way. For example the extraction of a Card Slot and simultaneously the extraction of the Energy Source.

Modelling these rules and according to the type of task to be executed (*Tc* o *Tp*) the decisions trees are constructed. These determine the assignment of all the actions to be done; to disassemble the product in an optimal and cooperative way.

From the relational model graph the different rules are obtained. These rules are divided into actions **A**, for each action corresponds a tool **T**, and each action is divided into sub-action if it is possible.

In general, to construct the decision trees and to model the system, the following sets are defined:

Number of Robots = $\left[R_1, R_2, ..., R_i, ..., R_j\right]$

Number of Humans = $\left[H_1, H_2, ..., H_i, ..., H_j\right]$

Task's Type = $\left[Tc, Tp\right]$

where: *Tc:* Common Task.
Tp: Parallel Task.

Rules = Task = $\left[Ts_1, Ts_2, ..., Ts_m\right]$

each task is divided in actions.

Actions = $\left[A_1, A_2, ..., A_n\right]$

and each action is divided into sub-actions

$$\Rightarrow \quad A_1 = \left[A_{11}, A_{12}, ..., A_{1p}\right]$$
$$A_2 = \left[A_{21}, A_{22}, ..., A_{2q}\right]$$
$$\vdots$$
$$A_r = \left[A_{r1}, A_{r2}, ..., A_{rs}\right]$$

For each action, a respective tool exists. In other words, it exist the same number of actions as tools:

Tools = $\left[T_1, T_2, ..., T_n\right]$

In tasks in where robot and human cooperate. The actions are assigned to the human due to their qualities and abilities. It is obvious that the hands are considered as the tool that the worker used to execute action.

According to the sets described before and to the type of task, the trees that determine the optimal allocations of the actions were constructed like are developed in (Díaz *et al.*, 2006). In order to determine the optimal path an information gain is empirically assigned for each robot or human. In this work the costs are assigned according to the characteristics of each action. Time is the most important characteristic in this application.

There are to highlight that the system does not handle with synchronizing the task between human and robot, it only perform the distribution of task between them. The synchronization between them is ensured by the vision and position control system.

4 APPLICATION EXAMPLE

Here a disassembled cooperative task is executed working in a cell compose by only one human operator and one robot manipulator Mitsubishi® PA-10. Named H_1 and R_1 respectively

Modeling from Rule 1 obtained from the relational model graph shown in Fig. 4 it is obtained:
Rule 1= Remove Top + Separate internal boll + Remove Screw 1+ Separate external case.

It is sub-divide into two tasks:

T_{s1} = *Remove internal ball*
= Remove Top + Separate Ball

T_{s2} = *Separate external case*
= Remove Screw 1 + Separate Case.

Executing T_{s1} for this application is divided into two actions according to the corresponding tool used to execute each action. To execute T_{s1}, first one of the entities has to hold the mouse while the other removes the top, then:

T_{s1} = *Remove internal ball*
= Grasp mouse + Remove Top + Separate Ball

where: A_1 = *Grasp mouse and Separate ball*
(Parrallel Jaw)
A_2 = *Remove top*
(Handing Extraction)

$A_1 = A_{11} + A_{12}$ it is sub-divided in two actions:
where: A_{11} = *Grasp Mouse*
A_{12} = *Deposit Ball*

$A_2 = A_{21} + A_{22}$ it is sub-divide into two actions::
where:
A_{21} = *Extrac Top*
A_{22} = *Deposit Top*

In this application the task is a Common Type Task *Tc.* The human and the robot work simultaneously on the same object, and the work area must be the intersection E_{12} as shown in Figure 2. The decisions trees where the actions are assigned result obviously, given the simplicity of the working cell, then result:

$R_1 \rightarrow A_1$ the first action is asigned to the robot PA-10.

$H_1 \rightarrow A_2$ the second action is asigned to the human.

The decision tree shown in Figure 5 is constructed for planning T_{s1}.

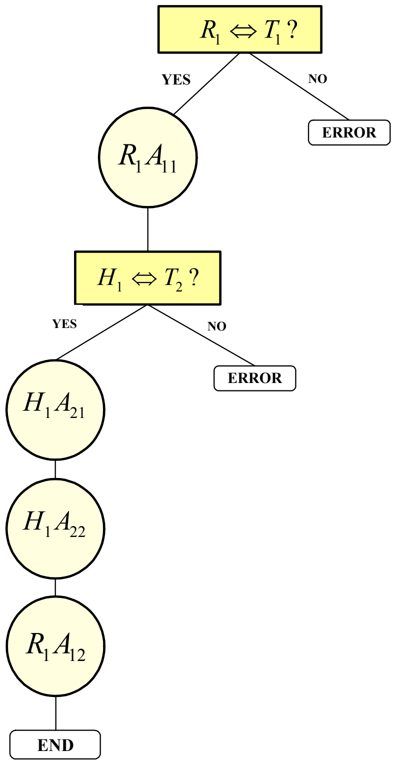


Figure 5: Decision Tree T_{s1}.

In Figure 5 it is observed that the tool availability is checked in each moment to make the system more reliable. There, actions cannot be executed in a parallel way because of the precedence between them or to avoid the human may suffer any physical damage. For example, action A_{12} which corresponds to deposit the ball, cannot be started until the action A_{22} (human extracting the top) has finished.

In Figure 6 the real sequence to execute T_{s1} is shown.

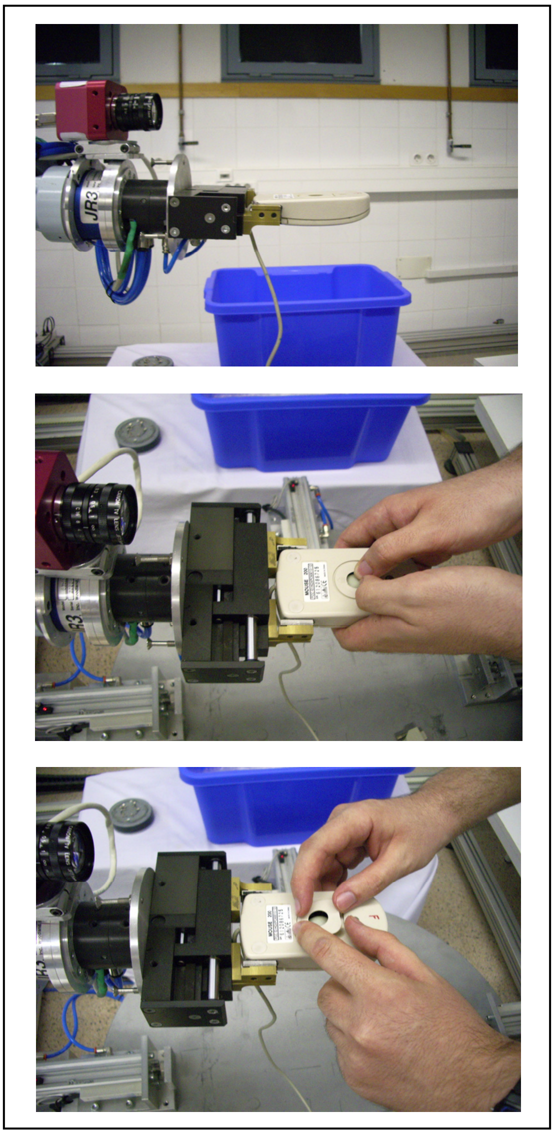


Figure 6: Sequence T_{s1}.

5 CONCLUSIONS

A cooperative Task Planner is set out. It allows developing cooperative task between manipulator and task in which robot and human interaction is needed inside a disassembly system. The main goal is to provide a safe and flexible cooperative system. This system could achieve greater productivity in the industry.

Robots are use to assist the human operators in some specific industrial tasks, reduce the fatigue, and increase the accuracy in areas in which only a human can bring global knowledge, experience, and comprehension in the executing of the task

It is observed, that the modifications made in the task planning block to adapt it to cooperative tasks between man-robot, are minimum. Therefore a future project work might consider extending the use of the Task Planner to other types of applications like services robots, where work between robots and humans has a great potential.

ACKNOWLEDGEMENTS

This work was funded by the Spanish MEC project "Diseño, implementación y experimentación de escenarios de manipulación inteligentes para aplicaciones de ensamblado y desensamblado automático" (DPI2005-06222), and by the Spanish G.V. project "Desensamblado automático cooperativo para el reciclado de productos" (GV05/003).

REFERENCES

Adams, J., Skubic, M., 2005. Introduction to the Special issue on Human-Robot Interaction. *IEEE Transaction on System, Man, and Cybernetics. Part A*. **Vol. 35, No. 4**, pp. 433-437

Burke, J., Murphy, R., Rogers E., Scholtz J., Lumelsky V., 2003. Final Report for the DARPA/NSF Interdisiplinary study on Human-Robot Interaction. *Transaction IEEE Systems, Man, and Cybernetics*. Part A, **Vol. 34**, No. 2

Corrales, J.A., Torres, F., Candelas, F., 2006. Tecnologías en la Inteligencia Ambiental. *XXVII Jornadas de Automática*. Almería. Spain.

Corke, P. I. 1999. Safety of advance robots in human enviroments. *Proceeding of the IARP 1999.*

Diaz, C., Puente, S.T., Torres, F. 2006. Task Planner for a Cooperative Disassembly Robotic System. *Proceeding of the 12th IFAC Symposium on Information Control Problems in Manufacturing*. Lyon. France. Vol. 1, pp. 163-168.

Ikuta, K., Nokata, M., 2003. Safety evaluation method of desing and control for human care robots. *The International Journal of Robotics Research.* **Vol. 22**, pp. 281-297.

Kulic, D., Croft, E., 2005. Safe Planning for Human-Robot Interaction. *Journal of Robotics System*. pp. 433-436

Navarro-Serment, L., Grabowski, R., Paredis, C., Khosla, P., 2002. Millibotspag. *IEEE Robotics & Automation Magazine*. **Vol 9, No. 4**, pp 31-40.

Torres, F., Puente, S.T., Aracil ,R. 2003. Disassembly planning based on precedence relations among assemblies. *The International Journal of Advanced Manufacturing Technology.***Vol. 21. No. 5**. pp. 317-327.

Torres, F., Puente, S.T., 2006. Editorial: Intelligent disassembly in the demanufacturing process. *The International Journal of Advanced Manufacturing Technology.***Vol. 30.** pp. 479-480.

ADAPTIVE CONTROL BY NEURO-FUZZY SYSTEM OF AN OMNI-DIRECTIONAL WHEELCHAIR USING A TOUCH PANEL AS HUMAN-FRIENDLY INTERFACE

Kazuhiko Terashima, Yoshiyuki Noda, Juan Urbano, Sou Kitamura and Takanori Miyoshi
Department of Production Systems Engineering, Toyohashi University of Technology
Hibarigaoka 1-1, Toyohashi, 441-8580, Japan
(terashima, juan, kitamura)@syscon.pse.tut.ac.jp

Hideo Kitagawa
Department of Electronic Control Engineering, Gifu National College of Technology
Kamimakuwa, Motosu, Gifu, 501-0495, Japan
kita@m.ieice.org

Keywords: Omni-directional wheelchair, power assistant, operability, human-machine interface.

Abstract: For improving the operability of an omni-directional wheelchair provided with a power assist system, the system must be able to adapt to the individual characteristics of the many different attendants that will use it. For achieving this purpose, an innovative human-interface using a touch panel that provides easy input and feedback information in real time of the operation of a power-assisted wheelchair was developed. The system was tested experimentally with many different attendants and the results show that in addition to providing a human friendly interface by using the touch panel system with monitor it can adapt successfully to the particular habits of the attendants.

1 INTRODUCTION

In order to satisfy the demand for higher mobility, designers have created new driving concepts such as omni-directional movement which allows any combination of forward, sideways, and rotational movement, thus ensuring users much more freedom and safety in wide or narrow spaces. A variety of wheelchairs with different options and special add-on features have been developed to meet a wide range of needs (Wada and Asada, 1999)-(West and Asada, 1992).

In the author's laboratory, a holonomic Omni-directional Wheelchair (OMW) which can act as an autonomous (Kitagawa et al., 2002) or semi-autonomous (Kitagawa et al., 2001) omni-directional wheelchair has been developed. Comfort has been a subject of study in the case with and without the joystick (Kitagawa et al., 2002), (Terashima et al., 2004).

For handicapped people or elderly people that can use their arms freely, many power assisted wheelchairs have been developed such as Seki (Seki et al., 2005) and Frank Mobility Systems (FrankMobilitySystems, 2002), for example. However, it is necessary to consider that some elderly people or handicapped people can not use their arms because they are damaged or they are so weak. These people needs the help of an attendant. Considering this background, a power assist system that helps attendants to move a heavy load has been designed and developed in author's laboratory (Kitagawa et al., 2004). Application of power assist for supporting the attendant of an omni-directional wheelchair is one of a novel research. To the authors knowledge, no other report about this topic has appeared yet. However, there is some research about power system for omni-directional vehicles, but it is related to *carts* (Maeda et al., 2000), *not to wheelchairs*. Moreover, it still has some problems in rotation and in occupant's comfort since this system was developed for a food tray carry vehicle in a hospital.

However, there is a problem related to the operability of the OMW. Due to the application of the power assist system, operability of the OMW degrades, especially when the attendant tries to rotate in clockwise (CW), or counter-clockwise (CCW) direction around the center of gravity (***CG***) of the OMW. This problem is generated from the fact that it is difficult to give the human force exactly towards the target direction by means of the handle attached to the wheelchair, hence the movement of the OMW using conventional power assist does not provide to the target exactly. Further-

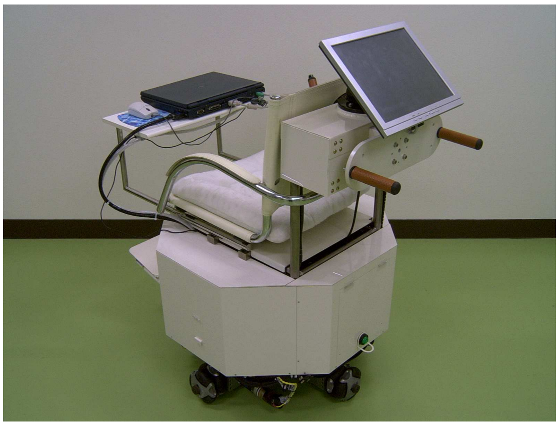


Figure 1: Omni-directional wheelchair (OMW).

more, the sensor position to measure the force added by human for power assist is different from the position of the gravity center of the OMW, and therefore the force generated by its difference must be compensated.

It was impossible to find general rules to solve the both problems stated in the above, but the relationship was found by authors between lateral and rotational movements. These relationships were used as the base for constructing a fuzzy reasoning system that helped to improve the operability of the OMW.

Nevertheless, when the system was tested by different attendants, it was found that a complete satisfactory result was not obtained by every attendant. It is because each person has its own tendencies and the fuzzy inference system must be tuned to respond to them. Tuning of the fuzzy inference system by trial and error thus has been tried by authors' group. However it is a time consuming and needs a lot of trials of the attendants, then these can become tired and bored.

Thus, a better tuning method, a method that allows tuning of the fuzzy inference system, is needed. It can be obtained by adding Neural Networks (NN) to the fuzzy inference system, obtaining what is known as a neuro-fuzzy system. There is a lot of research in this topic (Jang, 1993)-(Lin and Lee, 1991), being the basic difference the kind of NN that is used in combination with the fuzzy inference system.

Jang (Jang, 1993) developed ANFIS: Adaptive-Network-based Fuzzy Inference Systems, a neuro-fuzzy system in which the fuzzy inference system is tuned by using the input data of the system.

The desired direction of motion of the attendant as the teaching reference for the learning could be input by just using the keyboard of the computer. However keyboard input is not user-freindly. Furthermore, this method does not provide feedback information to the attendant in order to know how well he is accomplishing the desired motion. Then, a human interface that provides information to the attendant is desired. This can be achieved by using a touch panel system with monitor, which is a device that can be used as an input and at the same time can show the resultant motion of the OMW. The desired motion and the real motion of the OMW are compared in order to obtain the difference, or error, that will be used for the training of ANFIS, as explained in a previous paper (Terashima et al., 2006).

In a previous paper (Terashima et al., 2006) by the authors, the forwards-bacwards velocity was not included in the ANFIS system of the OMW and then a Reduction Multiplicative Factor (RMF) was used for the improvement of the rotational motion of the OMW when there was some interference of the forwards-backwards velocity. By using the RMF it was possible to achieve good operability in the forwards-backwards motion, lateral motion and rotational motion. However the results were not satisfactory in the case of slanting motion. By including the forwards-backwards velocity in the ANFIS system as shown in Fig. 6, and with the use of the touch panel for providing teaching reference for the learning, it was possible to acomplish a general omni-directional motion. Simulation and experimental results in the case of diagonal motion are shown in Fig. 13 and Fig. 14.

Hence, in this paper, an innovative method for improving the operability of a power assist omni-directional wheelchair by using a touch panel with neuro-fuzzy controller as a human interface is proposed.

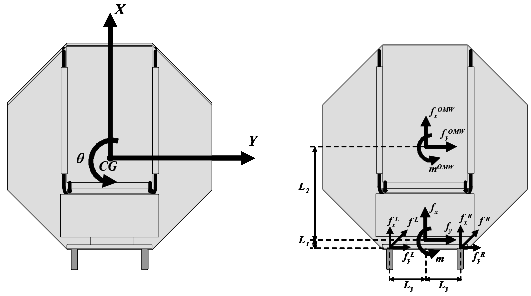
2 CONSTRUCTION OF OMW USING A TOUCH PANEL AS HUMAN INTERFACE

A holonomic omni-directional wheelchair (OMW) using omni-wheels has been built by authors' gropup, as is described in (Kitagawa et al., 2002)-(Kitagawa et al., 2001). Figure 1 shows an overview of the OMW developein by authors' group.

The OMW is able to move in any arbitrary direction without changing the direction of the wheels. In this system, four omni-directional wheels are individually and simply driven by four motors. Each wheel has passively driven free rollers at their circumference. The wheel that rolls perpendicularly to the direction of movement does not stop its movement because of the passively driven free rollers. These wheels thus allow movement that is holonomic and omni-directional.

The OMW is also equipped with a handle and a six-axis force sensor, as shown in Fig. 1, that allows the OMW's use in power-assist mode. The force that the

Figure 2: Touch panel used for the OMW.



(a) Coordinate system of the OMW. (b) Decomposition of the forces applied to the handles of the OMW.

Figure 4: Working force.

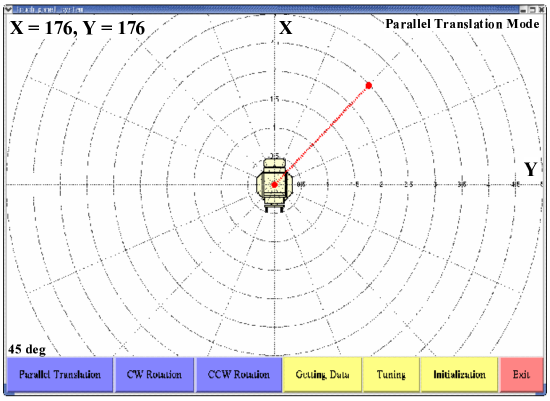


Figure 3: GUI developed for the touch panel.

Table 1: Fuzzy reasoning rules for lateral motion and rotational motion.

R	Antecedent		Consequent
1	If	$Vy < 0$ and $\omega < 0$,	then $\omega < 0$
2	If	$Vy \approx 0$ and $\omega < 0$,	then $\omega < 0$
3	If	$Vy > 0$ and $\omega < 0$,	then $Vy > 0$
4	If	$Vy < 0$ and $\omega \approx 0$,	then $Vy < 0$
5	If	$Vy \approx 0$ and $\omega \approx 0$,	then 0
6	If	$Vy > 0$ and $\omega \approx 0$,	then $Vy > 0$
7	If	$Vy < 0$ and $\omega > 0$,	then $Vy < 0$
8	If	$Vy \approx 0$ and $\omega > 0$,	then $\omega > 0$
9	If	$Vy > 0$ and $\omega > 0$,	then $\omega > 0$

attendant inputs to the grips of the handle is measured by this force sensor. Second order lag filter is used for the transformation from force to velocity (Terashima et al., 2006).

A touch panel is a display device that accepts user input by means of a touch sensitive screen. Because of their compact nature and ease-of-use, touch panels are typically deployed for user interfaces in automation systems, such as high-end residential and industrial control. Touch panels are also becoming common on portable computers such as Tablet PCs, Ultra-Mobile PCs and consumer devices such as VOIP phones. In this research, a touch panel as shown in Fig. 2 is used as an input device in which the attendant of the OMW draws the desired direction of motion. As shown in Fig. 2, the touch panel is mounted in the rear part of the OMW such as the attendant can reach to it easily. The touch panel used in this research is a TFT Touch Monitor HV-141T produced by ULTEC Corporation, Japan. A GUI (Graphical User Interface) was developed for making easy the interaction with the attendant, as show in Fig. 3. In this GUI the attendant can draw any kind of motion, be it an slanting motion, or a rotational movement.

3 NEURO-FUZZY SYSTEM FOR IMPROVING OPERABILITY

When the user tries to rotate OMW around its gravity center, OMW begins to slide and the radius of rotation sometimes becomes very big. Then, rotation around the center is very difficult (Kitagawa et al., 2004). A survey was conducted among various attendants trying to discover some relationships in the way they realized forwards-backwards, lateral and rotational movements. The goal of the survey was to find general rules that related the three mentioned motions. Even when it was impossible to find general rules that explained all cases, a relationship was found between lateral and rotational movements by authors. These relationships were used as the base for constructing a fuzzy reasoning system (MathWorks, 2002)-(Harris et al., 1993) that helped to improve the operability of the OMW. In order to establish the rules of direction inference, first, the force applied to the grips of the force sensor are changed to the center of OMW, as shown in Fig. 4. It is easy to derive a ba-

Table 2: Fuzzy rules for the change of Vx in order to improve operability.

R	Antecedent	Consequent
1	If $Vx < 0$ and $Vy < 0$,	then $Vx < 0$
2	If $Vx \approx 0$ and $Vy < 0$,	then $Vx \approx 0$
3	If $Vx > 0$ and $Vy < 0$,	then $Vx > 0$
4	If $Vx < 0$ and $Vy \approx 0$,	then $Vx < 0$
5	If $Vx \approx 0$ and $Vy \approx 0$,	then 0
6	If $Vx > 0$ and $Vy \approx 0$,	then $Vx > 0$
7	If $Vx < 0$ and $Vy > 0$,	then $Vx < 0$
8	If $Vx \approx 0$ and $Vy > 0$,	then $Vx \approx 0$
9	If $Vx > 0$ and $Vy > 0$,	then $Vx > 0$

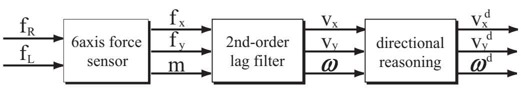


Figure 5: Block diagram of the power assist system.

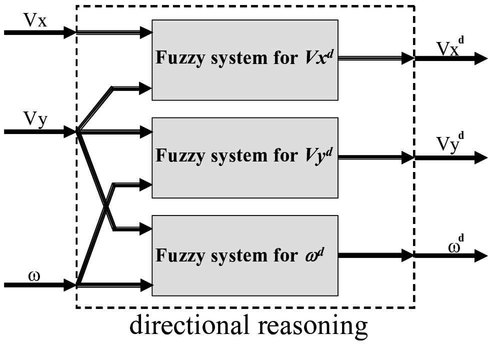


Figure 6: Contents of the block "directional reasoning".

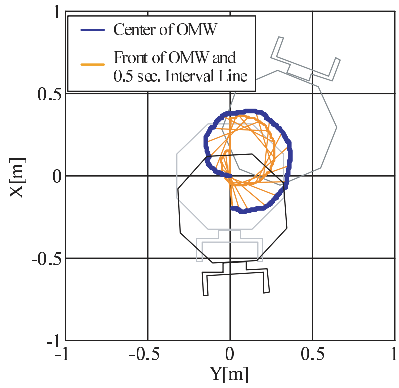


Figure 7: Results when fuzzy reasoning is not applied for improving operability.

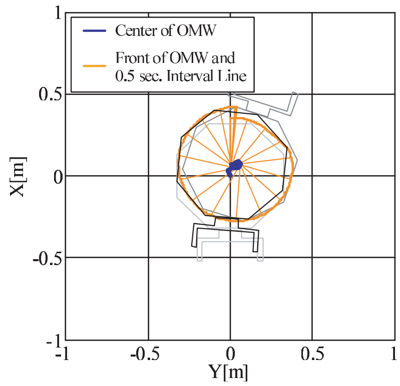


Figure 8: Results when fuzzy reasoning is used by "**Attendant 1**".

sic equation to compensate the difference between the sensor and the actuators allocation (Kitagawa et al., 2004). However, it is difficult to exactly give the force for the sensor to the target direction. Further, how to give the force for the gripper sensor is slightly different depending on persons even for the same target motion of the OMW. The rules of direction inference, in which just lateral motion and rotational motion are considered, are shown in Table 1. In Table 1, Vy represents the lateral velocity of the OMW, and ω represents the angular velocity of the OMW. The forwards and backwards velocity of the OMW is given by Vx.

The system in which fuzzy reasoning was applied just to the lateral and rotational velocity was tested, and it was found that even when the operability in lateral direction was improved, there were still some problems with the rotational movement because of a component Vx that did not allowed to achieved a perfect rotation over the center of gravity of the OMW. A Reduction Multiplicative Factor (RMF) which decreases the value of Vx in the case of rotational motion, and keeps it unchanged in the case of forwards-backwards movement was the solution provided by authors in previous research (Terashima et al., 2006). By using the RMF it was possible to improve the forwards-backwards motion, lateral motion and rotational motion over the gravity center of the OMW.

However, as Vy was subjected to fuzzy reasoning and Vx was not, it was not possible to achieve good operability for slanting motions, like diagonal motion. In the case of diagonal motion, for example, the attendant tries to move the OMW in such a way that the inputs of Vx and Vy are almost the same in the beginnig. Nevertheless, as Vy is subjected to directional reasoning, its value changes. Vx is not subjected to directional reasoning, then its value remains always the same. As a consequence, it is not possible to achieve good operability in diagonal motion.

For solving this problem, Vx was subjected to directional reasoning too using the fuzzy rules shown in Table 2. This rules make it possible to include Vx in the fuzzy reasoning system without disturbing the

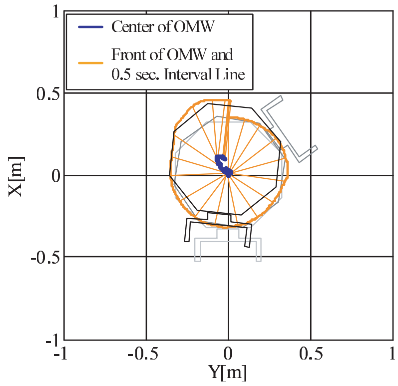


Figure 9: Results when fuzzy reasoning is used by "**Attendant 2**".

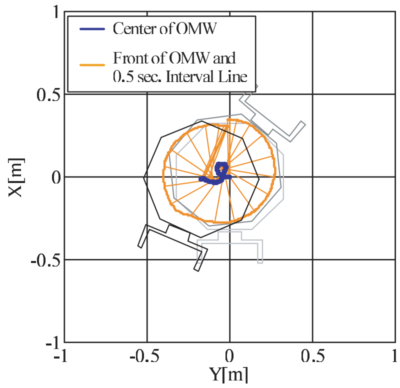


Figure 10: Results when fuzzy reasoning is used by "**Attendant 3**".

values of Vy or ω. The block diagram of the system that considers power assist and fuzzy reasoning is shown in Fig. 5, and the contents of the block labeled as "directional reasoning" are shown in Fig. 6. By including Vx in the ANFIS system it was possible to acomplish a general omni-directional motion.

Fig. 7 shows the results in the case of a counter-clockwise rotational over the center of gravity of the OMW when no fuzzy reasoning is used. It is possible to see that there is a deviation in the lateral direction as well as in the forwards-backwards direction. For solving this problem, the fuzzy system was used. It was tuned by trial and error, as explained in (Kitagawa et al., 2004), for an attendant that will be called "Attendant 1", and the results, presented in Fig. 8 shows that the rotational movement was improved considerably. However, when the same system was tested with two more different attendants, called "Attendant 2" and "Attendant 3", the results were not as good as in the case of "Attendant 1", as shown in Fig. 9 and Fig. 10. It means that the system must be tuned in order to respond to the individual characteristics of the different attendants. However, the tuning by trial and error is time consuming and boring for the attendant. For that reason, the automatic tuning of the system by using a neuro-fuzzy system, ANFIS (Adaptive-Neural Fuzzy Inference System) was proposed and developed as described in (Terashima et al., 2006). The ANFIS system of the OMW provided in this paper is shown in Fig. 11.

4 ADAPTIVE CONTROL WITH HUMAN INTERFACE AND RESULTS

In previous research (Terashima et al., 2006), the desired direction of motion of the attendant was input by using the keyboard of the computer of the OMW. However, the attendant could not get a clear idea of the direction in which he wanted to move, neither verify if the real motion of the OMW really corresponded to his desire. In order to provide the attendant with an easy way for inputing the desired direction of motion and for verifying the direction of motion, a human interface consisting of touch panel, as shown in Fig. 2 is used. A GUI (Graphical User Interface) was developed for making easy the interaction with the attendant, as shown in Fig. 3. In this GUI the attendant can draw any kind of motion, like, for example, an slanting motion, or a rotational movement. Moreover, it allows the attendat to follow the motion of the OMW in the screen of the touch panel, and compare the difference between the desired motion and the real motion of the OMW. The complete system, when the touch panel is included, is shown in Fig. 12.

The procedure for applying the touch panel is as follows:

1. First, the attendant draws in the touch panel the kind of movement that he desires to accomplish, as teaching signal for the learning of Neural Networks.
2. Then, the attendant moves the OMW trying to accomplish the desired motion.
3. However, in the general case, there as a difference between the desired motion and the real motion. This difference is used for the training of the ANFIS system of the OMW, as explained in (Terashima et al., 2006).

This system was used for supporting the operation of the attendant in many kinds of movements. Like for example forwards-backwards motion, lateral motion, rotational over the gravity center of the OMW in clockwise and counter-clockwise direction, and many cases of slanting motion. In Fig. 13 it is possible to observe the simulation results of one attendant for the case of diagonal movement to the upper right corner of the XY system shown. Fig. 13 (a) shows the diagonal trajectory obtained before tuning is accomplished. It is possible to see that it is more an arc than an straight diagonal line. By using the same input data used in Fig. 13 (a), the system is tuned by using ANFIS, and the trajectory obtained after the tuning is shown in Fig. 13 (b). It can be observed that the trajectory has been improved, as expected. The number of data used for the training of the ANFIS was in the range of 3500 $\sim$ 4000 data, and the learning time was around 30 [s] $\sim$ 40 [s] in a Pentimum III 1 [GHz] personal computer. The system was tested by experiment, for the same attendant, with the results shown in Fig. 14 (a) for the case before tuning, and Fig. 14 (b) for the case after tuning. As in the case of the simulation, the trajectory obtained in the experiments is not so good before tuning, but it was improved after the tuning of the ANFIS system of the OMW.

5 CONCLUSIONS

An innovative human-interface using a touch panel that provides easy input and feedback information in real time of the operation of a power-assisted wheelchair was developed. Furthermore, adaptive control using a neuro-fuzzy system was proposed in a human friendly fashion by means of a touch panel as a human-interface for improving the operability of the wheelchair. The system was tested by simulation and experiments, and its efectiveness was demonstrated.

ACKNOWLEDGEMENTS

This work was partially supported by The 21st Century COE (Center of Excellence) Program ”Intelligent Human Sensing”

REFERENCES

FrankMobilitySystems (2002). Frank mobility systems inc. In *http://www.frankmobility.com/*.

Harris, C. J. et al. (1993). Intelligent control. World Scientific.

Jang, J. (1993). Anfis: Adaptive-network-based fuzzy inference system. In *IEEE Transactions on Systems, Man, and Cybernetics, Vol. 23(3), pp. 665-685*.

Kitagawa, H. et al. (2001). Semi-autonomous obstacle avoidance of omnidirectional wheelchair by joystick impedance control. In *Proc. IEEE/RSJ Int. Symp. on Intelligent Robots and Systems, pp. 2148-2153*.

Kitagawa, H. et al. (2002). Motion control of omnidirectional wheelchair considering patient comfort. In *Proc. IFAC World Congress, T-Tu-E20*.

Kitagawa, H. et al. (2004). Fuzzy power assist control system for omni-directional transport wheelchair. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and SystemsC1580-1585*.

Lin, C. T. and Lee, C. S. G. (1991). Neural-network-based fuzzy logic control and decision system. In *IEEE Trans. on Computers, Vol. 40(12), pp. 1321-1336*.

Maeda, H. et al. (2000). Development of omni-directional cart with power assist system (in japanese). In *Proc. 18th Annual Conf. of Robotics Society of Japan, 15, pp.1155-1156*.

MathWorks (2002). Fuzzy logic toolbox user's guide version 2, pp. 3-18. The Matworks Inc.

Seki, H. et al. (2005). Novel driving control of power assisted wheelchair based on minimum jerk trajectory. In *IEEJ Trans. EIS Vol. 125(7) (in Japanese), pp. 1133 - 1139*.

Terashima, K. et al. (2004). Frequency shape control of omni-directional wheelchair to increase user's comfort. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation (ICRA), pp. 3119-3124*.

Terashima, K. et al. (2006). Enhancement of maneuverability of a power assist omni-directional wheelchair by application of neuro-fuzzy control. In *Proceedings of the 3nd International Conference on Informatics in Control Robotics and Automation (ICINCO 2006), pp. 67-75*.

Wada, M. and Asada, H. (1999). Design and control of a variable footprint mechanism for holonomic omnidirectional vehicles and its application to wheelchairs. In *Proc. IEEE Trans. Robot. Automat, 15, pp. 978-989*.

West, M. and Asada, H. (1992). Design of a holonomic omnidirectional vehicle. In *Proc. IEEE Int. Conf. Robot. Automat., pp. 97-103*.

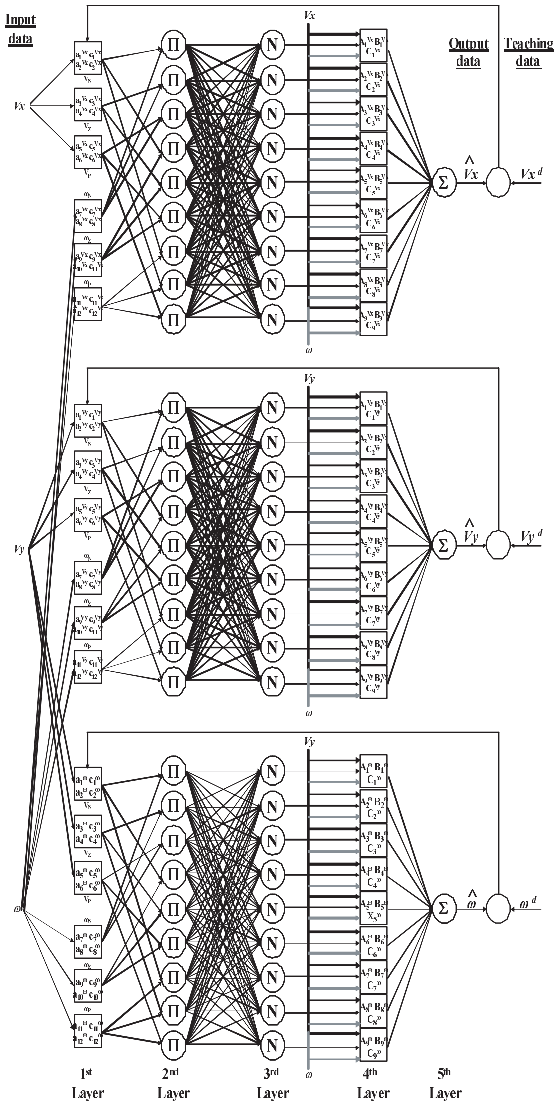


Figure 11: ANFIS systems of the OMW.

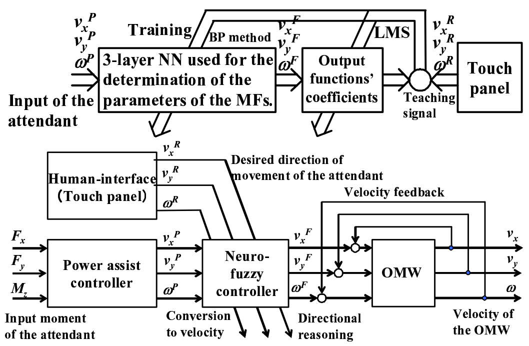


Figure 12: Complete system when the touch panel is included.

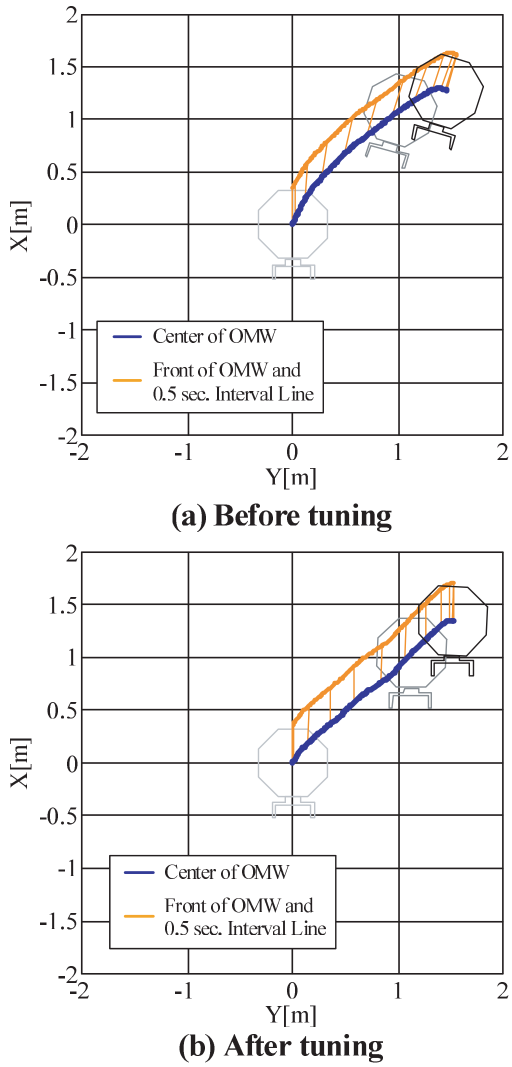


Figure 13: Simulation results for one attendant in the case of diagonal movemement to the right.

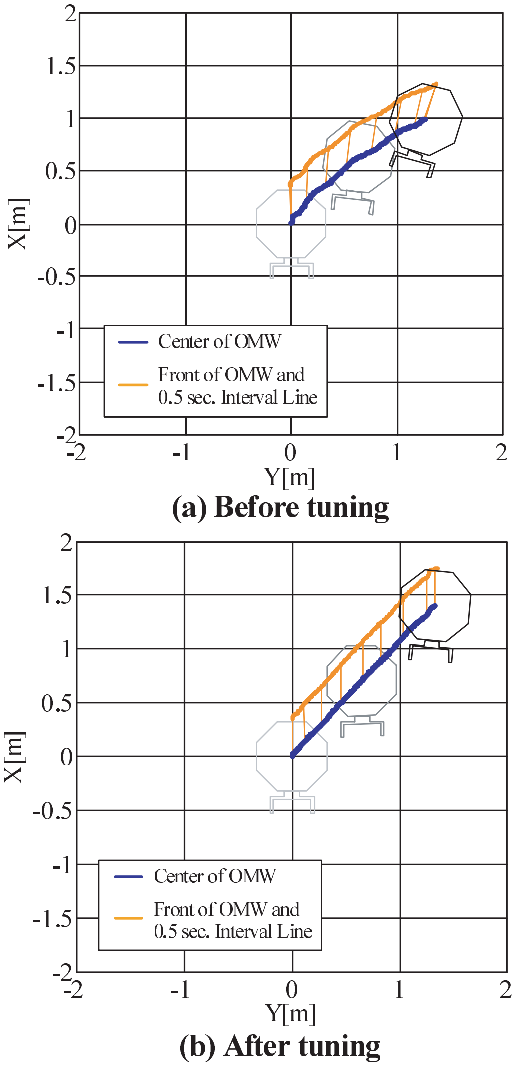


Figure 14: Experimental results for one attendant in the case of diagonal movemement to the right.

HUMAN-SCALE VIRTUAL REALITY CATCHING ROBOT SIMULATION

Ludovic Hamon, François-Xavier Inglese and Paul Richard
Laboratoire d'Ingénierie des Systèmes Automatisés, Université d'Angers
62 Avenue Notre Dame du Lac, 49000 Angers, France
ludovic.hamon@univ-angers.fr, inglese@istia.univ-angers.fr, paul.richard@univ-angers.fr

Keywords: Virtual reality, large-scale virtual environment, human-robot interaction, catching.

Abstract: This paper presents a human-scale virtual reality catching robot simulation. The virtual robot catches a ball that users throw in its workspace. User interacts with the virtual robot using a large-scale bimanual haptic interface. This interface is used to track user's hands movements and to display weight and inertia of the virtual balls. Stereoscopic viewing, haptic and auditory feedbacks are provided to improve user's immersion and simulation realisms.

1 INTRODUCTION

Robotic researchers tried to solve the problem of catching moving object (dynamic problem) while basing themselves on the use a priori of the trajectory of the object to limit the calculating time.

Most of the proposed methods generally rest on the following stages:

1) the detection of the ball,
2) the determination since it is in flight,
3) the follow-up and the prediction of its trajectory
4) the economic planning and the execution of a movement of interception.

Indeed, the prediction of balls trajectories in a controlled environment (no wind, etc.) is based on a priori knowledge of characteristics of this type of movement and, on the collection of information about the displacement of the ball, before beginning to make a prediction on the trajectory followed by the object.

Virtual Reality (VR) is a computer-generated immersive environment with which users have real-time interactions that may involve visual feedback, 3D sound, haptic feedback, and even smell and taste (Burdea, 1996 ; Richard, 1999 ; Bohm, 1992 ; Chapin, 1992 ; Burdea, 1993 ; Sundgren, 1992 ; Papin, 2003). By providing both multi-modal interaction techniques and multi-sensorial immersion, VR presents an exciting tool for simulation of (real) human – (virtual) robot interaction or cooperation. However, this requires a large-scale Virtual Environments (VEs) that provide efficient and multi-modal interaction techniques including multi-sensorial feedbacks.

2 HUMAN-SCALE VE

Our multi-modal VE is based on the SPIDAR interface (Figure 1). In this system, a total of 8 motors for both hands are placed surrounding the user (Sato, 2001). Motors mounted near the screen and behind the user; drive the strings attachments. One end of string attachment is wrapped around a pulley driven by a DC motor and the other is connected to the user's hand.

By controlling the tension and length of each string, the SPIDAR-H generates an appropriate force using four string attachments connected to a hand attachment. Since it is a string-based system, it has a transparent property so that the user can easily see the virtual world.

It also provides a space where the user can freely move around. The string attachments are soft, so there is no risk of the user hurting himself if he would get entangled in the strings. This human-scale haptic device allows the user to manipulate virtual objects and to naturally convey object physical properties to the user's body. Stereoscopic images are displayed on a back-projected large screen (2m x 2,5m) and viewed using polarized glasses. A 5.1 immersive sound system is used for simulation realism, auditory feedback and sensorial immersion. Olfactory information can be provided using a battery of olfactory displays.

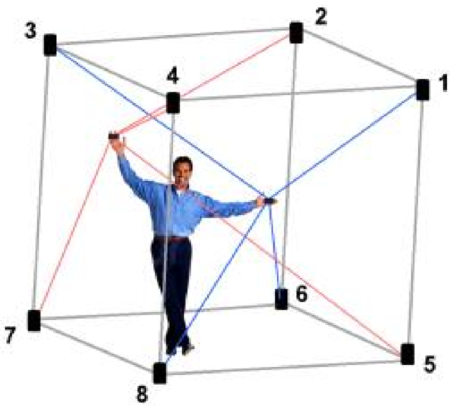


Figure 1: Workspace of the SPIDAR device.

3 CATCHING SIMULATION

3.1 Virtual Room

The virtual room in which simulation takes place is a right-angled parallelepiped which consists of a ground, a left wall and a right wall. The ceiling is left open. A wood texture was added on each face to increase the depth perception, as well as the ball shadow.

This virtual room contains objects such as a virtual ball, virtual hands (right and left), and a virtual robot (a Kuka KR6).

All calculation are made in cartesian co-ordinates X, Y, Z, according to an orthonormed reference frame whose origin O is located at the middle of the floor. The Z axis is directed towards the user. The Y axis is directed upwards. The X axis is directed towards the right compared to the user view.

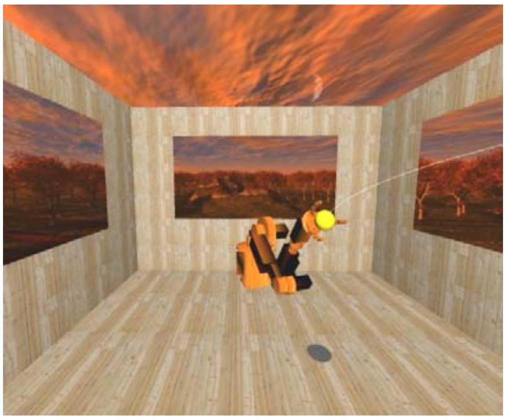


Figure 2: Snapshot of the robot reaching for the ball.

3.2 Robot Modelling

The robot chosen here is Kuka KR6 model. It is an arm manipulator with 6 degrees of freedom, having only rotoids axes. It is placed at the bottom of the virtual room. Each part of the model was modelled in Discreet 3D Studio Max 7.0 and then imported into OpenGL. The robot consists of 6 rotoïds axes whose angles are respectively q1, q2, q3, q4, q5, q6.

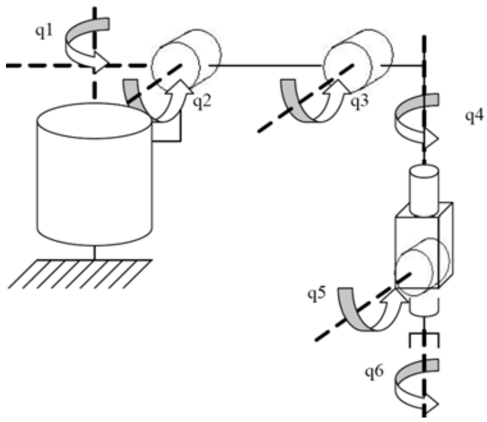


Figure 3: Illustration of the parameters used for the geometrical modelling of the Kuka KR6 robot.

To animate each robot part, elementary geometrical operations such as translations and rotations around the frame reference is used.

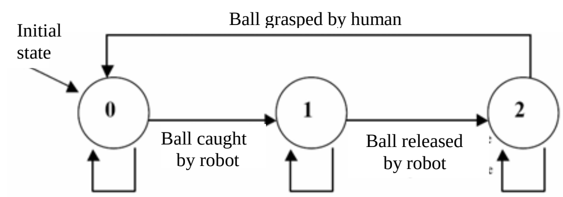


Figure 4: Finite state machine of the robot.

The virtual robot is subjected to the finite state machine given in figure 4. The various states are defined as follows:

State 0: the robot tries to catch the ball if in its workspace.

At the beginning of simulation, the robot waits until the ball is seized by the human operator, via the virtual hand, or till an external force is generated on the ball, to return to state 0.

State 1: the robot catches the ball if in state 0.

State 2: the robot releases the ball automatically, after a certain amount of time, and returns in its initial configuration.

The robot waits until the ball is grasped by the user (using one of the virtual hand) or till an external force is generated on the ball to return to state 0. Once the ball is caught, the robot automatically drops the ball and the simulation is reinitialised in its initial configuration.

The virtual ball is represented by a sphere and has a given mass "m", a given radius "R" and a given velocity "Vb" (or a Velocity Vector).

Assimilated to a single point which is the centre of the sphere, the ball is animated according to the fundamental law of dynamics: F=mA, i.e. the sum of the external forces F applied to the ball, is equal to the mass of the ball multiplied by acceleration.

Thus, the animation engine of the ball uses the following formulas:

Force = truncate(Force, max_force)
Acceleration = Force/m
Velocity = Velocity + Acceleration
Velocity = truncate(Velocity, max_velocity)
Position = Position + Velocity
Or Force=(Fx,Fy,Fz) , Acceleration=(Ax,Ay,Az) ,
Velocity = (Vx , Vy , Vz) ,
Position (Px , Py , Pz)

"max_force" is defined by the developer. It represents the maximum force that could be applied to the ball. Similarly, "max_velocity" represents the maximum velocity that could be set to the ball. Thus one truncates the force by "max_force" and velocity by "max_velocity" to avoid reaching a force or velocity of oversized magnitudes.

In this way, a new position of the ball could be calculated at any time (more precisely according to the main loop of the simulation), when the ball is free (not caught by the robot or grasped by the user). The ball is subjected to the finite state machine given in fig.5.

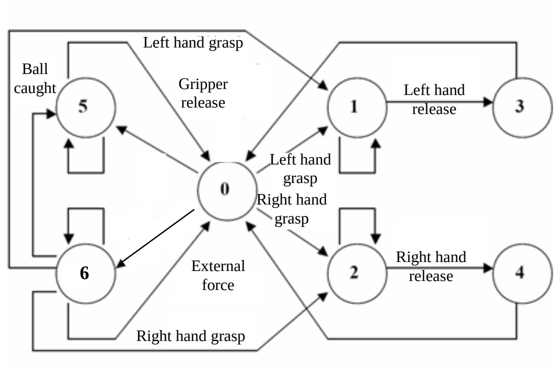


Figure 5: Finite state machine of the ball.

The various states are defined as follows:

State 0: the ball is free and is thus subjected to the animation engine described before.

State 1: the ball is caught by the left hand. The position of the ball is therefore directly linked to the position of this hand.

State 2: the ball is caught by the right hand. The position of the ball is therefore directly linked to the position of this hand.

State 3: the ball is released by the left hand. The position of the ball is no more established by the hand, but rather by the animation engine. The external Forces vector is equal, at this moment, to the hand velocity vector Vmx, Vmy, Vmz.

State 4: the ball is released by the right hand. The position of the ball is no more established by the hand, but rather by the animation engine. The external Forces vector is equal, at this moment, to the hand velocity vector Vmx, Vmy, Vmz.

State 5: the ball is caught by the robot. The position of the ball is no more established by the animation engine, but rather is a function of the robot gripper position.

State 6: the ball lies on the ground or closed to the ground. The Velocity vector magnitude is close to zero. The ball automatically moves to state 6, which is the end state and is immobilized on the ground.

User's hands position is tracked using the SPIDAR device.
A gain parameter between the user hand movements and the virtual hands can be introduced in order to enable him to increase his workspace. For example, it can be tuned so that the user can reach any location of the virtual room without moving too far from the centre of the SPIDAR frame of reference.

The closing of the virtual hand is carried out by the closing of a 5dt wireless data glove worn by the user (http://www.5dt.com). This could also be achieved using wireless mousses integrated to the SPIDAR device.

Each virtual hand is subjected to the finite state machine given in fig.6. The different states are defined as follows:

State 0: the left (respectively right) hand is open: it cannot grasp the ball.
State 1: the left (respectively right) hand is closed: it can grasp the ball if the latter is in state 0 or 6 or 1 (respectively 2).

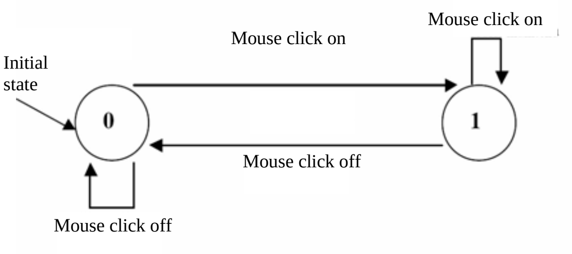


Figure 6: Finite state machine for both hands.

To do the ball grasping, a sphere of detection is used. Its size is defined by the designer and it is invisible during simulation. If the ball and the sphere are in contact, it is considered that the ball is seized, and the position of the ball is readjusted according to the hand.

3.3 Ball Launching

The virtual ball is thrown by the human operator, which can grasp and move it using the virtual hands. Once the ball is grasped, a method to launch the ball, corresponding to the animation engine, is proposed and validated. This method allows efficient velocity transfer of a user hand to the virtual ball.

To do this, hand velocity must be calculated. Thus an array of size S (S being defined by the designer), is created and is used to record the hand position at each loop cycle of the main program loop.

Fig. 7 illustrates an example with an array of size S = 4. At the initialisation, the array is empty.

This method is easy to implement and is not CPU-time consuming. It gives good results to reproduce realistic "launched balls". However, this requires an optimisation of the size (S) of the array.

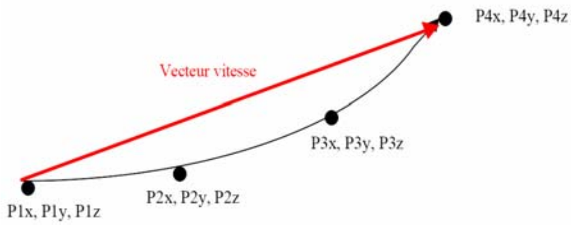


Figure 7: Illustration of the method proposed to efficiently launch the virtual ball with an array size S=4.

3.4 Ball Catching

Ball catching is achieved using a detection sphere of predefined size and invisible during the simulation. If the ball and the sphere are in contact, it is considered that the ball is caught. Then the ball position is readjusted according to the robot gripper position.

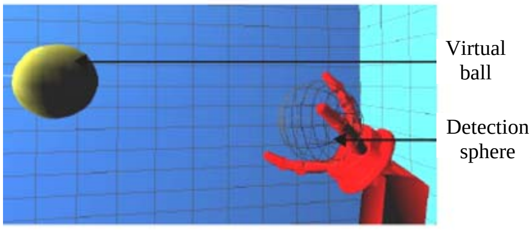


Figure 8: Illustration of the algorithm used for ball catching by the robot gripper.

This requires knowing both the cartesian position of the gripper according to the 6 angles q1, q2, q3, q4, q5, q6 and the dimensions of each part of the robot. The gripper is subjected to the finite state machine illustrated on fig. 9.

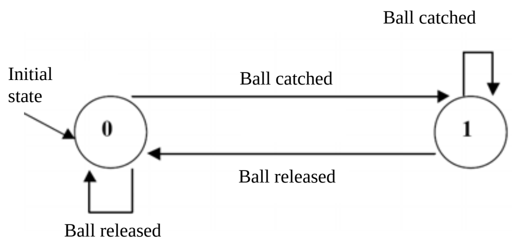


Figure 9: Finite state machine for the gripper.

The states of the gripper are defined as follows:

State 0: the gripper is open; the grip is open when the ball is not caught.

State 1: the gripper is closed; the grip is closed when the ball is caught.

In order for the robot to catch the ball, it is necessary to know: (1) the cartesian position of the gripper at any time according to the 6 angles q1, q2, q3, q4, q5, and q6 of the robot and, (2) the Cartesian space reached by the robot (workspace). This is given by the direct geometrical model defined by X=f(Q), with X=(x,y,z) and Q=(q1,q2,q3,q4,q5,q6).

It is also necessary to know the value of the 6 angles of the robot, according to the Cartesian position of the gripper (X, Y, Z). The inverse geometrical model can obtain these.

Thus, it can be resolved by determining the joint coordinates Q making it possible to obtain a desired location for the gripper specified by the space coordinates.

Here, we are confronted with a system of 3 equations with 6 unknown variables. To solve this system, the method proposed by Paul (1981) was used. This method allows obtaining the whole solutions set, when they exist.

In our simulation, the robot always faces the ball.
However, it will carry out a catching movement towards the ball only if the latter is in its workspace, defined by the whole set of points in the Cartesian space that the robot gripper can reach.

Under the hypotheses that the robot can reach all the points of its workspace at any time, and that there is no constraint on the rotation angles of the joint, the workspace of the robot is a TORE defined by equation 3.

$$\left(\sqrt{(x^2+z^2)} - A\right)^2 + y^2 = R^2 \qquad (3)$$

Figure 10: Snapshot of the robot oriented towards the ball.

Figure 11: Snapshot of the robot realising the ball.

4 CONCLUSION

We presented a human-scale virtual reality catching robot simulation.. The user interacts with a virtual robot by throwing virtual balls towards it, using a large-scale bimanual haptic interface. The interface is used to track user’s hands movements and to display various aspects of force feedback associated mainly with contact, weight, and inertia. We presented the robot modelling, as well as the ball launching and catching procedures.

REFERENCES

Inglese, F.-X., Lucidarme, Ph., Richard, P., Ferrier, J.-L., 2005. PREVISE : A Human-Scale Virtual

Environment with Haptic Feedback. In *Proceedings of ICINCO 2005*. Barcelona, Spain, pp. 140-145.

Burdea, G. , Coiffet, Ph.., Richard, P., 1996. Integration of multi-modal I/Os for Virtual Environments. In *International Journal of Human-Computer Interaction (IJHCI), Special Issue on Human-Virtual Environment Interaction*. March, (1), pp. 5-24.

Richard, P., Coiffet, Ph., 1999. Dextrous haptic interaction in Virtual Environments: human performance evaluation. In *Proceedings of the 8th IEEE International Workshop on Robot and Human Interaction*. October 27-29, Pisa, Italy, pp. 315-320.

Bohm, K., Hubner, K., Vaanaen, W., 1992. GIVEN: Gesture driven Interactions in Virtual Environments. A Toolkit Approach to 3D Interactions. In *Proceedings of Interfaces to Real and Virtual Worlds*. Montpellier, France, March, pp. 243-254.

Chapin, W., Foster, S., 1992. Virtual Environment Display for a 3D Audio Room Simulation. *In Proceedings of SPIE Stereoscopic Display and Applications*. Vol.12.

Burdea, G., Gomez, D., Langrana, N., 1993. Distributed Virtual Force Feedback. In *Proceedings of IEEE Workshop on Force Display in Virtual Environments and its Application to Robotic Teleoperation*. Atlanta, May 2.

Sundgren, H., Winquist, F., Lundstrom, I., 1992. Artificial Olfactory System Based on Field Effect Devices. In *Proceedings of Interfaces to Real and Virtual World.* Montpellier, France, pp. 463-472, March.

Papin, J.-P., Bouallagui, M., Ouali, A., Richard, P., Tijou, A., Poisson, P., Bartoli, W., 2003. DIODE: Smell-diffusion in real and virtual environments. In *Proceedings of the 5th International Conference on Virtual Reality.* Laval, France, pp.113-117, May 14-17.

Bowman, D.A., Kruijff, E., LaViola, J.J., Poupyrev, I., 2004. *3D User Interfaces: Theory and Practice.* Addison Wesley / Pearson Education.

Richard, P., Birebent, G., Burdea, G., Gomez, D., Langrana, N., Coiffet, Ph., 1996. Effect of frame rate and force feedback on virtual objects manipulation. In *Presence - Teleoperators and Virtual Environments*. MIT Press, 15, pp. 95-108.

Bouguila, L., Ishii, M., Sato, M. , 2000. A Large Workspace Haptic Device For Human-Scale Virtual Environments. In *Proceedings of the Workshop on Haptic Human-Computer Interaction*. Scotland.

Sato, M., 2001. Evolution of SPIDAR, In Proceedings of the 3rd International Virtual Reality Conference. Laval, May, France.

Richard, P., 1981. *Robot Manipulators--Mathematics, Programming, and Control*, MIT Press.

A DUAL MODE ADAPTIVE ROBUST CONTROLLER FOR DIFFERENTIAL DRIVE TWO ACTUATED WHEELED MOBILE ROBOT

Samaherni M. Dias, Aldayr D. Araujo
Department of Electrical Engineering, Federal University of Rio Grande do Norte, Brazil
sama@dca.ufrn.br, aldayr@dca.ufrn.br

Pablo J. Alsina
Department of Computer Engineering, Federal University of Rio Grande do Norte, Brazil
pablo@dca.ufrn.br

Keywords: Variable Structure Systems, Adaptive Control, Decoupling, Nonholonomic Mobile Robot.

Abstract: This paper is addressed to dynamic control problem of nonholonomic differential wheeled mobile robot. It presents a dynamic controller to mobile robot, which requires only information of the robot configuration, that are collected by an absolute positioning system. The control strategy developed uses a linear representation of mobile robot dynamic model. This linear model is decoupled into two single-input single-output systems, one to linear displacement and one to angle of robot. For each resulting system is designed a dual-mode adaptive robust controller, which uses as inputs the reference signals calculated by a kinematic controller. Finally, simulation results are included to illustrate the performance of the closed loop system.

1 INTRODUCTION

The control of mobile robot is a well known problem with nonholonomic constraints. There are many works about it, and several of these works use kinematic model. Dynamic model, which is composed by kinematic model plus the dynamic of robot, is used in a few works.

An adaptive controller that compensates for camera and mechanical uncertain parameters and ensures global asymptotic position/orientation tracking was presented by Dixon (Dixon et al., 2001). Chang (Chang et al., 2004) proposes a novel way to design and analysis nonlinear controllers to deal with the tracking problem of a wheeled mobile robots with nonholonomic constraints.

The exact mobile robot model is complex and the model has a lot of parameters to be considered, so we will use a simplified model that consider some of them. But this isn't the only problem, because some parameters can suffer variations, as the mass and the frictional coefficients. For example, when robot is doing a task as to carry an object, the object mass isn't considered by the model. So, we are using an association between a dual mode adaptive robust controller (DMARC) and a kinematic controller to the robot linear model.

The DMARC controller was presented in (Cunha and Araujo, 2004) and it proposes a connection between a variable structure model reference adaptive controller (VS-MRAC, proposed by Hsu and Costa(Hsu and Costa, 1989)) and a conventional model reference adaptive controller (MRAC). The goal is to have a robust system with fast response and small oscillations (characteristics of VS-MRAC), and a smooth steady-state control signal (characteristics of the MRAC).

2 NONHOLONOMIC MOBILE ROBOT

The robot considered in this paper is a nonholonomic direct differential-drive two-actuated-wheeled mobile robot, which is showed in Fig. 1. The robot is of symmetric shape and the center of mass is at the geometric center C of the body. It has two driving wheels fixed to the axis that passes through C. Each wheel is controlled by one independent motor.

Based on Fig 1, we have that d is distance between wheels, $r_{d,e}$ are right and left wheels radius, $\omega_{d,e}$ are right and left wheels angle velocity, ω_r is the angle robot velocity, $v_{d,e}$ are right and left wheels linear ve-

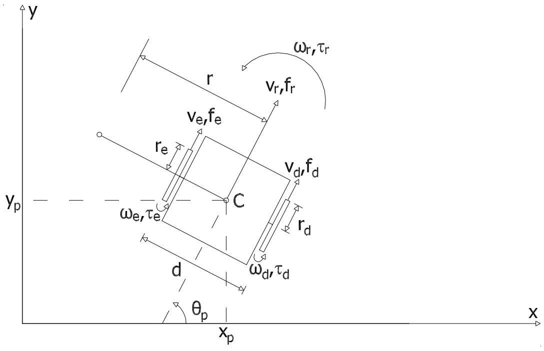


Figure 1: Differential-drive two-actuated-wheeled mobile robot.

locity, v_r is the linear robot velocity, $\tau_{d,e}$ are right and left wheels torque, τ_r is the robot torque, $f_{d,e}$ are right and left wheels force, f_r is the robot force, θ_p is the angle of robot, x_p is the x coordinate of C and y_p is the y coordinate of C.

2.1 Kinematic Model

The robot kinematic is described by equation (1),

$$\dot{q} = {}^{q}T_v v \tag{1}$$

$$q = \begin{bmatrix} x_p \\ y_p \\ \theta_p \end{bmatrix} \quad {}^{q}T_v = \begin{bmatrix} \cos(\theta_p) & 0 \\ \sin(\theta_p) & 0 \\ 0 & 1 \end{bmatrix} \quad v = \begin{bmatrix} v_r \\ \omega_r \end{bmatrix}$$

The relation between velocity vectors (ω_{at} and v) is given by the equation $\omega_{at} = {}^{\omega}T_v v$, with

$${}^{\omega}T_v = \begin{bmatrix} 1/r_d & d/2r_d \\ 1/r_e & -d/2r_e \end{bmatrix} \quad \omega_{at} = \begin{bmatrix} \omega_d \\ \omega_e \end{bmatrix}$$

This mobile robot has a nonholonomic constraint, because the driving wheels purely roll and don't slip. This constraint is described by

$$\frac{\partial y_p}{\partial x_p} = \frac{\sin(\theta_p)}{\cos(\theta_p} \rightarrow \dot{y}_p \cos(\theta_p) - \dot{x}_p \sin(\theta_p) = 0 \tag{2}$$

2.2 Dynamic Model

The robot dynamic is represented by the following equation

$$f = M_r \dot{v} + B_r v \tag{3}$$

which is composed by matrix M_r and B_r

$$f = \begin{bmatrix} f_r \\ \tau_r \end{bmatrix} \quad M_r = \begin{bmatrix} m & 0 \\ 0 & J \end{bmatrix} \quad B_r = \begin{bmatrix} \beta_{lin} & 0 \\ 0 & \beta_{ang} \end{bmatrix}$$

where β_{lin} is the frictional coefficient to linear movements, β_{ang} is the frictional coefficient to angle movements, m is the robot mass and J is the inertia moment.

It's very important to consider the following relation

$$f = {}^{\omega}T_v^T \tau \tag{4}$$

with τ is the vector $\begin{bmatrix} \tau_d & \tau_e \end{bmatrix}^T$.

The DC motors dynamic is obtained of electrical (5) and mechanical (6) equations,

$$\begin{aligned} i &= R_m e - R_m K_m \omega_{at} && (5) \\ \tau &= K_m i - J_m \dot{\omega}_{at} - B_m \omega_{at} && (6) \end{aligned}$$

where the vectors are

$$i = \begin{bmatrix} i_d \\ i_e \end{bmatrix} \quad R_m = \begin{bmatrix} 1/R_d & 0 \\ 0 & 1/R_e \end{bmatrix} \quad e = \begin{bmatrix} e_d \\ e_e \end{bmatrix}$$

$$K_m = \begin{bmatrix} K_d & 0 \\ 0 & K_e \end{bmatrix} \quad J_m = \begin{bmatrix} J_d & 0 \\ 0 & J_e \end{bmatrix} \quad B_m = \begin{bmatrix} \beta_d & 0 \\ 0 & \beta_e \end{bmatrix}$$

and $e_{d,e}$ are armature motor voltages, $R_{d,e}$ are windings resistances, $K_{d,e}$ are constants of induced voltage, $J_{d,e}$ are inertia moments of rotors, $\beta_{d,e}$ are motors frictional coefficients and $i_{d,e}$ are armature motors currents.

Using equation (5) in (6) and the result in (4), and the force f , from (4), in equation (3), gives

$$K_v = M_v \dot{v} + B_v v \tag{7}$$

where

$$\begin{aligned} K_v &= {}^{\omega}T_v^T R_m K_m \\ M_v &= M_r + {}^{\omega}T_v^T \cdot J_m \cdot {}^{\omega}T_v \\ B_v &= B_r + {}^{\omega}T_v^T (R_m K_m^2 + B_m) {}^{\omega}T_v \end{aligned}$$

2.3 Linear Representation to Mobile Robot Dynamic Model

A state-space model, with output vector $Y = [S \quad \theta_p]^T$, is obtained from equation (7) and described by

$$\begin{cases} \dot{x} = Ax + Be \\ Y = Cx \end{cases} \tag{8}$$

where S is the linear displacement and

$$x = \begin{bmatrix} v_r \\ \omega_r \\ S \\ \theta_p \end{bmatrix} \quad A = \begin{bmatrix} -M_v^{-1} B_v & \vdots & 0_{2\times 2} \\ I_{2\times 2} & \vdots & 0_{2\times 2} \end{bmatrix} \quad B = \begin{bmatrix} M_v^{-1} K_v \\ 0_{2\times 2} \end{bmatrix}$$

3 INVERSE SYSTEM

The right inverse system is used as an output controller to force the original system output $Y(t)$ to track a given signal $\begin{bmatrix} U_S & U_\theta \end{bmatrix}$. The inverse system, in this paper, is applied to decouple the original MIMO

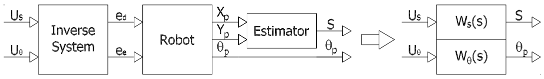


Figure 2: Decoupling based on an inverse system.

(Multiple-Input Multiple-Output) system (8) into two SISO (Single-input Single-Output) systems (Fig. 2).

To get a stable inverse system, it's necessary assume that the system discussed is minimum phase. The inversion algorithm of Hirschorn (Hirschorn, 1979) is applied to decouple the system (8). Deriving the output vector $Y = \begin{bmatrix} S & \theta_p \end{bmatrix}^T$ two times the resulting system is

$$\begin{cases} \dot{x} = Ax + Be \\ \ddot{Y} = C_2 x + D_2 e \end{cases} \quad (9)$$

where D_2 is an invertibility matrix, so the inverse system is

$$\begin{cases} \dot{\widehat{x}} = \widehat{A}\widehat{x} + \widehat{B}\widehat{u} \\ e = \widehat{Y} = \widehat{C}\widehat{x} + \widehat{D}\widehat{u} \end{cases}$$

$$\widehat{A} = A - BD_2^{-1}C_2 \qquad \widehat{C} = -D_2^{-1}C_2$$
$$\widehat{B} = BD_2^{-1}H_2 \qquad \widehat{D} = D_2^{-1}H_2$$
$$\widehat{u} = \begin{bmatrix} 0 & 0 & U_S & U_\theta \end{bmatrix}^T \qquad H_2 = \begin{bmatrix} 0_{2\times 2} & I_{2\times 2} \end{bmatrix}$$

The block diagram in Fig. 2 shows two independent systems, which have the same transfer function $W_S(s) = W_\theta(s) = 1/s^2$ obtained by

$$W_{S,\theta}(s) = [C(sI - A)^{-1}B] \cdot [\widehat{C}(sI - \widehat{A})^{-1}\widehat{B} + \widehat{D}]$$

where $W_{S,\theta}(S) = \begin{bmatrix} W_S(s) & W_\theta(s) \end{bmatrix}^T$.

It's important to remember that linear displacement (S) is not measurable, so an estimated signal is used. The linear model and the inverse system need the exact robot parameters, and we are using a simplified model with uncertain parameters. So a robust adaptive controller will be used to guarantee a good transient under unknown parameters and disturbances. Two controllers are designed, one to each transfer function.

4 CONTROLLER STRUCTURE

The controller structure is divided in five blocks (Fig. 3). First block, which is called kinematic controller, it will calculate reference values (S_{ref} and θ_{ref}) based on desired values (x_d, y_d, θ_d) and absolute positioning system (x_p, y_p, θ_p). Second block is composed by two DMRAC controllers, which are projected to do the robot to reach reference values. These controllers based on references values will supply two control signals (U_S and U_θ) to the inverse system. Third block is the inverse system, fourth block is the robot and the fifth block is the estimator.

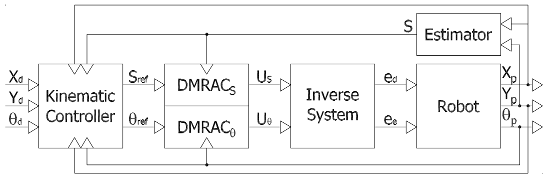


Figure 3: Controller strategy block diagram.

4.1 Estimator

The linear displacement estimator is given by

$$S = \sum_i \mathrm{sgn}(X) \cdot \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \quad (10)$$

where X is

$$X = x_{i+1}\cos(\theta_i) + y_{i+1}\sin(\theta_i) - x_i\cos(\theta_i) - y_i\sin(\theta_i)$$

So, if $X > 0$, we have the linear displacement ($\widehat{\triangle l} > 0$) and if $X < 0$, we have $\widehat{\triangle l} < 0$.

4.2 Kinematic Controller

The Fig 4 shows the new variables used in kinematic controller as Δl_{pos}, that is the distance between robot and any reference point (x_{ref}, y_{ref}), $\Delta\lambda_{pos}$ is the distance of robot to point R_{pos} that is nearest reference point in robot orientation axis. ϕ_{pos} is the angle of robot orientation axis, ($\Delta\phi_{pos} = \phi_{pos} - \theta_p$). θ_d is desired orientation angle and γ is the difference between ϕ and θ_d angles ($\gamma = \phi - \theta_d$).

To get that a robot leaves a point and reaches another point it's necessary $\Delta l_{pos} \to 0$ when $t \to \infty$. Based on decoupled linear model of robot a new auxiliar point R_{pos} (Fig 4) was proposed, to design a positioning controller, because only in this point $\Delta\lambda_{pos} = S_{ref} - S$. So, if $\Delta\lambda_{pos} \to 0$ and $\Delta\phi_{pos} \to 0$, the $\Delta l_{pos} \to 0$. The reference of DMARC$_S$ controller is calculated by

$$S_{ref} = \Delta l_{pos} \cdot \cos(\Delta\phi_{pos}) + S \quad (11)$$

and the reference signal to DMARC$_\theta$ controller is given by

$$\theta_{ref} = \phi_{pos} = \tan^{-1}\left(\frac{y_{ref} - y_p}{x_{ref} - x_p}\right) \quad (12)$$

where the equations (11) and (12) represent the positioning controller and are using only informations from absolute positioning system.

This work uses a mobile reference system to generate new points to the positioning controller for each step of the algorithm. It's based just in the robot kinematic model. The kinematic controller objective is to

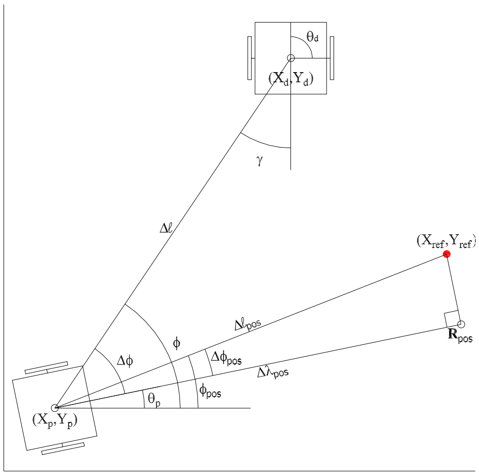


Figure 4: Mobile robot coordinates.

do $\theta_p = \theta_d$ and $\Delta l \to 0$, when $t \to \infty$. For this, we have to do $\theta_p \to \phi_{pos}$, $\phi_{pos} \to \phi$ and $\phi \to \theta_d$, at the same time that $\Delta l \to 0$ when $t \to \infty$.

Based on the functioning of the positioning controller, it was proposed that each new reference point has the same distance between robot and desired point. So

$$\Delta l = \Delta l_{pos} = \sqrt{(x_{ref} - x_p)^2 + (y_{ref} - y_p)^2} \qquad (13)$$

and that the angle of this point is

$$\tan^{-1}\left(\frac{y_{ref} - y_p}{x_{ref} - x_p}\right) = \phi + \gamma \qquad (14)$$

To the equations (13) and (13), we obtain the following equations to positioning controller

$$\begin{cases} x_{ref} &= x_p + \Delta l \cdot \cos(\phi + \gamma) \\ y_{ref} &= y_p + \Delta l \cdot \sin(\phi + \gamma) \end{cases}$$

4.3 Dual-Mode Adaptive Robust Controller

The DMARC uses the compact VS-MRAC structure, that was proposed by Araujo and Hsu (Araujo and Hsu, 1990), changing just the last control signal u_1 (Fig 5) to choose between a VS-MRAC and conventional MRAC controllers.

Consider a linear single-input/single-output time invariant plant with uncertain parameters and transfer function,

$$W(s) = k_p \frac{n_p(s)}{d_p(s)} = \frac{1}{s^2 + \alpha_1 s + \alpha_2}$$

with input u and output y. The reference model is

$$M(s) = k_m \frac{n_m(s)}{d_m(s)} = \frac{k_m}{s^2 + \alpha_{m1} s + \alpha_{m2}}$$

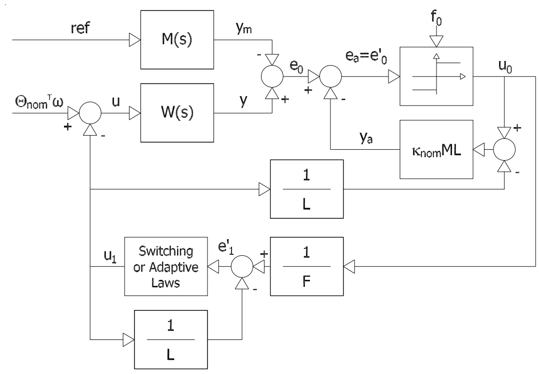


Figure 5: Block diagram of DMRAC controller with $n^* = 2$.

with input ref and output y_m.

The aim is to find u such that the output error

$$e_0 = y - y_m$$

tends to zero asymptotically for arbitrary initial conditions and arbitrary piece-wise continuous uniformly bounded reference signals $ref(t)$.

Following conventional assumptions are made:

1. the plant is observable, controllable, minimum phase ($n_p(s)$ is Hurwitz) and with unknown or uncertain bounded parameters. $d_p(s)$ and $n_p(s)$ are monics polynomials with degree $[d_p(s)] = 2$, degree $[n_p(s)] = 0$ and relative degree $n^* = 2$;
2. the reference model is stable and minimum phase ($n_m(s)$, $d_m(s)$ are Hurwitz). $d_m(s)$ and $n_m(s)$ are monics polynomials with relative degree known (n^*) ($[M(s)]$ has the same relative degree than $W(s)$ and signals $\mathrm{sgn}(k_p) = \mathrm{sgn}(k_m) > 0$ (positive, for simplicity).
3. only input/output measurements are used to find control law $u(t)$.

The following input and output filters are used

$$\begin{cases} \dot{Q}_1 = -\lambda Q_1 + gu \\ \dot{Q}_2 = -\lambda Q_2 + gy \end{cases}$$

where $Q_1, Q_2 \in \Re$ and λ is chosen such that $N_m(s)$ is a factor of $det(sI - L)$. The regressor vector is defined as $\omega^T = \begin{bmatrix} Q_1 & y & Q_2 & ref \end{bmatrix}^T$, and the control u is given by

$$u = \Theta^T \omega$$

where $\Theta^T = \begin{bmatrix} \Theta_{Q1} & \Theta_n & \Theta_{Q2} & \Theta_{2n} \end{bmatrix}^T$ is the adaptive parameter vector.

Considering the above assumptions exists a unique constant vector Θ^*, such that the transfer function of the closed-loop plant $W(S)$, with $u = \Theta^{*T}\omega$, is $M(s)$ (Matching condition). But θ^* is obtained of exact plant parameters. Usually this is not possible in practice, then the Θ is adapted until $e_0 \to 0$ when $t \to \infty$, and eventually (under some signal richness condition) $\Theta \to \Theta^*$.

The vector Θ^* is obtained as following

$$\Theta^* = \begin{bmatrix} (\alpha_1 - \alpha_{m1})/g \\ (\lambda(\alpha_1 - \alpha_{m1}) + (\alpha_2 - \alpha_{m2}) - \alpha_1 g \Theta^*_{v1})/g \\ (\lambda(\alpha_2 - \alpha_{m2}) - \alpha_2 g \Theta^*_{v1} - k_p \lambda \Theta^*_n)/(k_p \cdot g) \\ k_m/k_p \end{bmatrix}$$

The θ_{nom} is a nominal value of adaptive parameter vector (ideally, $\Theta_{nom} = \Theta^*$) and it is calculated as Q^* and based on the decoupled plants $(W_S(s), W_\theta(s))$ where $\alpha_1 = \alpha_2 = 0$.

Suppose that exists a polynomial $L(s) = (s+\delta)$ of degree $N = 1$ where $\delta > 0$ and $\delta \in \Re$ such that $M(s)L(s)$ is SPR (Strictly Positive Real). Consider the following auxiliary signal (a prediction of the output error e_0)

$$y_a = ML\Theta_{2n+1}[L^{-1}u - \Theta^T L^{-1}\omega]$$

where Θ_{2n+1} and Θ are estimates for $1/\Theta^*_{2n}$ and Θ^* (Matching parameters), respectively. The following augmented error is defined

$$e_a = (y - y_m) - y_a = e_0 - y_a$$

Narendra proposes the following modification to guarantee the global stability of the adaptive system

$$\begin{aligned} y_a &= ML[\Theta_{2n+1}(L^{-1}\Theta^T - \Theta^T L^{-1})\omega \\ &\quad + \alpha e_a (L^{-1}\omega)^T (L^{-1}\omega)], \quad \alpha > 0 \end{aligned}$$

To update Θ and Θ_{2n+1} are used the following adaptive integral laws to MRAC controller

$$\begin{aligned} \dot{\Theta} &= -e_a(L{-}1\omega) \\ \dot{\Theta}_{2n+1} &= e_a(L{-}1\Theta^T - \Theta^T L^{-1})\omega \end{aligned}$$

To VS-MRAC controller, we have to introduce the following filtered signals $\xi_0 = L^{-1}\omega$, $\xi_1 = \omega$, $\chi_0 = L^{-1}u$, $\chi_1 = u$.

The upper bounds are defined by

$$\begin{aligned} \overline{\Theta}_{11} &> |\Theta^*_{Q1} - \Theta_{nom,Q1}| & \overline{\Theta}_{12} &> |\Theta^*_n - \Theta_{nom,n}| \\ \overline{\Theta}_{13} &> |\Theta^*_{Q2} - \Theta_{nom,Q2}| & \overline{\Theta}_{14} &> |\Theta^*_{2n} - \Theta_{nom,2n}| \end{aligned}$$

$$\begin{aligned} \overline{\Theta}_1 &= \begin{bmatrix} \overline{\Theta}_{11} & \overline{\Theta}_{12} & \overline{\Theta}_{13} & \overline{\Theta}_{14} \end{bmatrix}^T \\ \overline{\Theta}_{0j} &> \rho \cdot \overline{\Theta}_{1j}, \quad j = 1,2,3,4 \end{aligned}$$

$$\overline{\kappa} > \left| \frac{\kappa^* - \kappa_{nom}}{\kappa_{nom}} \right|$$

where $\rho = \kappa^*/\kappa_{nom}$ (ideally $\rho = 1$) and κ_{nom} is a nominal value for $k^* = 1/\Theta^*_{2n}$ (it is assumed $k_{nom} \neq 0$). Further, it is defined

$$u_{nom} = \Theta^T_{nom}\omega \tag{15}$$

We have plants with $n^* = 2$ and in this case, a VS-MRAC structure, needs a chain of auxiliary errors to get the matching condition. The switching laws are chosen so that the auxiliary errors $e'_i (i = 0,1)$ become sliding modes after some finite time. The equivalent controls $((u_i)_{eq})$ are, in practice, obtained of (u_i) by means of a low pass filter (F) with high enough cut-off frequency.

To adjust the DMARC controller an expression for the μ parameter, based on the idea of the Takagi-Sugeno model, was used. This expression is given by (16), where ℓ is a parameter to be adjusted.

$$\mu = \mathrm{e}^{-(e'_1)^2/\ell} \tag{16}$$

Notice that when $e'_1 \to 0$, $\mu \to 1$ and the algorithm is the MRAC. When e'_1 becomes reasonably high, μ assumes a value sufficiently small, tending to the VSMRAC algorithm. The ℓ parameter has a great importance in the transition between MRAC and VS-MRAC. If ℓ is small the VS-MRAC action will be big.

The DMARC algorithm applied to plants with relative degree $n^* = 2$ is summarized in the Table 1.

Table 1: Algorithm of DMARC controller.

$$\begin{aligned} u &= -u_1 + u_{nom} \\ y_a &= \kappa_{nom} ML \cdot [u_0 - L^{-1}u_1] \\ e'_0 &= e_a = e_0 - y_a \\ (u_0)_{eq} &= F^{-1}(u_0) \\ e'_1 &= (u_0)_{eq} - L^{-1}(u_1) \\ f_0 &= \overline{\kappa}|\chi_0 - \Theta^T_{nom}\xi_0| + \overline{\Theta}^T_0 \cdot |\xi_0| \\ f_1 &= \overline{\Theta}^T_1 \cdot |\xi_1| \\ u_0 &= f_0 \cdot \mathrm{sgn}(e'_0) \\ u_1 &= -\Theta^T_1 \omega \\ \mu\dot{\Theta}_1 &= -\alpha\Theta_1 - \alpha\overline{\Theta}_1 \cdot \mathrm{sgn}(e'_1\omega), \quad \alpha > 0 \end{aligned}$$

5 RESULTS

The result showed in this paper is based on simulation of a micro robot soccer structure. The simulated result considers the main nonlinearities (see (Laura et al., 2006)) as dead zone between $\pm 150mV$ and saturation to values out of $\pm 10V$. It also has noise in input and output signals. The noise was calculated by a random variable with a normal distribution of probability. To inputs noises (e_d, e_e) we have values between $\pm 100mV$. To output noises (x_p, y_p and θ_p) we have values between $\pm 1cm$ to cartesian points and $\pm 8{,}5^o$ to angle.

The unknown parameters are in the dynamic model, and they are obtained of the exact mobile robot model with a normal distribution random variable of probability between 1% and 10%. The parameters used in the simulation are of a realistic robot, that is not symmetrical.

A controller DMRAC is applied to each plant ($W_S(s)$ and $W_\theta(s)$ with relative degree $\rightarrow n^* = 2$). Choosing a reference model to each plant, we have

$$M_S(s) = \frac{2.25}{s^2 + 3s + 2.25} \qquad M_\theta(s) = \frac{100}{s^2 + 20s + 100}$$

and as filters

$$\begin{cases} \dot{Q}_{S1} = -1.5Q_{S1} + 1.5U_S \\ \dot{Q}_{S2} = -1.5Q_{S2} + 1.5S \end{cases} \quad \begin{cases} \dot{Q}_{\theta 1} = -10Q_{\theta 1} + 10U_\theta \\ \dot{Q}_{\theta 2} = -10Q_{\theta 2} + 10\theta_p \end{cases}$$

We choose the polynomial $L_S(s) = s + 1.5$ to DMARC$_S$ with $F = 22s + 1$, $\ell_S = 0.7551$, $\alpha_S = 0.05$ and

$$\begin{aligned} \Theta_{S,nom} &= [\ -2.0 \quad -6.75 \quad 4.5 \quad 2.25\ ]^T \\ \overline{\Theta}_{S,1} &= [\ 2.420 \ \ 13.282 \ \ 2.657 \ \ 0.024\ ]^T \end{aligned}$$

We choose the polynomial $L_\theta(s) = s + 10$ to DMARC$_\theta$ with $F = 56s + 1$, $\ell_\theta = 0.0051$, $\alpha_\theta = 0.05$ and

$$\begin{aligned} \Theta_{\theta,nom} &= [\ -2.0 \quad -300.0 \quad 200.0 \ \ 9.7\ ]^T \\ \overline{\Theta}_{\theta,1} &= [\ 0.073 \ \ 101.161 \ \ 175.750 \ \ 0.1\ ]^T \end{aligned}$$

The Fig. 6 shows a simulation result of the closed loop system. The robot going from the initial point $\begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$ to desired point $\begin{bmatrix} 1 & 1 & 0 \end{bmatrix}^T$. When the robot reaches a circle of $1cm$ radius with center in desired point, it's considered that the task was accomplished. The controller shows a good transient behavior that means in $3.49s$ the robot reaches the desired position without vibrations.

The simulation result is closed to real physical systems, so it includes the main nonlinearities, a plant with unknown parameters, bounded disturbances in the absolute positioning system and in motors driver system. It uses a sampling interval of $10ms$. These facts did not affect the closed loop system behavior.

In the Fig. 6 is possible identify the functioning of the DMARC controller. When the error is big a DMARC operates as VS-MRAC and when the error is small the DMRAC operates as MRAC.

6 CONCLUSIONS

In this paper a controller that decouples multivariable system into two monovariable systems was proposed. For each monovariable system an adaptive robust controller is applied to get desired responses.

The closed loop system with the proposed controller presented a good transient response and robustness to disturbances and errors in the absolute positioning system and in driver system.

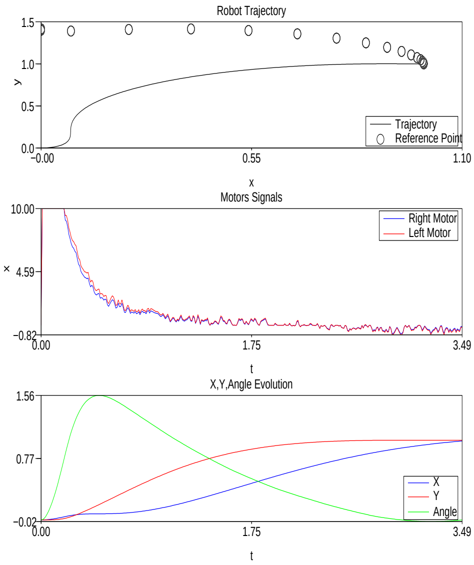


Figure 6: Robot signals (control signal and output plant).

REFERENCES

Araujo, A. D. and Hsu, L. (1990). Further developments in variable structure adaptive control based only on input/output measurements. volume 4, pages 293–298. 11th IFAC World Congress.

Chang, C., Huang, C., and Fu, L. (2004). Adaptive tracking control of a wheeled mobile robot via uncalibrated camera system. volume 6, pages 5404–5410. IEEE International Conference on Systems, Man and Cybernetics.

Cunha, C. D. and Araujo, A. D. (2004). A dual-mode adaptive robust controller for plants with arbitrary relative degree. Proceedings of the 8th International Workshop on Variable Structure Systems, Spain.

Dixon, W. E., Dawson, D. M., Zergeroglu, E., and Behal, A. (2001). Adaptive tracking control of a wheeled mobile robot via uncalibrated camera system. *IEEE Transactiond on on Systems, Man and Cybernetics*, 31(3):341–352.

Hirschorn, R. M. (1979). Invertibility of multivariable nonlinear control systems. *IEEE Transactions on Automatic Control*, AC-24:855–865.

Hsu, L. and Costa, R. R. (1989). Variable structure model reference adaptive control using only input and output measurements - part 1. *International Journal of Control*, 49(02):399–416.

Laura, T. L., Cerqueira, J. J. F., Paim, C. C., Pomlio, J. A., and Madrid, M. K. (2006). Modelo dinmico da estrutura de base de robs mveis com incluso de no linearidades - o desenvolvimento do modelo. volume 1, pages 2879–2884. Anais do XVI Congresso Brasileiro de Automtica, 2006.

ESCAPE LANES NAVIGATOR
To Control "RAOUL" Autonomous Mobile Robot

Nicolas Morette, Cyril Novales and Laurence Josserand
Laboratoire Vision & Robotique,Orleans Universiy,63 avenue de Lattres de Tassigny,18020 Bourges cedex, France
nicolas.morette@bourges.univ-orleans.fr

Keywords: Robotics, Mobile robots, Autonomous robot, Navigation, Escape lanes.

Abstract: This paper presents a navigation method to control autonomous mobile robots : the escapes lanes, applied to RAOUL mobile system. First, the formalism is introduced to model an automated system, and then is applied to a mobile robot. Then, this formalism is used to describe the escape lanes navigation method, and is applied to our RAOUL mobile robot. Finally, implementation results simulations validate the concept.

1 INTRODUCTION

In the autonomous robotics field, the navigator is defined as a module whose purpose is to find a relevant trajectory for a robot within its local surrounding area. The navigator has to deal with various robot constraints (such as kinematic constraints or saturations), and has to find a trajectory which avoid collisions with static or mobile obstacles. The trajectory proposed by the navigator respecting these constraints is followed by the robot using the pilot and servoings modules.

There are various methods to generate a path for an autonomous mobile robot in an unknown environment. Artificial potential fields methods (Agirrebeitia, 2005) generate a path along the weakest potential gradient, in a potential fields map where obstacles are associated with strong potentials. The neural networks methods define a succession of nodes in a neural network map of the robot environment (Lebedev, 2005). Fuzzy Logic methods define a set of logic rules and can drive a robot to move safely in its environment (Xu, 1999).

However, according to our definition, a navigation method is a complement to this kind of approach (called path planning methods), whose purpose is to find a trajectory in order to follow a path generated by a path planning method under local constraints. Moreover, a navigation method has to generate smooth trajectories.

Two kinds of trajectory generation methods are identified: those using an inverse model of the robot, and those using a direct model. Among the first ones, the flat output method controls the robot in order to follow a calculated trajectory (Fraisse, 2002). (Munoz, 1994) have parametered B-splines methods to determine a time parametered trajectory for the robot. Among the second ones (Belker, 2002) used an hybrid neural networks method to project acceptable trajectories for the robot in a local neural network map. In this paper, another direct model method based on escape lanes (Novales, 1994) is developed to project acceptable trajectories for the robot in its local environment.

2 MODEL OF THE ROBOT

Lots of mobile robot navigation methods are based on inverse model: the robot desired trajectory is given by the inverse model, which delivers the articular set points for the servoings. Due to the mechanical design, sometimes there is no solution (non-holonomy). Moreover, if the robot evolves in a constrained environment, cartesian constraints must be expressed in the robot articular space.

We have chosen to use a direct model of the mobile robot to control it. There is always a solution, and the environmental constraints can be expressed directly in the cartesian space. As a consequence, we need to use a global model of the robot and of its environment, which allows to project all the admissible trajectories in a near future (a time horizon of few seconds).

2.1 Formalism of an Automated System

Based on the formalism developed by Sontag (Sontag, 1990), a mobile robot can be modeled with spaces and functions defined as follows (Figure 1):

$$\Sigma = (t, \chi, U, \varphi, \Psi, h) \quad (1)$$

where t is the time space,

U is the input space of the system; u are the vectors of this input space,

χ is the state space of the system; X are the vectors of this state space,

Ψ is the output space of the system; Y are the vectors of this output space,

φ is the state function which gives a current state vector X_a (at time t_a) knowing the initial time t_0, the current time t_a, the initial state X_0 and the input function ω (defined later).

h is the output function which gives the current output vector Y_a (at time t_a) knowing the current time t_a and state X_a.

The input function ω associates an input vector u_a to the current time t_a ($\omega(t_a) = u_a$).

The transfer function λ associates the current output vector Y_a to the current input vector U_a at time t_a and the input function ω (usually it is $\varphi \circ h$).

Sontag defines simplified functions $\xi(t)$ and $\lambda(t)$ associating either a state vector or an output vector directly from the current time.

$\xi(t)$, the simplified state function, makes it possible to define state X_a of the system time t_a: $\xi(t_a) = X_a$. Note that ξ is the image of ω by φ in the state space: $\xi(t_a) = \varphi(t_0, t_a, X_0, \omega)$

$\bar{\lambda}$ (t) is the simplified exit function which gives the exit vector Y_a of the system at time t_a: $\bar{\lambda}$ (t_a) = Y_a. Note that $\bar{\lambda}$ is the image of ξ by h in the space of exit ψ: $\bar{\lambda}(t_a) = h \circ \xi(t_a)$.

2.2 Spaces Specification

In the case of a mobile robot, the contents of the vectors associated with each space are specified, in order to introduce our navigation method in section 3.

In the input space U of the robot, vectors contains set points of the actuator servoings input, i.e. the articular velocities of the robot: $u = (\dot{q}_1, \dot{q}_2, \ldots, \dot{q}_n)^T$.

In the state space χ of the mobile robot, the state vectors are defined as kinematics values, i.e. the curvilinear velocities and the rotation velocity of the mobile robot: $X = (\dot{s}, \dot{\theta})^T$.

The robot output space Ψ is the configuration space (Lozano-Perez, 1983), where output vectors are defined as the coordinates/orientation of the robot: $Y = (x, y, \theta)^T$.

This implies the output function becomes a differential function and does not depend only on the current state. The new output function L which gives the current output vector Y_a at the current time t_a depends on the initial time t_0, on the initial output Y_0 and on the simplified state function ξ.

Our mobile robot is then defined by $\Sigma = (t, \chi, U, \varphi, \Psi, L)$.

2.3 Definition of a Trajectory

A trajectory Γ of a given system Σ defined on an interval of time $[t_0, t_0 + \tau]$ is composed by the simplified state function ξ and the function of entry ω defined on the same interval: $\Gamma = (\xi, \omega)$

nb: the simplified exit function $\bar{L}$ (t) is the image of the trajectory $\Gamma(t)$ in the output space. Typically, it is the "trace on the floor" of the trajectory of the robot.

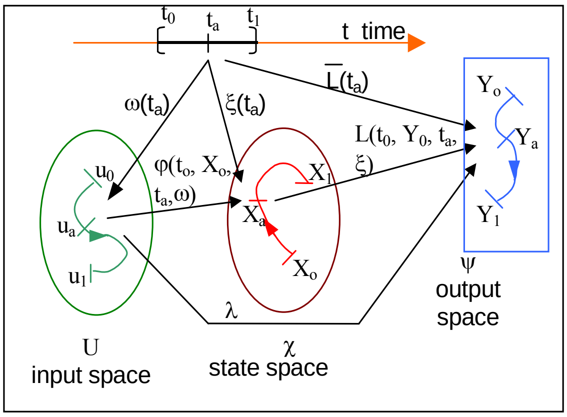


Figure 1: Automated system $\Sigma = (t, \chi, U, \varphi, \Psi, L)$.

3 ESCAPE LANES FORMALISM

Similarly to animal strategy, our navigation is based on direct models. When a mobile robot is considered in a given state, we can project in a few seconds horizon all trajectories that it can perform. These trajectories can be blended with the environment and with the motion goal to select the most appropriate trajectory. These selected trajectories become the new set points of the mobile robot. In order to give the ability to the robot to react in real time to its environment, the process is repeated periodically.

The different steps of the escape lanes method are to:

- generate all acceptable trajectories Γ to be performed by the robot (called escape lanes) on a temporal horizon τ,
- eliminate the blocked escape lanes (e.g. intersect or pass too close to obstacles),
- choose a free escape lane for the robot, using a selection criterion.

The strong point of this method is that it uses only the robot direct model, the inverse function of φ does not need to be defined.

The whole operation is reiterated periodically and is represented on Figure 2.

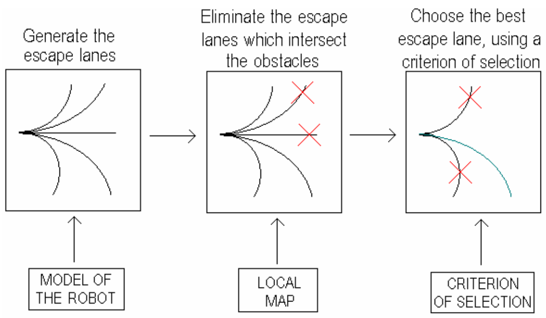


Figure 2: Escape lanes principle.

3.1 Acceptable Trajectories by the Robot and Escape Lanes

An input function ω is acceptable by the robot on an interval $[t_0, t_0+ \tau]$ if it respects the following constraints:

-constraints corresponding to the direct kinematics model of the robot, *i.e.* its possibilities of movements (kinematics constraints),

-constraints of actuators saturation,

-constraints on the dynamics of the actuators (maximum accelerations, saturations…)

$\Omega[t_0, t_0+ \tau]$ is called the entire set of the acceptable input functions on the interval $[t_0, t_0+ \tau]$:

$$\Omega[t_0, t_0+\tau] = \{\omega \in U[t_0, t_0+\tau] \,/\, \omega \text{ acceptable } \} \quad (2)$$

$\Lambda[t_0, t_0+ \tau]$ is called the set of the acceptable trajectories by the robot on the interval $[t_0, t_0+ \tau]$. It corresponds to the set of trajectories associated with the input function $\omega \in \Omega[t_0, t_0+ \tau]$. $\Lambda[t_0, t_0+ \tau]$ is also called the escape lanes of the robot at time t_0 and on the temporal horizon τ.

$$\Lambda[t_0, t_0+\tau] = \{\Gamma = (\xi, \omega) \,/\, \omega \in \Omega[t_0, t_0+\tau] \} \quad (3)$$

3.2 Elimination of the Blocked Escape Lanes

The following stage consists in comparing the image of each trajectory $\Gamma \in \Lambda$ in the output space Ψ to the obstacles map called Θ. This map can be displayed in several forms, but must imperatively present the obstacles position with respect to the robot. Using the elimination criterion C_{el}, the ensemble of the free escape lanes Λ_L is determinated by:

$$\Lambda_L = \{\Gamma = (\xi, \omega) \,/\, C_{el}(\lambda_\Gamma, \Theta) = 0, \Gamma \in \Lambda[t_0, t_0+\tau] \} \quad (4)$$

Escape lanes which intersect the obstacles or pass too close to them are eliminated ($C_{el} \neq 0$), according to C_{el}. The remaining escape lanes constitute the free escape lanes set Λ_L.

By association, the ensemble of the free acceptable input functions Ω_L can be defined as the entire set of the input functions ω which correspond to the free escape lanes Λ_L.

$$\Omega_L = \{\omega \in \Omega \,/\, , \Gamma \in \Lambda_L \} \quad (5)$$

3.3 Selection of the Best Free Escape Lane

The last stage consists in determining the best escape lane among Λ_L to reach a target ε provided by the higher control level (*i.e.* the path planner).

To select the best free escape lane a criterion C_{choice}, is applied to quantify the relevance of each free escape lane to achieve this goal. The selected escape lane is associated to the optimal value of the criterion, and is called Γ_{chosen}. The corresponding function of entry is called ω_{chosen}, and is sent to the lower control level (*i.e.* the pilot).

$$\Gamma_{chosen} = \{\Gamma = (\xi, \omega) \,/\, C_{choice}(\Gamma, \varepsilon) \text{ optimal}, \Gamma \in \Lambda_L \} \quad (6)$$

$$\omega_{chosen} = \{\omega \in \Omega_L \,/\, \Gamma = \Gamma_{chosen\,L} \} \quad (7)$$

3.4 Discussion on Criteria

The choice of the C_{choice} and C_{el} criteria depends on the application to be carried out by the robot. For an exploration mission, when the purpose of the robot is to chart the surrounding area, a severe criterion of elimination C_{el} has to be established to make sure that the robot passes at a safe distance away from obstacles (the obstacles positions are known, but their shapes and volumes are not). A criterion of selection C_{choice} that supports the most efficient trajectories in term of energy may be used.

On the other hand, when considering missions of an industrial type, the robot velocity is favored, and thus the C_{choice} criterion must support the fastest trajectories (at the expense of the energy consumption). Thus, these criteria must be adapted to the situation (mission and kind of environment).

The input function corresponding to the selected trajectory (ω_{chosen}) is actually applied to the robot. Indeed the complete operation (from the generation of escape lanes of the robot to the selected selection of Γ_{chosen} and ω_{chosen}), is performed periodically, and with a period T_e about ten times smaller τ. Therefore only a short portion of the Γ_{chosen} trajectory is actually followed by the robot; a new one is proposed every T_e.

4 APPLICATION TO RAOUL MOBILE ROBOT

4.1 Presentation of RAOUL

RAOUL (figure 3) is an autonomous mobile robot, able to run in an unknown environment, finding a trajectory on its own to avoid collision with static or mobile obstacles. Raoul is built on a Robuter platform (Robosoft) with an additional computer (PC/RTAI-Linux) and exteroceptive sensors: two Sick telemeters laser (a front one and a rear one) and one laser goniometer.

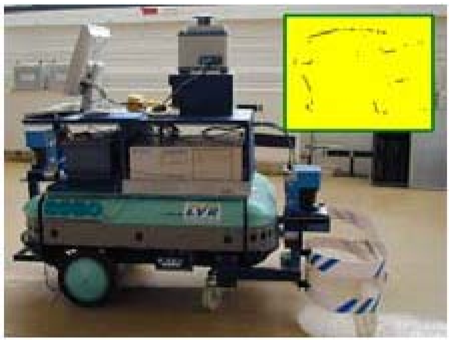


Figure 3: The RAOUL robot.

The control architecture (figure 4) is a multi-level architecture developed by the laboratory of vision and robotic of the University of Orleans (Mourioux, 2006). Each level corresponds to a perception/action loop. Low levels correspond to fast reaction behaviors and high levels to "intelligent" behaviors.

Level 0 represents the Articulated Mechanical System (AMS). Level 1 of the architecture corresponds to the servoings loop using proprioceptive sensor informations. On level 2, the "pilot" module performs emergency reactive decisions, using data from the two laser range telemeters. The third level, the "navigator", is in charge of finding a local trajectory for the robot, using a local map module (Canou, 2004). On the upper level, the "path planner" has the mission to find a global path for the robot.

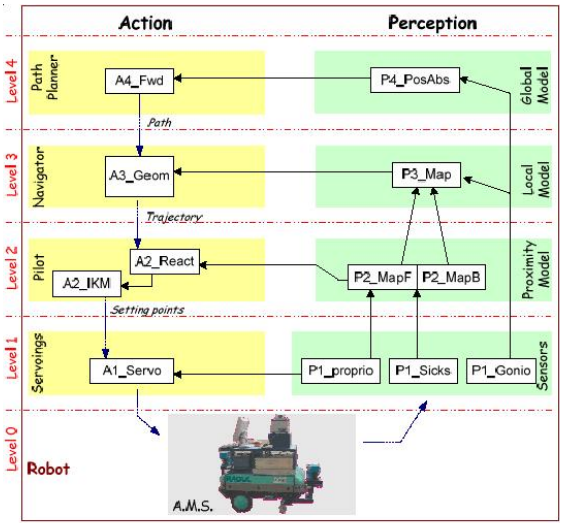


Figure 4: the control architecture.

The part developed in this work concern only the navigation level.

4.2 Acceptable Trajectories

First the constraints of our system are determined to define the ensemble of the functions of acceptable entry Ω on a temporal horizon τ.

RAOUL is a mobile robot with differential wheels; we assume that the motion is done without slipless on the ground. According to the trigonometrical direction for w_1 and w_2 (the rotation velocity of the two driving wheels) we obtain:

$$\dot{s} = \frac{s_1 + s_2}{2} = \frac{w_2 R - w_1 R}{2} \quad (8)$$

$$\dot{\theta} = \frac{s_1 - s_2}{2L} = \frac{w_2 R + w_1 R}{2L} \quad (9)$$

Where s corresponds to the curvilinear distance traversed by point C, and θ the orientation of the robot.

The dynamic constraints and saturations depend on the robot itself and correspond to $w_{\min}$, w_{max}, $\dot{w}_{\min}$, $\dot{w}_{\max}$.

It is then necessary to choose a family (or a number finished families) of input functions, that must be selected according to the robot capacities. For example, a car-like robot is not controlled the same way as a robot with differential wheels.

Linear functions are used for RAOUL robot and are represented on Figure 5.

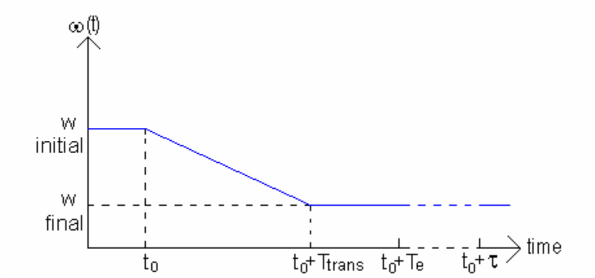


Figure 5: Function of entry.

This family of input functions can be expressed in the following way:

$$w(t) = w_{initial} + \frac{(w_{final} - w_{initial}) \times (t - t_0)}{T_{trans}} \quad (10)$$
$$for \quad t_0 < t < t_0 + T_{trans}$$

Under the constraints:

$$\begin{aligned} & w_{\min} \le w_{initial} \le w_{\max} \\ & w_{\min} \le w_{final} \le w_{\max} \\ & \dot{w}_{\min} \le \frac{(w_{final} - w_{initial})}{T_{trans}} \le \dot{w}_{\max} \end{aligned} \quad (11)$$

From this family of acceptable input functions, we obtain a family of acceptable trajectories $\Lambda_{limited}$:

$$\Lambda_{LIMITED} = \{\Gamma = (\xi, w_{familly}) \,/\, w_{familly} \in \Omega[t_0, t_0 + \tau]\} \quad (12)$$

With:

$$\begin{aligned} & w_{familly} = (w_1(t), w_2(t))^T \\ & \xi = (\dot{s}, \dot{\theta}) \end{aligned} \quad (13)$$

Where:

$$\begin{aligned} & \dot{s} = \frac{R}{2} \cdot (w_2(t) - w_1(t)) \\ & \dot{s} = \frac{R}{2} \cdot \left\{ (w_{2i} - w_{1i}) + \left[(w_{2f} - w_{2i}) - (w_{1f} - w_{1i} \right] \cdot \frac{t - t_0}{T_{trans}} \right\} \end{aligned} \quad (14)$$

$$\begin{aligned} & \dot{\theta} = \frac{R}{2} \cdot (w_1(t) + w_2(t)) \\ & \dot{\theta} = \frac{R}{2} \cdot \left\{ (w_{2i} + w_{1i}) + \left[(w_{2f} - w_{2i}) + (w_{1f} - w_{1i} \right] \cdot \frac{t - t_0}{T_{trans}} \right\} \end{aligned} \quad (15)$$

nb: these trajectories are projected starting from the X_0 state and of the Y_0 output of the robot at time t_0.

4.3 Elimination of the Blocked Escape Lanes

To eliminate the blocked escape lanes a comparison criterion between $\Lambda_{limited}$ and the map of the obstacles Θ is proposed.

The local model of the robot environment made by (Canou, 2004) is used, built in-line using 2 laser-range finders. The obstacles map Θ is composed as a set of segments of known two ends coordinates. These segments represent the obstacles perimeter in the local environment of the robot, projected within the moving plane of the robot.

To determine if an escape lane of $\Lambda_{limited}$ is free or blocked, the C_{el} elimination criterion is used between its image λ_i in Ψ and each obstacle segment of Θ. C_{el} (λ_i, Θ) = 0 if:

$$dist\left\{\lambda_i(t) / sgmt_j\right\} - (L + margin) > 0 \ \forall\ j. \qquad (16)$$

L: maximum distance between the periphery of the robot and the center C of the axis connecting the two driving wheels of the robot; L is also called width of the robot.

dist{$\lambda_i(t)$ / sgmt$_j$} is the minimal distance between λ_i points and the points belonging to segment $_j$.

Escape lanes that do not verify C_{el} (λ_Γ, Θ) = 0 are eliminated. The ensemble of the remaining escape lanes is $\Lambda_{L\,/\,limited}$.

4.4 Selection of the Best Free Escape Lane

Finally a criterion is given to choose the escape lane to be kept as set point for the robot. The choice of this criterion takes into account the distance and the orientation of the robot at the end of the trajectory, compared to its target ε (provided by the path-planner).

The criterion is given by:

$$C_{choice}(\Gamma_i, \varepsilon) = \sqrt{(x_i(t_0+\tau) - x_\varepsilon)^2 + (y_i(t_0+\tau) - y_\varepsilon)^2} \times fact \qquad (17)$$

$$fact = 1 + k_\theta \times \left|\theta_i(t_0+\tau) - \theta_{target/robot}\right| \qquad (18)$$

k_θ is a positive real fixed empirically by carrying out simulations and experimentations.

This criterion computes the cartesian distance from the robot with ε by balancing it with its orientation compared to ε.

The $\Lambda_{L\,/\,limited}$ escape lane which minimizes this criterion is chosen and called Γ_{chosen}. The associated input function, ω_{chosen}, is applied to the robot.

To improve the relevance of the elimination criterion C_{el}, the margin value in the formula (17) may be adjusted according to the orientation of the robot with respect to the obstacle (C-obstacles (Lozano, 1983)).

5 IMPLEMENTATION

Matlab was used to carry out simulations and the implementation on the robot was made in C language on Linux RTAI system.

5.1 Discretization of the Input Space

In our case the input space is discretized on 5 by 5 mode, *i.e.* when the generation of possible trajectories is carried out, 5x5 = 25 trajectories are generated corresponding to the 25 possible input functions ; each one moving towards one of the 25 possible couples (w_1, w_2) reached at the end of T_{trans}.

The more the space of entry is discretized in a great number, the greater trajectories number is possible for our robot at time t_0. As a consequence the computation time is higher. It is thus necessary to find a compromise.

5.2 Generation Off-Line or On-Line of $\Lambda_{Limited}$

Experimentally, there are two possibilities to carry out the generation of the possible $\Lambda_{limited}$. Thos generation can be performed on line, *i.e.* while the robot is moving and just before their application, or to do it partially off line.

When fully done on line, 5x5=25 trajectories have to be generated for the robot (or n^2 if Ω is discretized by n on n). That is multiplied by the number of points on each trajectory, *i.e.* τ divided by the step of calculation over time. In simulations, a temporal horizon τ=3s is used for a step of 0.05s, *i.e.* a total of 60 points per trajectory and thus 1500 points overall.

Another solution consists in carrying out the generation of $\Lambda_{limited}$ off line. However, it is then necessary to generate 5x5 x 5x5 = 625 trajectories. The trajectories to join the 25 discretized final couples (w_1, w_2) have to be computed, with the 25 possible initial couples as a starting point. It is then necessary to store all these trajectories during on-line navigation; the 25 trajectories corresponding to the initial state of the robot are projected, using the following transform matrix:

$$\begin{bmatrix} x \\ y \\ \theta \end{bmatrix}_{t+t_0} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}_{t_0} + \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}_{t/\Gamma_{chosen}} \qquad (19)$$

This solution is used when the robot does not have the necessary power to perform all the calculations on line. The totality of the generated trajectories is represented on figure 6.

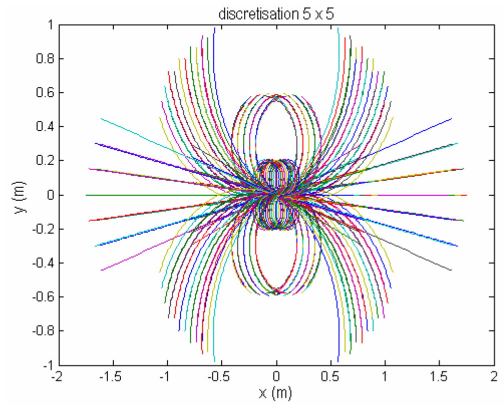


Figure 6: Possible trajectories of the robot stored in memory.

5.3 Implementation

Experimentally, we are constrained to perform the navigation calculation one period T_e ahead of time. Indeed, calculations for the period [t_0, t_0+T_e] as from time t_0 since it takes a certain time for calculation and since the robot continues to move forward during this time.

The navigation calculation for the period [t_0, t_0+T_e] must thus be carried out during the period [t_0-T_e, t_0]. To know the robot output Y at t_0, we carry out the approximation which the robot will have followed exactly the escape lane chosen over the period [t_0-T_e, t_0], starting from its output Y at the moment t_0- T_e.

$$\begin{bmatrix} x \\ y \\ \theta \end{bmatrix}_{t_0} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}_{t_0-T_e} + \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}_{t_0-T_e} \bullet \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}_{T_e / \Gamma_{chosen} \atop on\,[t_0-T_e,t_0]} \quad (20)$$

6 RESULTS

6.1 Simulation

The simulation program that we have developed is composed of three parts. The first part calculates and stores in memory the possible movements of the robot (off-line generation of $\Lambda_{limited}$).

The second creates a chart of a virtual environment in which the robot will navigate. This environment consists in obstacles viewed as segments (these segments represent the periphery of the obstacles in the navigation plan).

The last part corresponds to navigation itself, and includes the projection stages of the robot escape lanes, the elimination of the blocked escape lanes and the choice of the most suitable escape lane. This part is performed until the virtual robot reaches a desired localization (beyond a certain number of loops the program stops). Finally the program traces the path that the robot carries out on the chart, and gives the inputs list sent to the robot with each iteration.

6.2 Results

Figure 7 shows a simulation representing displacements of the robot using a differential wheels type model. The robot starts at point coordinates (-8, -8), and must go to (-5, 4).

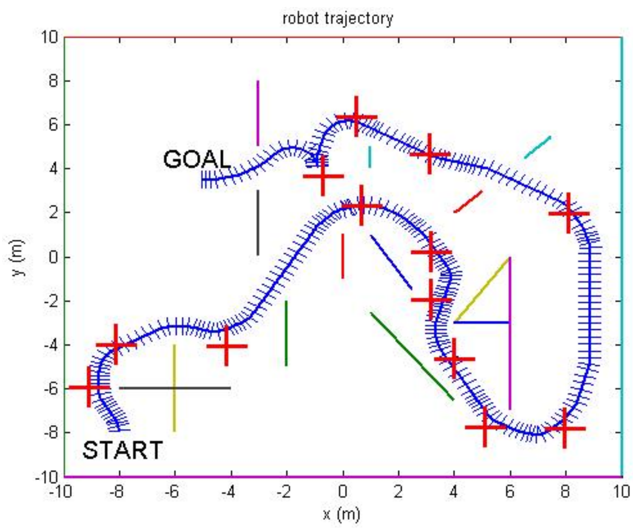


Figure 7: Simulation of obtained trajectory.

The trace of the displacements achieved by the center of the robot is represented in blue, and each new iteration of the program of navigation is symbolized by a small feature perpendicular to the trace. The red crosses represent the points of passage provided to the navigator.

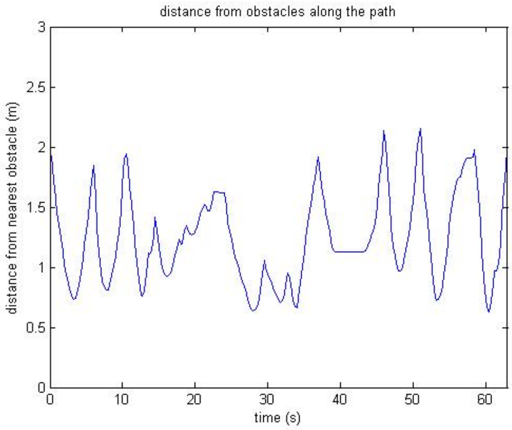


Figure 8: distance robot/obstacle.

The distance between the robot and its nearest obstacle along the path is represented on figure 8. We can note that the robot never enters in collision with the obstacles because the distance from obstacles is larger than the robot width (50cm). Moreover in the event of appearance of an unexpected obstacle, the pilot (*i.e.* the lower level of control) has the capacity to perform reflex decisions, of absolute priority, to avoid this kind of obstacle in emergency. We can note the great flexibility of the generated trajectories.

7 CONCLUSION

This study showed that escape lanes is an efficient navigation method, able to generate fluent moves for RAOUL autonomous mobile robot. Its strong point is that it only uses the direct kinematics model of the robot, ensuring that the robot is actually able to perform the desired moves. Indeed it's more difficult to transpose constraints in the input space of the robot using an inverse model.

However it's only a navigation method, meaning that it must work cooperatively with a path planner module, which gives a global path to follow to the navigator. The purpose of the navigator is to follow this path as close as possible, under the local constraints of the environment the robot evolves in.

The next step is to implement this method with a planner on a Cycab robot, using a GPS for the path planning module. Cycab is a car-like robot, with only one mobility degree. This will provide a stronger non holonomy constraint in the implementation of the proposed navigation method.

REFERENCES

Agirrebeitia, J., Aviles, R., de BUSTOS, I. F., Ajuria, G., June 2005, *A new APF strategy for path planning in environments with obstacles*, Mechanism and Machine Theory, Volume 40, Issue 6, , Pages 645-658

Belker, T., Schulz, D., *Intelligent Robots and System, 2002, Local Action Planning for Mobile Robot Collision Avoidance*, IEEE/RSJ International Conference on Volume 1, 30 Sept.-5 Oct. 2002 Page(s):601 - 606 vol.1

Canou, J., Mourioux, G., Novales, C. Poisson, G., April 26 – May 1st 2004 , *A local map building process for a reactive navigation of a mobile robot*, Proceedings of IEEE International Conference on Robotics and Automation, pp4839-4844, New Orleans, USA

Fraisse, P., Gil, A. P., Zapata, R., Perruquetti, W., Divoux, T., 2002, *Stratégie de commande collaborative réactive pour des réseaux de robots*.

Lebedev, D. V., April 2005. *The dynamic wave expansion neural network model for robot motion planning in time-varying environments, Neural Networks*, Volume 18, Issue 3, Pages 267-285.

Lozano-Perez, T., 1983, *Spatial Planning: a Configuration Space Approach, IEEE Transaction and computers,* vol. C-32, no. 2, Fev. 1983.

Novales, C., 1994, *Navigation Locale par Lignes de Fuite, rapport de thèse : Pilotage par actions réflexes et navigation locale de robots mobiles rapides*, chapitre IV, soutenue le 20 octobre 1994, Pages 87 à 107

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

Munoz, V., Ollero, A., Prado, M., Simon, A., 1994 IEEE International Conference on 8-13 May 1994, *Mobile robot trajectory planning with dynamic and kinematic constraints, Robotics and Automation*, 1994. Proceedings, Page(s):2802 - 2807 vol.4

Sontag, E. D., 1990. *Mathematical control Theory – Deterministic Finite Dimensional Systems*, ED-Spronger-Velag New-York 1990.

Xu, W. L., August 1999, *A virtual target approach for resolving the limit cycle problem in navigation of a fuzzy behavior-based mobile robot*, Institue of Technology and Institute of Technology and Engineering, College of Sciences, Massey University, Palmerston North, New Zealand ; Received 22 June 1998 ; accepted 20 August 1999

INITIAL DEVELOPMENT OF HABLA (HARDWARE ABSTRACTION LAYER)
A Middleware Software Tool

Andrés Faíña, Francisco Bellas and Richard J. Duro
Integrated Group for Engineering Research, University of Coruña, Mendizabal S/N, Ferrol, Spain
Email: afaina@udc.es, fran@udc.es, richard@udc.es

Keywords: Middleware software tool, control system abstraction, autonomous robotics.

Abstract: In this work we present the initial implementation of a middleware software tool called the Hardware Abstraction Layer (HABLA). This tool isolates the control architecture of an autonomous computational system, like a robot, from its particular hardware implementation. It is provided with a set of general sensors and typical sensorial processing mechanisms of this kind of autonomous systems allowing for its application to different commercial platforms. This way, the HABLA permits the control designer to focus its work on higher-level tasks minimizing the time spent on the adaptation of the control architecture to different hardware configurations. Another important feature of the HABLA is that both hardware-HABLA and HABLA-control communications take place through standard TCP sockets, permitting the distribution of the computational cost over different computers. In addition, it has been developed in JAVA, so it is platform independent. After presenting the general HABLA diagram and operation structure, we consider a real application using the same deliberative control architecture on two different autonomous robots: an Aibo legged robot and a Pioneer 2Dx wheeled robot.

1 INTRODUCTION

The origin of this work can be found in the research on autonomous robotics carried out in our group. It usually implies the acquisition of one or more commercial robots or the construction of new ones. In any case, the robots must be programmed in their own programming language (C, C++, Lisp, etc) and their particular hardware architecture must be taken into account when developing its control software. Manufacturers usually provide an API (Application Program Interface) with a reduced set of high level commands to develop basic functionalities. Examples of these tools are AIBO SDE (AIBO, 2007) (a software development environment for AIBO robots) or Aria (Aria, 2006) (that supports different robot models from Activmedia Robotics). The main problem with these tools is that they are specific for a given family of robots and they require the designer to develop the control architecture in a preestablished programming language.

Nowadays, there is no standardization or even a general preference about the most appropriate programming language to be used in autonomous robotics, and the trend is to continue in the same way. As a consequence, even though two robots may have the same sensors and actuators, if we want to execute the same control architecture on both, we must modify the programming and adapt it to the particular language and implementation of each robot.

In order to deal with this problem, more general frameworks have been developed in recent years trying to achieve complete independence from the robot manufacturer. These tools can be classified as *middleware software* that abstracts the control architecture from the particular hardware. Examples of this kind of tools are Miro (Utz, 2002), Webots (Michel, 2004) or YARP (Metta, 2006) that permit development in a broad range of different robotic systems. Miro is a distributed object oriented framework for mobile robot control, based on CORBA (Common Object Request Broker Architecture) technology. It is focused on wheeled robots such as Pioneer or Sparrow and it does not provide support, for example, for legged robots like Sony's AIBO (very popular in autonomous robotics research) or humanoid prototypes. Webots "provides a rapid prototyping environment for modelling, programming and simulating mobile robots". It includes several libraries that allow the designer to transfer the control programs to many commercially available real mobile robots. Finally, YARP "is written by and for researchers in humanoid robotics, who find themselves with a complicated pile of hardware to control with an equally complicated pile of software".

The abstraction level provided by these tools could be necessary in other autonomous computational systems (different from robotics) with sensors and actuators of very different nature and with a complex control system like, for example, domotic applications. Taking into account these generalization, in this work we propose the creation of a middleware tool to be applied in different autonomous computational systems characterized by different sensors and actuators controlled through a complex architecture that could be executed remotely. The only reference we have found that follows this philosophy and supports different hardware devices and application fields is Player (Gerkey, 2003). This tool provides an interface and a protocol to manage sensorial devices and robots over a network and accepts different programming languages for the control architecture. It runs on several robotic platforms and supports a wide range of sensors. At this time the developments outside the robotics field are limited. The general middleware tool we propose is called the Hardware Abstraction Layer (HABLA).

2 HARDWARE ABSTRACTION LAYER (HABLA)

The desired features for the Hardware Abstraction Layer can be summarized into a group of six:

Device independence: it must support the most common sensors and actuators present in autonomous computational systems such as cameras, microphones, infrared sensors, sonar sensors, motion sensors, etc. In addition, it should be provided with particular implementations for the most typical commercial platforms, for example, the robots used in research like Pioneer, Kephera, Aibo, etc.

Virtual sensing and actuation: it must provide typical sensorial processing such as color segmentation or sound analysis so that higher level information like distance to nearby objects or sounds can be considered by the control architecture.

Computational cost distribution: it must support communications through a computer network by TCP sockets in order to execute the control architecture, the low-level control program and the different elements of the HABLA itself over different computers. It seems obvious that, for example, sound or image processing should not be executed directly in the robot.

Control architecture independence: it must be independent of the programming language used in the control architecture, this is, we do not impose any particular programming language.

Scalability: the HABLA should present a modular design and an open architecture in order to increase the number of supported sensors and actuators corresponding to new commercial platforms.

Operating System independence: it must be implemented in JAVA to achieve operating system independence. In addition, JAVA is the most standard object oriented language, so the HABLA could easily include contributions from the research community.

Figure 1 shows a general diagram of the Hardware Abstraction Layer for a typical autonomous computational system. The left block represents the control architecture that requests sensorial information from the low level devices and provides the action or actions to be executed through the actuators. A basic idea behind the HABLA development is that we assume that the control architecture requires high level sensorial information, this is, the basic sensorial processing is not executed in the control architecture. In addition, the actions selected can be complex actions, and not only individual commands to the actuators.

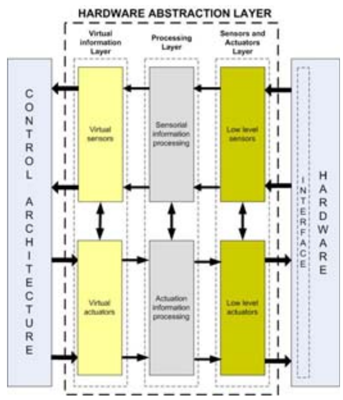


Figure 1: General diagram of the Hardware Abstraction Layer for a typical robotic system.

The right block in Figure 1 represents the hardware (sensors and actuators) that provides sensorial information to the control architecture and receives the action or actions that must be applied. In this case, the sensors provide low level information (with no processing) and the actuators require low level data too.

As shown in Figure 1, the middle block represents the Hardware Abstraction Layer, an element that isolates the high level information handled by the control architecture from the low level information handled by the sensors and actuators. The communications between control architecture and HABLA and between HABLA and hardware use TCP sockets, as commented before, in order to permit the distributed execution of these three basic elements.

Inside the HABLA we can see three sequential layers: the *sensors and actuators layer,* the *processing layer* and the *virtual information layer* in order of increasing processing of the information. The HABLA is implemented using JAVA and each layer contains methods that perform a particular function. The methods of a given layer can use and provide information from/to the neighboring layer as represented by the arrows in Figure 1. The information exchange between methods of the same layer is also possible, but not the exchange between non-neighboring layers (such as the case of the *sensors and actuators layer* and the *virtual information layer*). The *sensors and actuators layer* includes a general set of methods that store the sensorial information provided by the physical devices and that provide the commands to be applied to them. These methods may perform some kind of processing, as we will see later, and provide their outputs to the methods of the *processing layer*. In this layer, the sensorial information is processed in a general way, carrying out common signal processing tasks. In addition, the general processing of the commands is executed in this layer when required. The last layer is the *virtual information layer* where the information provided by the methods of the *processing layer* is treated and presented to the control architecture. As we can see, we are assuming a very general case where the low level sensorial information must be treated in two higher levels prior to the presentation to the control architecture. This scheme includes the simple case where the control architecture requires low level information, because "trivial" methods that simply transmit this information without processing could be present in the *processing layer* and *virtual information layer*.

Although the HABLA has been designed to be run in a single computer, the methods are independent and can execute a routine or program in a different computer by means of TCP socket communications. This way, a highly time consuming process can be run outside the HABLA computer to improve efficiency.

All of the methods present in the HABLA must be as general as possible in order to apply the HABLA to very different hardware devices or robotic platforms without changes. This is achieved by establishing a clear methodology in the creation of the methods for the *sensors and actuators layer* and the *virtual information layer*. In the case of the low level methods, we have created a protocol that must be followed by any software that controls the hardware at low level. As displayed in Figure 1, the right block that represents the hardware includes an internal part called *interface*. This element represents the methods or routines that must be programmed in the native language of the hardware device controller in order to provide sensorial information to the HABLA. For example, if the HABLA is working with a given robotic platform and we want to use a different one, we will have to program this interface layer in the new robot to communicate it with the HABLA according to our simple protocol.

In the case of the high level methods (*virtual information layer*), the HABLA is endowed with a configuration file that provides the list of active TCP sockets and the information that is provided on each. The control architecture that uses the HABLA must be reprogrammed in order to read the sensorial information or to write the commands in the appropriate socket. But, as commented before, a very important feature of the HABLA is that no limitation is imposed on the type of programming language for the control architecture.

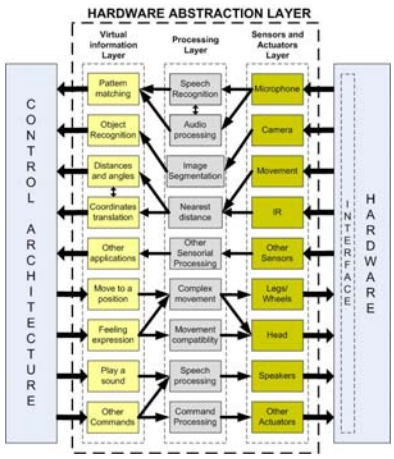


Figure 2: Diagram of the Hardware Abstraction Layer with sample methods for the case of an autonomous robot.

Figure 2 shows an example of a more detailed diagram of HABLA in the typical autonomous computational system we are dealing with, containing some of the methods that have been implemented in the HABLA at this time. In the *sensors and actuators layer* we have methods that store sensorial information from typical sensors such as microphones, cameras, infrared sensors, sonar sensors, bumpers, light sensors, GPS, motion sensors, etc. In addition, in this layer we have methods that send actuation commands, such as movements of the legs, wheels or head, or a sound to be played by the speakers, to the interface layer.

In the *processing layer* we have methods that carry out, for example, speech recognition or image segmentation and methods that can compose complex actuations or control and prevent impossible movements. These are typical processing routines, general to different robotic platforms and environmental intelligence systems. On one hand, these methods need sensorial information from the low level methods and provide information to be used by different methods in the *virtual information layer*. On the other, these methods receive data from the high level ones and execute low level methods to apply the commands. In general, the methods of this layer perform a processing function required by more that one high level method or that affect more than one actuator. For example, as represented in Figure 2, the information provided by the sonar sensors and by the infrared sensors could be combined in method that provides the distance to the *nearest object*.

The *virtual information layer* has methods that perform high level processing and present the information to the control architecture. An example of this kind of applications could be an *emotion recognition method* that provides the control architecture a string of data corresponding to a sentence or word that the user has said to the system with information about the intonation or the volume to detect the emotion in the user. This method needs information from the *speech recognition method* that provides the spoken sentence or word and information from the *audio processing method* that provides details of the physical signal in order to determine an emotion.

In Figure 2 we have represented two typical communications between methods of the same layer. For example, in the *virtual information layer* communications could take place between the method that calculates the *distance and angle* to all the objects in the vision field of the robot and the method that performs the *translation of coordinates*.

After presenting the general HABLA structure, in the next section we will try to make it clearer through robotic application examples.

3 PIONEER 2 WITH MDB

In order to show the basic operation of the HABLA in a real experiment with a real robotic platform, we have decided to reproduce the example presented in (Bellas, 2005). In this experiment we used a wheeled robot from Activmedia, the Pioneer 2 DX model, and a deliberative control architecture developed in our group called the Multilevel Darwinist Brain (MDB) and first presented in (Duro, 2000).

The MDB is a general cognitive architecture that has been designed to provide an autonomous robot with the capability of selecting the action (or sequence of actions) it must apply in its environment in order to achieve its goals. The details of the MDB are not relevant in this work and can be found in (Bellas, 2005). In the experiment presented in that paper we demonstrate the basic operation of the MDB in a real robot with a high level task. As commented before, the robot was a Pioneer 2 DX robot, a wheeled robot with a sonar array around its body and with a platform on the top in which we placed a laptop where the MDB was executed. Basically, the experiment consists on a teacher that provides commands to the robot in order to capture an object. The commands were translated into musical notes perceived by a microphone. Initially, the robot had no idea of what each command meant. After sensing the command, the robot acts and, depending on the degree of obedience, the teacher provides a reward or a punishment through a numerical value as a pain or pleasure signal introduced via keyboard.

The main objective was to show that the MDB allows the agent to create, at least, two kinds of models that come about when modeling different sets of sensors: one related to the sound sensor for the operation when the teacher is present and an induced model or models relating to the remaining sensors. The robot will have to resort to these models when the teacher is not present in order to fulfill its motivations. In this experiment, an induced behavior appears from the fact that each time the robot applies the correct action according to the teacher's commands, the distance to the object decreases. This way, once the teacher disappears, the robot can continue with the task because it developed a satisfaction model related to the remaining sensors that tells it to perform actions that reduce the distance to the object.

The execution of this experiment as explained in (Bellas, 2005) involved the programming of all the sensorial processing in the MDB. The sonar values were processed in a function to calculate the distance and angle to the object, and the audio signal perceived through the microphone was analyzed and treated in another function. The action selected by the MDB was decoded into the Pioneer ranges in another function that was programmed in the MDB. As we can see, with this basic set up we were overloading the laptop's CPU with the low level tasks and with the high level calculations (MDB).

At this point, we decided to introduce the HABLA with the set of sensors and actuators of this robot. Figure 3 shows the basic HABLA diagram particularized for this example, with the methods developed. In the *sensors and actuators layer* we have five methods according to the sensors and actuators used in the experiment. For example, the *Microphone* method reads from a socket the sound data received in the laptop's microphone and performs a basic filtering process to eliminate signal noise and provides the data to the methods of the next layer. In the *processing layer*, we have included six methods that provide typical processed information. For example, the *Nearest Distance* method receives an array of sonar values, detects the nearest one and provides this information to the *Distance and Angle* method. In the last layer (*virtual information layer*) we have programmed six high level methods. To continue with the same example, the *Distance and Angle* method calculates the angle from the information provided by the *Nearest Distance* method and sends the MDB the exact distance and angle in a fixed socket.

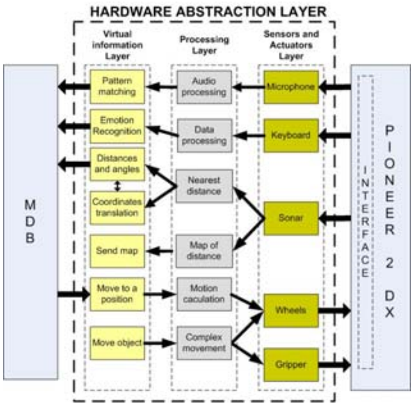


Figure 3: HABLA diagram with the particular methods of the Pioneer 2 robot.

With this basic implementation of the HABLA, we have re-run the example presented in (Bellas, 2005) obtaining the same high level result, this is, the Pioneer 2 robot autonomously obtained an induced behavior. What is important in this experiment is that, using the HABLA, the MDB doesn't have to compute low level processes. This allows us to work with a more general version of the architecture which is highly platform independent. In addition, we can execute the MDB in a much more powerful computer and use the laptop just for the HABLA and the communications.

4 AIBO WITH MDB

Once the successful operation of the MDB with a real robot has been shown, our objective is simply to repeat the experiment but using a different robot, in this case the robot is a Sony Aibo. The example is the same as in the previous case from the control architecture's point of view, but, as the robot is different, we have used a different group of sensors and actuators. In this case, the Aibo robot has to reach a pink ball it senses through a camera and the commands are spoken words provided by the teacher. In addition, the punishment or reward signal is provided by touching the back or the head of the robot, this is, using a contact sensor. The robot movements are different and it is able to speak some words through its speakers to show some emotion. In this case, the experiment was performed in a closed scenario with walls.

Figure 4 represents the HABLA with the new methods included for this robot. The philosophy behind the programming of the new methods is that they should be as general as possible in order to be useful for other robotic platforms similar, in this case, to the Aibo robot. Furthermore, we can see in Figure 4 that the previous methods developed for the Pioneer 2 robot are still present and, as we will explain later, some of them are used again.

The first thing we had to do in this experiment was to program the low level routines in the Interface layer of the Aibo robot using the Tekkotsu development framework (Touretzky, 2005). In this case, this tool follows the same idea as we use in the HAL, and all the sensorial information from the robot can be accessed by TCP sockets, so programming cost involved was very low. In the case of commands, Tekkotsu provides very simple functions to move the legs with a given gait that are accessed by sockets again.

In the sensors and actuators layer, we have included very general methods to deal with sensors such as a camera, infrared sensors, buttons or with actuators like a head or a speaker. In the processing layer we have included, as in the case of the Pioneer robot, very general processing related with the new sensors of the previous layer, such as image segmentation or speech recognition. In fact, the speech recognition method was the most important development in this experiment because this feature is not present in Tekkotsu software or in Sony's original framework. We think that owner-dog communication through spoken words is very important because it is a very intuitive way to teach this robot. In fact, the speech recognition was implemented using Sphinx-4 (Walker, 2004) which is a speech recognizer written entirely in the Java programming language. In our case, the Speech Recognition method basically executes Sphinx-4, which obtains the sound data from the microphone method, and outputs a string of data with the recognized word or phrase. In this case, we have used a reduced grammar but Sphinx includes a configuration file where more words or phrases can be added, so the method is very general. Aibo is always sending the data of the two microphones to a fixed port with a Tekkotsu behavior called "Microphone Server".

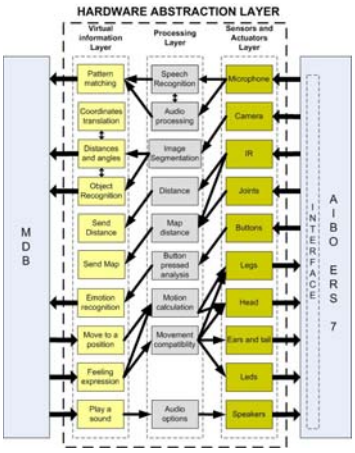


Figure 4: HABLA diagram with the particular methods of the Aibo robot

As shown in Figure 4, in the *processing layer* we find, for example, a method called *Audio Processing* that was created for the Pioneer robot, and is reused here. In the *virtual information layer* we have created more abstract methods than in the previous case, because the new sensors and actuators of this robot (like the buttons in the back or the head) permit us to create new methods such as *Emotion Recognition*, that provide information to the MDB related to the teacher's attitude.

Finally, we must point out that the execution result was successful, obtaining exactly the same behavior as in the Pioneer robot (Bellas, 2006). What is more relevant in this case is that there was no time spent in MDB reprogramming, because using the HABLA the low level processing was absolutely transparent to the control architecture. In addition, in this experiment we have executed the Tekkotsu software on the Aibo's processors, the HABLA in another computer and the MDB in a different one, optimizing this way the computational cost.

5 CONCLUSIONS

In this paper we have presented the initial implementation of the Hardware Abstraction Layer (HABLA) middleware tool. Its main features are: hardware devices independence, virtual sensing and actuation capabilities, computational cost distribution, control architecture independence, scalability and operating system independence. We have presented practical implementations of the methods in the HABLA that support two very different robotic platforms (Pioneer 2 and Aibo) in a real application example using the MDB control architecture. Currently, we are expanding the HABLA concept to different application fields, developing a practical example in an "intelligent" room.

ACKNOWLEDGEMENTS

This work was partially funded by the Ministerio de Educación y Ciencia through projects DEP2006-56158-C03-02/EQUI and DPI2006-15346-C03-01.

REFERENCES

AIBO SDE webpage, 2007: http://openr.aibo.com/

ARIA webpage, 2006: http://www.activrobots.com/SOFTWARE/aria.html

Bellas, F., Becerra, J.A., Duro, R.J., 2005. Induced behaviour in a Real Agent using the Multilevel Darwinist Brain, *LNCS*, Vol 3562, Springer, 425-434.

Bellas, F., Faiña, A., Prieto, A., Duro, R.J., 2006, Adaptive Learning Application of the MDB Evolutionary Cognitive Architecture in Physical Agents, LNAI 4095, 434-445

Duro, R. J., Santos, J., Bellas, F., Lamas, A., 2000. On Line Darwinist Cognitive Mechanism for an Artificial Organism, *Proc. supplement book SAB2000*, 215-224.

Genesereth, M.R., Nilsson, N., 1987. Logical Foundations of Artificial Intelligence, *Morgan Kauffman*.

Gerkey, B. P. , Vaughan, R. T., Howard, A., 2003. The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems, *In Proc. of the International Conference on Advanced Robotics*, 317-323.

Metta, G., Fitzpatrick, P. Natale, L., 2006, YARP: Yet Another Robot Platform, *International Journal on Advanced Robotics Systems*, 3(1):43-48

Michel, O., 2004. Webots: Professional Mobile Robot Simulation, *International Journal of Advanced Robotic Systems*, Vol. 1, Num. 1, 39-42.

Touretzky, D. S., Tira-Thompson, E.J., 2005. Tekkotsu: A framework for AIBO cognitive robotics, *Proc. of the Twentieth National Conference on Artificial Intelligence.*

Utz, H., Sablatnög, S., Enderle, S., Kraetzschmar, G, 2002. Miro - Middleware for Mobile Robot Applications, *IEEE Transactions on Robotics and Automation, Special Issue,* Vol. 18, No. 4, 493-497.

Walker, W., Lamere, P., Kwok, P, Raj, B., Singh, R., Gouvea, E., Wolf, P., Woelfel, J., 2004. Sphinx-4: A flexible open source framework for speech recognition, *Technical Report SMLI TR2004-0811, Sun Microsystems, Inc.*

ELECTRO HYDRAULIC PRE-ACTUATOR MODELLING OF AN HYDRAULIC JACK

Salazar Garcia Mahuampy, Viot Philippe
Laboratoire Matériaux Endommagement Fiabilité et Ingénierie des Procédés (LAMEFIP), France
mahuampy.sg@lamef.bordeaux.ensam.fr, philippe.viot@lamef.bordeaux.ensam.fr

Nouillant Michel
m.nouillant@iupgm.u-bordeaux1.fr

Keywords: Three-stage servo-valve, modeling, hydraulic jack.

Abstract: Before the realization of testing devices such as a high speed (5m/s) hydraulic hexapod with a high load capacity (60 tons and 6 tons for static and dynamic operating mode), the simulation is an essential step. Hence from softwares such as SimMecanics, we have performed an electro hydraulic model of the servo-valve-jack part by using parameters and recorded results with mono axis testing bench of high-speed hydraulic jack (5m/s), which has a high loading capacity (10 tons for static and 1 ton for dynamic operating mode). The high-speed jack is provided by two parallel three-stage servo-valves. Each three-stage servo-valve supplies 600L/mm. Therefore the unit allows us to obtain a realistic model of an extrapolated hexapod from the mono axis currently used. The aim of this article is to provide a modeling of the second and third stage servo valves by comparison of the typical experimental reading and the computed curves obtained from simulation.

1 INTRODUCTION

The main difficulties of the modeling of an actuator hydraulic hexapod are servo valve models. In the case of electrical actuators operating in control voltage, the torque is got from the control voltage and the rotation speed of the motor. In the case of hydraulic actuator, for a current control of servo valve given, the efforts provided by the jack depend on the flow rate and consequently they depend also on the speed of the jack and leakages. This aspect is not taken into account by using the classical frequency model of the servo valve (Faisandier, 1999), (Mare, 2002), (Thayer, 1965).

At first, we will present the hydraulic model of the servo valve + jack system. In a second time, a nonlinear modeling will be developed. Lastly, the results obtained from the modeling will be presented and analyzed.

2 ELECTRO HYDRAULIC SYSTEM

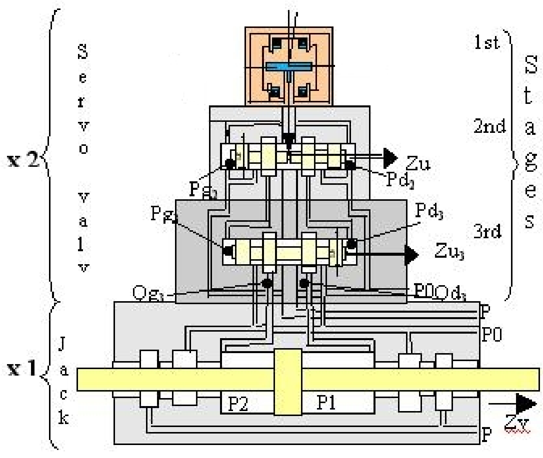


Figure 1: Diagrammatic section of system.

Figure 1 shows a model with three-stage servo valve and a jack. The symbol "x2" and "x1" indicate respectively the presence in the system of two parallel servo valves supplying the jack. A servo valve is controlled by an electrical stage (1st stage), followed by a 2nd stage and a 3rd hydraulic amplification stage (Faisandier,1999), (Guillon, 1961).

A control current applied to the system input is named i. The double potentiometric hydraulic divisor of the first stage, leads to two pressures Pg_2 and Pd_2 applied to the end of the slide on the 2nd stage, the flow rates provided by this 2nd stage are named Qg_2 and Qd_2, applied to the control of the slide displacement of the 3rd stage of the servo valve.

The third stage generates the flow rates Qg_3 and Qd_3 taking part in the sum of the flows entering and going out of the jack chambers, and leading, through instantaneous volumes of the chambers, and the compressibility coefficient of the oil, to the pressure variation in each chamber. The servo valve is supplied with pressure P_0 and reservoir return line P.

3 MODELING

3.1 Servo Valve Linear Model

Figure 2 shows linear diagram of a servo valve modeling (Pommier,2000).

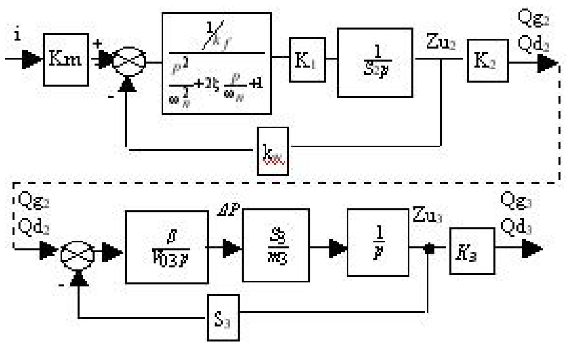


Figure 2: Linear diagram of the servo valve.

Km: Gain between the current and the torque.
ωn: Undamped natural frequency of the pallet inertia of the 1st stage.
ξ : Damping ratio of the friction of the pallet.
kf: Gain displacement in torque of the nozzle pallet unit.
K_1: Gain in flow rate of hydraulic amplification.
S_2: Surface of the slide ends of the 2nd stage.
k_w: Gain of the feedback torque.
K'_2: Gain in flow rate of the 2nd stage slide.
V_{03}: Effective volume when the slide is centered.
S_3 : Surface of the slide ends of the 3rd stage.
m_3: Slide mass of the 3rd stage.
K'_3: Gain in flow rate of the 3rd stage.

One notes that for this model, the K'_2 and K'_3 coefficients characterize the flow rate as a function of the slide positions. For the second stage, the K'_2 coefficient gives a good approximation within a large part of the operating area, contrary to the K'3 coefficient suggesting that the speed of the jack does not depend on the load.

3.2 Hydraulic Nonlinear Model

Figure 3 shows the general diagram describing the non-linear model of servo-valve-jack system.

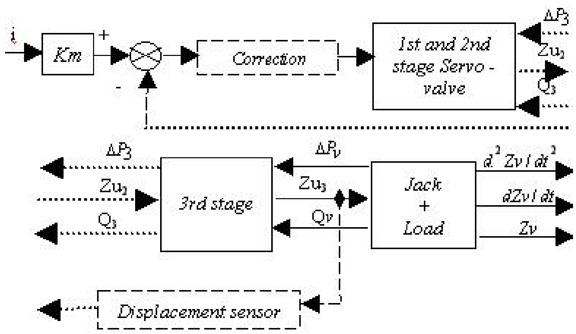


Figure 3: Nonlinear model of the servo valve + jack.

One takes into account the hydrodynamic forces applied to the various slides of the servo valves. These forces take part in the nonlinear behavior of the device. This functional diagram shows that the behavior of the servo valve and the jack cannot be dissociated and must be treated as such.

We have developed a nonlinear model of the servo-valve-jack unit allowing to determine, for a given control current, the effort supplied by the jack from the pressure variations within the control volumes formed by the jack chambers. This pressure variation after temporal integration defines the pressure within the right and left chambers of the jack. The difference of these pressures multiplied by the active section of the piston gives the effort supplied by the jack.

In our case, the main difficulty is the nonlinear modeling of the servo valve; more particularly we have to take into account a finite number of sensitive parameters and the hydraulic nonlinear behavior laws. In addition to the nonlinearities resulting from the hydraulic potentiometer, the pressure flow will be taken into account from the following relation:

$$Q = K\sqrt{|\Delta P|}. \tag{1}$$

We assume that the flow rate of the fluid is viscous and incompressible, as turbulent type.

The transient flow rate associated with fluid incompressibility is proportional to the rate of change of pressure in a volume of control and may be expressed as:

$$(V/\beta)(dP/dt)+(dV/dt)=Q_e-Q_s \tag{2}$$

Where β is the bulk modulus of the fluid.

We also take into account the hydrodynamic forces on the various slides of the servo-valves, leading to the nonlinear behavior of the device.

$$F_{hd} = Q\sqrt{2(\gamma/g)} \tag{3}$$

Where γ is the specific gravity (kg/m3) and g is the gravitational acceleration.

The angular displacement of the frame engine versus the current is given by:

$$iKm+l_s\Delta P_2-Kr_2LZu_2=J\ddot{\theta}+\varphi\dot{\theta}+Kr_2L^2\theta+F_s sign(\dot{\theta}) \tag{4}$$

With ΔP2=Pg2-Pd2 difference of pressure between the two slide ends of the 2nd stage.From the equation (2) one can obtain the evolution laws of the pressures Pg2 and Pd2.

$$\frac{dPg_2}{dt}=\frac{\beta}{V_{02}+S_2Zu_2}\left(Qg_1-Q_b+Kf_2\sqrt{|P_0-Pg_2|}-S_2\dot{Zu_2}\right) \tag{5}$$

$$\frac{dPd_2}{dt}=\frac{\beta}{V_{02}-S_2Zu_2}\left(Qd_1-Q_b+Kf_2\sqrt{|P_0-Pd_2|}+S_2\dot{Zu_2}\right) \tag{6}$$

Where, S_2Z_{u2} is the flow rate caused by the motion of the slide and Qg_1 and Qd_1 are the flow rates resulted from the pressure applied to the section of fixed openings *S1*, *Qb* is the flow rate from the nozzle tip.

By applying the fundamental principle of dynamics we obtain the sum of the forces applied to the slide of the 2nd stage:

$$S_2\Delta P_2-Kr_2L\theta-F_{hd2}=m_2\ddot{Zu_2}+\psi_2\dot{Zu_2}+Kr_2Zu_2+F_s sign(\dot{\theta}) \tag{7}$$

Where S2ΔP2 is the difference of forces applied to the ends of the slide, Kr2Lθ is the force due to the stiffness and the deformation of the pallet.

The flow rate Qd2 and Qg2 provided by the 2nd stage is given by the equation (1). The equation of Qd2, Qg2 taking into account the slide covering and the resulting leakage. The modeling of the slide covering is exponential modeling, where ε is a constant. The very low value of ε ensures the continuity in the opening model of the slide and the leakage resulted from the slide covering.

From the equation (2), and (7), one can obtain the evolution laws of the pressures and the sum of the forces related to the third stage and the jack hydraulic.

4 IMPLEMENTATION AND RESULTS

At first, we have identified the parameters of the nonlinear model from the high-speed 5m/s servo-valve-jack unit of LAMEFIP laboratory. This first step gives us some experimental reference parameters for the validity of our servo-valve model.

The servo valve composed of the first and second stages and the second servo valve composed of the third stage are respectively hydaustar 550 and 1160 type servo valve. All the parameter values of the model estimated or measured of the electro hydraulic system for the second and third stages are summarized in table 1.

Table 1: Parameter values of the second and third stages.

Parameters		Manufacturer data	Estimated data
L	Length magnetic pallet	31.2e-3 (m)	-----------
l	Outdistance between magnetic axis pallets and metering jets	10.5e-3 (m)	------------
Dbuse	Diameter of nozzle tip	0.18e-3 (m)	----------
Sbuse	Surface of nozzle tip	2.54e-8 (m^2)	
xo	Length between magnetic axis pallets and metering jets	----------	55e-6 (m)
Kbuse	Gain of nozzle tip	----------	0.039 (m^3 /s/m)
K1	Gain of flow rate of the fixed section	----------	1.5e-9 (m^3 /s/m)
Kf_2	Gain of leakage in the stage	----------	5.15e-12 (m^3 /s/m)
Vo_2	Volume control when the slide is centered	19.5e-9 (m^3)	-----------
S_2	Surface of slide	3.38e-5 (m^2)	-----------
J	Pallet inertia	----------	4e-7 (Kg m^2)

Φ	Viscous friction coefficient in rotation of the pallet	----------	9e-4 (Nm/rd s)
K_{r2}	Stiffness of the slide of the stage	2100 (N/m)	------------
M_2	Slide masse	9e-3 (Kg)	------------
Ψ_2	Viscous friction coefficient in slide	--------	6 (Ns/m)
Kf_3	Gain of leakage in the stage	-----------	1e-12 (m^3 /s/m)
E	Lap	-----------	3.7e-6 (m)
K_2	Gain of flow rate of volume control in the stage	-----------	6e-4 (m^3 /s/m)
S_3	Surface of slide	5e-4 (m^2)	------------
Vo_3	Volume control when the slide is centered	8.6e-6 (m^3)	------------
M_3	Slide masse	276e-3 (Kg)	------------
Ψ_3	Viscous friction coefficient in slide	-----------	1000 (Ns/m)

After completing the estimated parameters, we compare the typical experimental datas supplied by the manufacturer such as the flow rate – current characteristics for the second stage, the frequency response of the second and third stages with the curves resulting from the model using both the estimated values and the measured values.

4.1 Flow Rate – Current Characteristics

We have compared the flow rate *Qd2* of the second stage of the servo valve under differential pressure of 70 bar when the current i varies in the range [$-I_{max}$; $+I_{max}$], I_{max} is the maximum value of the current modulus. The figure 4 shows the flow rate Qd2 – current i characteristics for the 550 Hydraustar servo valve (second stage) supplied by the manufacturer and those obtained from the model.

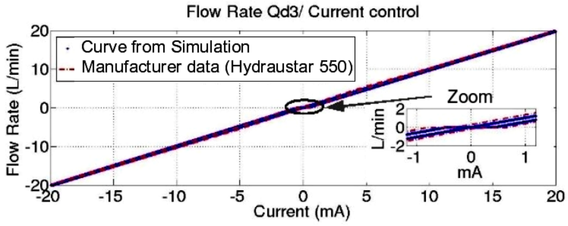


Figure 4: Flow rates / control currents characteristics supplied by manufacturer and those obtained from the model.

As shown in the figure 4, we can observe that the result obtained from the model and that provided by the manufacturer are similar when the current input reaches the maximal value of 20 mA. The maximal flow rate value provided by the driver servo valve reaches 20 l/min. One can observe the role of the leakages when the slide position is near the hydraulic zero as shown in the inset of the figure4. These leakages result from the taking into account of the laps and the clearances between the slide and the sleeve. The figure 5 shows the computed flow rate *Qd3* of the third stage of the servo valve under differential pressure of 70 bar when the courant varies from $-I_{max}$ and + I_{max}. We obtain in this particular configuration, a maximal computed value of the flow rate provided by the servo valve of +/- 600 l/min. This value matches the manufacturer data for this servo valve model.

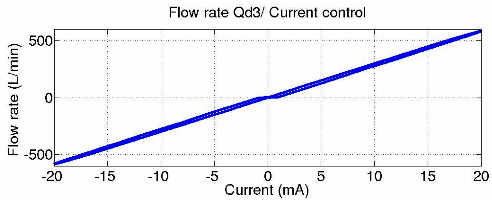


Figure 5: Flow rate Qd3/ control current characteristics obtained by simulation.

4.2 Frequency Response

We have compared the characteristic curves obtained from the bench test measurement (by using a servo valve flow with normalized decreasing of the pressure of 70 bar for different input levels) with the curves computed from our model.

4.2.1 First and Second Stage

The figure 6 shows the comparison of the manufacturer curves with those obtained from the model for the similar operating conditions: provided pressure Ps = 210 bar, differential pressure 70 bar, 100% nominal current.

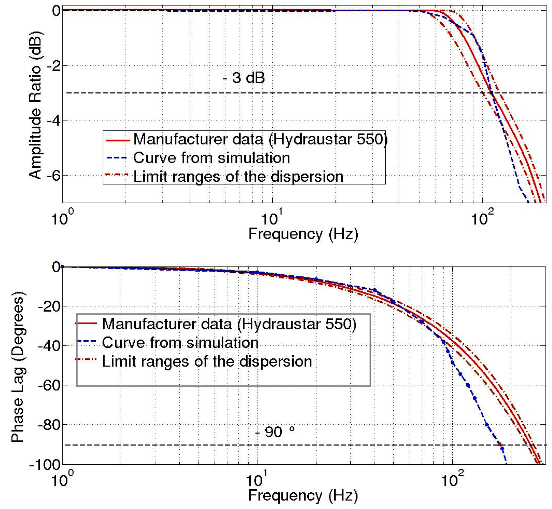


Figure 6: Frequency response for 100% of the nominal current. (for -3 dB, 110 Hz < f < 130 Hz, -90°, 230 Hz < f < 250 Hz, theoretical curve).

The frequency response corresponding to 100 % of nominal current obtained from the model gives a cut-off frequency of 115 Hz for a gain of –3 dB and a frequency of 180 Hz for a phase lag of 90°. The cut-off frequency for –3 dB obtained with the model is in the limit range of the dispersion of frequencies provided by the manufacturer. For phase lag of 90°, the frequency error between both curves is estimated from 20% to 28 %.

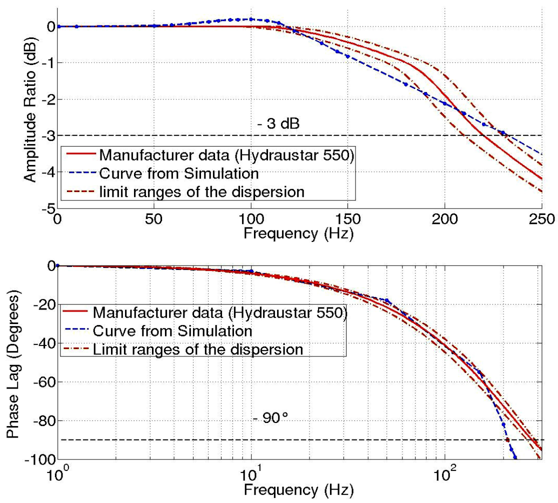


Figure 7: Frequency response for 25 % of the nominal current. (for –3 dB, 210 Hz < f < 230 Hz and for -90°, 250 Hz < f < 280 Hz, theoretical curve).

The figure 7 shows the frequency comparison of the manufacturer curves and those obtained with the model for the similar operating conditions: Provided pressure Ps = 210 bar, pressure difference 70 bar, 25 % of the nominal current.

The frequency response for 25% of the nominal current obtained from the model gives us a cut-off frequency of 220 Hz for a gain of -3 dB and a frequency of 210 Hz for a phase lag of 90°. The cut-off frequency corresponding to a gain of -3 dB is in the limit range of the dispersion of the manufacturer frequencies. The frequencies related to the phase lag of 90° obtained by modeling and those provided by manufacturer are different. The error is estimated from 16 % to 25 %. The computed curve shapes for the second stage are different from those supplied by the manufacturer. This difference should depend directly on the estimated values of the pallet inertia and its friction coefficient. An infinitesimal variation of these values leads to the under damping observed in amplitude plot of the bode diagram.

4.2.2 Third Stage

We compare the frequency responses of the serial servo valve 1160, corresponding to the third stage, with the coupling of the serial servo valve 550 related to the driving stage. The figure 8 shows the typical response supplied by the manufacturer for 100 % of the nominal input signal.

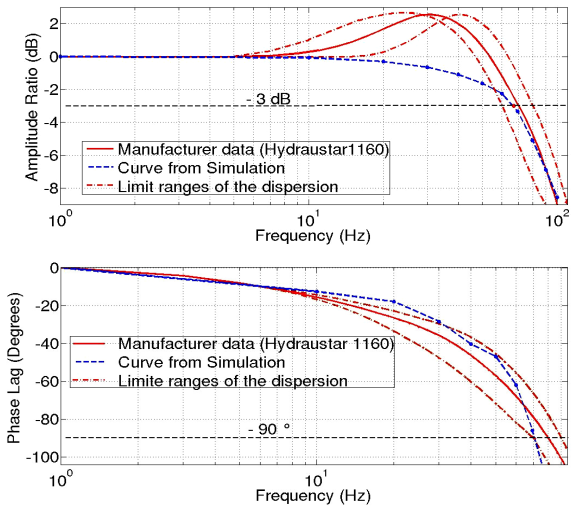


Figure 8: Frequency response for 100 % of the nominal current.

The driving servo valve is provided with a pressure Ps of 210 bar. The cut-off frequencies in the limit ranges of the dispersion for a frequency response corresponding to 100 % of the nominal current match respectively 70 Hz and 85 Hz for –3 dB and phase lag 90° for flow rates Qd3 and Qg3 of 600 l/min.

The frequency response for 100 % of the nominal value of the current obtained from the model gives a cut-off frequency of 70 Hz corresponding to a gain of –3 dB and a frequency of 73 Hz for a phase lag of 90°. One observes that the results obtained for the servo valves corresponding to the second and third stages from the model and those supplied by the manufacturer are close. Nevertheless, one can note that a difference between the curves amplitude from the manufacturer and those obtained from the model. Indeed, the manufacturer curves show an under damping probably caused by the implementation of the corrector in the system whereas we have performed the modeling without corrections in loop control.

5 CONCLUSION

In our work, one can note that the servo valve is the limiting element of the servo valve + jack system. The flow rate values (20 l/min and 600 l/min provided by the second and the third stage respectively), like the bandwidth values of the driving servo valve corresponding to the second stage and those of the amplification stage e.g. the third stage, obtained from the model and supplied by manufacturer are rather similar. These results suggest that the estimated values, which cannot be measured, such as the gap *xo* between the nozzle and the pallet, the inertia *J* and the viscous friction coefficients of the pallet Φ, the friction coefficients of the slides Ψ_2, Ψ_3 and the flow rate gains *Kbuse, K1,* Kf_2, *K2*, Kf_3, are fairly close to the physical values of the type 550 and 1160 Hydraustar servo valve. We have shown that the nonlinear model presented in this work have allowed us the accurate simulation of the nonlinear behavior of three stage servo valves between the current input and the flow rate output. This model allows the taking into account of the pressure into the jack chambers as a function of the forced stress. Hence with this model is possible to computed the dynamic and static behavior, the latter corresponds to the short circuit generated by jack stoppers.

At this stage, the interfacing with the SimMecanics software, by introducing as input this corresponding to the effort between two ”bodies”, the quantity *(P1-P2)Sp* (*Sp*: useful piston surface), will provide as output from the software, the speed and the relative position of both bodies.

ACKNOWLEDGEMENTS

Special thanks go to Mr. Terrade (Hydraustar company) for supplying technical support.

REFERENCES

Faisandier, J., 1999. *Mécanismes hydrauliques et pneumatiques* , 8e édition, Paris, Dunod.
Mare, J.C., 2002, *Actionneurs Hydrauliques,* Commande . Techniques de l'ingénieurs, traité l'informatique industrielle.S731.
Thayer, W.J., 1965, *Transfer functions for moog servo valves* , Technical bulletin I03, MOOG INC. Springer-Verlag.
Guillon, M., 1961, *Etude et détermination des systèmes hydrauliques*, Paris, Dunod.
Pommier, V., Lanusse P., Oustaloup A., 2000, *Commande CRONE d'un actionneur hydraulique*. In Journée Franco-Tunisiennes, Monastir, Tunisie.

A SET APPROACH TO THE SIMULTANEOUS LOCALIZATION AND MAP BUILDING
Application to Underwater Robots

Luc Jaulin[1], Frédéric Dabe[2], Alain Bertholom[2] and Michel Legris[1]
[1]*E3I2, ENSIETA, 2 rue Franois Verny, 29200 Brest*
[2]*GESMA, Brest*
jaulinlu@ensieta.fr

Keywords: Bounded-error, constraint propagation, interval analysis, SLAM, state estimation, submarine robots, robotics.

Abstract: This paper proposes a set approach for the simultaneous localization and mapping (SLAM) in a submarine context. It shows that this problem can be cast into a constraint satisfaction problem which can be solve efficiently using interval analysis and propagation algorithms. The efficiency of the resulting propagation method is illustrated on the localization of submarine robot, named *Redermor*. The experiments have been collected by the GESMA (Groupe d'Etude Sous-Marine de l'Atlantique) in the Douarnenez Bay, in Brittany.

1 INTRODUCTION

This paper deals with the *simultaneous localization and map building* problem (SLAM) in a submarine context (see (Leonard and Durrant-Whyte, 1992) for the general SLAM problem). A set membership approach for SLAM (see e.g., (Marcoet al., 2000)) will be considered and it will be shown that this approach leads us to a *constraints satisfaction problem (CSP)* (see e.g., (Jaulin et al., 2001) for notions related CSP and applications) which can be solved efficiently using interval constraints propagation. The efficiency of the approach is illustrated on an experiment where an actual underwater vehicle is involved. In this problem, we try to find an envelope for the trajectory of the robot and to compute sets which contains some detected sea marks (such as mines).

Set-membership methods have often been considered for the localization of robots (see, e.g., (Meizel et al., 1996), (Halbwachs and Meizel, 1996), in the case where the problem is linear). In situations where strong nonlinearities are involved, interval analysis has been shown to be particularly usefull (see e.g. (Meizel et al., 2002), where one of the first localization of an actual robot has been solved with interval methods). Interval analysis has been shown to be efficient in several SLAM applications (see (Drocourt et al., 2005) and (Porta, 2005) where interval analysis has already been used in the context of SLAM for wheeled robots). But the approach is here made more efficient by the addition of constraint propagation techniques. Note that there exist many other robotics applications where interval constraint propagation methods have been successful (see, *e.g.*, (Baguenard, 2005) for the calibration of robots, (Raissi et al., 2004) for state estimation, (Gning and Bonnifait, 2006) for dynamic localization of robots, (Lydoire and Poignet, 2003), (Vinas et al., 2006) for control of robots, (Delanoue et al., 2006) for topology analysis of configuration spaces, ...).

2 ROBOT

The robot to be considered in our application is an autonomous underwater vehicle (AUV), named *Redermor* (see Figure 1). This robot, developed by GESMA (Groupe d'Etude Sous-Marine de l'Atlantique), has a length of 6 m, a diameter of 1 m and a weight of 3800 Kg. It has powerful propulsion and control system able to provide hovering capabilities. The main purpose of the *Redermor* is to evaluate improved navigation by the use of sonar information. It is equipped with a KLEIN 5400 side scan sonar which makes it possible to localize objects such as rocks or mines. It also encloses other sophisticated sensors such as a Lock-Doppler to estimate its speed and a gyrocom-

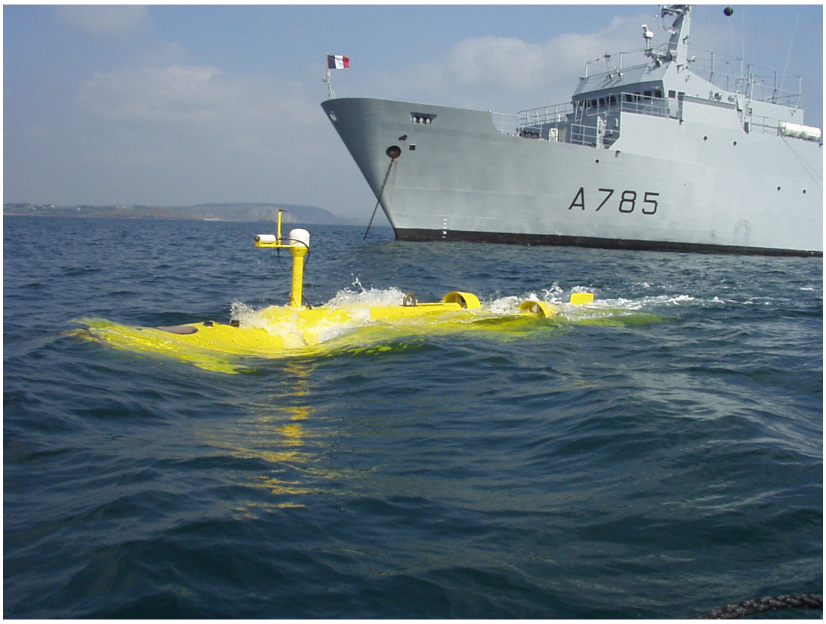


Figure 1: The Redermor inside the water and the boat from which it has been dropped.

pass to get its Euler angles.

3 METHOD

In the graphSLAM approach (Thrun and Montemerlo, 2005), a criterion is built by taking all constraints into account. Then, a local minimization method, such as conjugate gradient, is used to find a good solution of the SLAM problem. Here, we adopt a similar approach, but instead of building a criterion, we cast the SLAM problem into a huge constraints satisfaction problem (CSP). For our problem, these constraints are given below.

$$\left\{\begin{array}{l}
t \in \{6000.0, 6000.1, 6000.2, \ldots, 11999.4\}, i \in \{0, 1, \ldots, 11\}, \\
\begin{pmatrix} p_x(t) \\ p_y(t) \end{pmatrix} = 111120. \begin{pmatrix} 0 & 1 \\ \cos\left(\ell_y(t) * \frac{\pi}{180}\right) & 0 \end{pmatrix} \begin{pmatrix} \ell_x(t) - \ell_x^0 \\ \ell_y(t) - \ell_y^0 \end{pmatrix}, \\
\mathbf{R}(t) = \begin{pmatrix} \cos\psi_t & -\sin\psi_t & 0 \\ \sin\psi_t & \cos\psi_t & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta_t & 0 & \sin\theta_t \\ 0 & 1 & 0 \\ -\sin\theta_t & 0 & \cos\theta_t \end{pmatrix} \\
\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi_t & -\sin\varphi_t \\ 0 & \sin\varphi_t & \cos\varphi_t \end{pmatrix}, \\
\mathbf{p}(t) = (p_x(t), p_y(t), p_z(t)), \quad \mathbf{p}(t+0.1) = \mathbf{p}(t) + 0.1 * \mathbf{R}(t).\mathbf{v}_r(t), \\
||\mathbf{m}(\sigma(i)) - \mathbf{p}(\tau(i))|| = r(i), \\
\mathbf{R}^{\mathrm{T}}(\tau(i))\,(\mathbf{m}(\sigma(i)) - \mathbf{p}(\tau(i))) \in [0,0] \times [0,\infty] \times [0,\infty], \\
m_z(\sigma(i)) - p_z(\tau(i)) - a(\tau(i)) \in [-0.5, 0.5]
\end{array}\right.$$

In these constraints, $\mathbf{p} = (p_x, p_y, p_z)$ denotes center of the robot, (ψ, θ, φ) denote the Euler angles of the robot, $\sigma(i)$ is the number of the ith detected object, $\tau(i)$ is the time corresponding the ith detection and $\mathbf{m}(j)$ is the location of the jth object. From this CSP, a constraint propagation procedure (see, e.g., (Jaulin et al., 2001)) can thus be used to contract all domains for the variables without loosing a single solution.

4 RESULTS

A constraints propagation procedure has been used to contract all domains of our CSP. The results obtained are represented on Figure 2. Subfigure (a) represents a punctual estimation of the trajectory of the robot. This estimation has been obtained by integrating the state equations from the initial point (represented on lower part). We have also represented the 6 objects that have been dropped at the bottom of the ocean during the experiments. Subfigure (b) represents an envelope of the trajectory obtained using an interval integration, from a small initial box, obtained by the GPS at the beginning of the mission. In Subfigure (c) a final GPS point has also been considered and a forward-backward propagation has been performed up to equilibrium. In Figure (d) the constraints involving the object have been considered for the propagation. The envelope is now thinner and enveloping boxes containing the objects have also been obtained (see Subfigure (e)). We have checked that the unknown actual positions for the objects (that have been measured independently during the experiments) all belong to the associated box, painted black. In Subfigure (f), a zooming perspective of the trajectory and the enveloping boxes for the detected objects have been represented. The computing time to get all these envelopes in less than one minute with a Pentium III. About ten forward-backward interval propagations have been performed to get the steady box of the CSP.

In the case where the position of the marks is approximately known, the SLAM problem translates into a state estimation problem. The envelope for the trajectory becomes very thin and a short computation time is needed. The capabilities of interval propagation methods for state estimation in a bounded error context have already been demonstrated in several applications (see e.g., (Gning, 2006), (Bouron, 2002), (Baguenard, 2005), (Gning and Bonnifait, 2006), (Jaulin et al., 2001)).

Figure 3 contains the reconstructed waterfall (above) and one zoom (below). Each line corresponds to one of the six seamarks ($i = 0, \ldots, 5$) that have been detected. The gray areas contains the set of all feasible pairs $(t, \|\mathbf{p} - \mathbf{m}_i\|)$, associated to the *migration hyperbola*. The twelve small black disks correspond to the detected marks. From each disk, we can get the time t at which the mark has been detected (x-axis), the number of the mark (corresponding to the line), and the distance r_i between the robot and the mark (y-axis). Black areas correspond to all feasible (t, r_i). Some of these areas are tiny and are covered by a black disk. Some are larger and do not contain any black disk. In such a case, an existing mark may have been missed by the operator during the scrolling of the waterfall. As a consequence, with the help of Figure 3, the operator could scan again the waterfall and find undetected marks much more efficiently. If the operator (which could be a human or a program) is able to detect at least one more mark, then, the propagation algorithm could be thrown once more to get a thinner envelope for the trajectory, thinner black areas in the reconstructed waterfall and thus a higher probability to detect new marks on the waterfall, ... The operator can thus be seen as a contractor ((Jaulin et al., 2001)) inside a constraint propagation process.

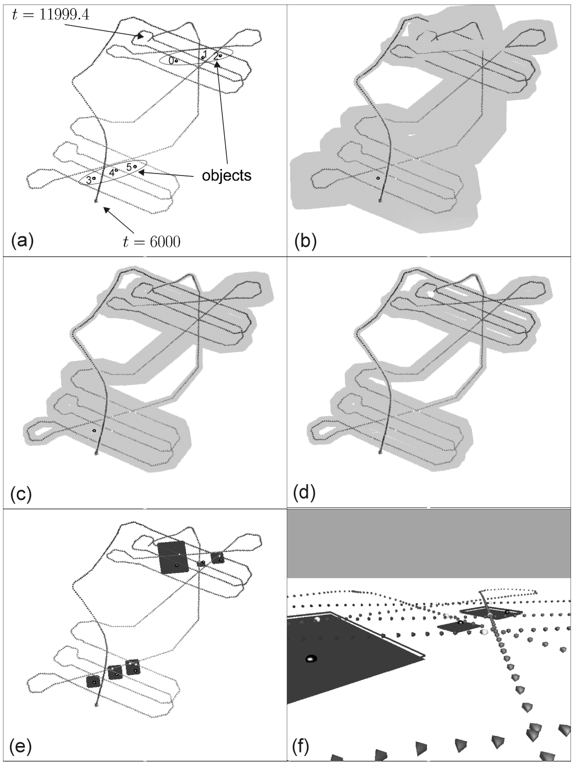


Figure 2: Results obtained by the interval constraint propagation.

5 CONCLUSION

In this paper, we have shown that interval constraints propagation could be applied to solve efficiently SLAM problems. The approach has been demonstrated on an experiment made with an actual underwater robot (the *Redermor*). The experiment lasted two hours and involved thousands of data. If all assumptions on the bounds of the sensors, on the flat bottom, on the model of the robot, ... are satisfied, then their exists always at least one solution of our problem: that corresponding to the actual trajectory of the robot.

When outliers occur during the experiment, our approach is not reliable anymore and one should take care about any false interpretation of the results. Consider now three different situation that should be known by any user of our approach for SLAM.

Situation 1. The solution set is empty and an empty set is returned by the propagation procedure. Our approach detects that their exists at least one outlier but it is not able to return any estimation of the trajectory and the positions of the marks. It is also not able to detect which sensor is responsible for the failure.

Situation 2. The solution set is empty but nonempty thin intervals for the variables are returned by the propagation. Our approach is not efficient enough to detect that outliers exist and we can wrongly interpret that an accurate and guaranteed estimation of the trajectory of the robot has been done. Other more efficient algorithms could be able to prove that no solution exists which would lead us to the situation 1.

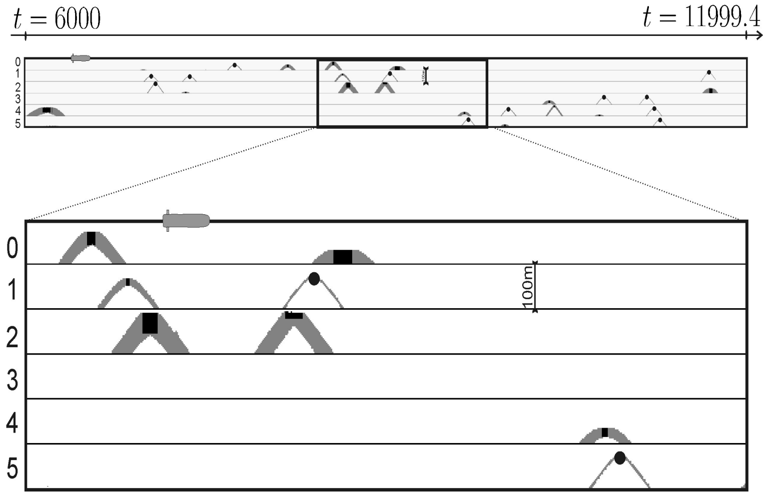


Figure 3: The reconstructed waterfalls can help to find undetected marks.

Situation 3. The solution set is not empty but it does not contain the actual trajectory of the robot. No method could be able to prove that outliers occur. Again, our approach could lead us to the false conclusion that a guaranteed estimation of the trajectory of the robot has been done, whereas, the robot might be somewhere else.

Now, for our experiment made on the Redermor, it is clear that outliers might be present. We have observed that when we corrupt some data voluntarily (to create outliers), the propagation method usually returns rapidly that no solution exists for our set of constraints. For our experiment, with the data collected, we did not obtain an empty set. The only thing that we can conclude is that if no outlier exist (which cannot be guaranteed), then the provided envelope contains the actual trajectory for the robot.

REFERENCES

X. Baguenard. Propagation de contraintes sur les intervalles. Application à l'étalonnage des robots. PhD dissertation, Université d'Angers, Angers, France 2005. Available at: www.istia.univ-angers.fr/~baguenar/.

P. Bouron. Méthodes ensemblistes pour le diagnostic, l'estimation d'état et la fusion de données temporelles. PhD dissertation, Université de Compiègne, Compiègne, France, (Juillet 2002).

N. Delanoue, L. Jaulin and B. Cottenceau. Using interval arithmetic to prove that a set is path-connected. Theoretical Computer Science, Special issue: Real Numbers and Computers 351(1), 119–128, 2006.

C. Drocourt, L. Delahoche, B. M. E. Brassart and A. Clerentin. Incremental construction of the robot's environmental map using interval analysis. Global Optimization and Constraint Satisfaction: Second International Workshop, COCOS 2003 3478, 127–141 2005.

A. Gning. Localisation garantie d'automobiles. Contribution aux techniques de satisfaction de contraintes sur les intervalles. PhD dissertation, Université de Technologie de Compiègne, Compiègne, France 2006.

A. Gning and P. Bonnifait. Constraints propagation techniques on intervals for a guaranteed localization using redundant data. Automatica 42(7), 1167–1175 (2006).

E. Halbwachs and D. Meizel. Bounded-error estimation for mobile vehicule localization. CESA'96 IMACS Multiconference (Symposium on Modelling, Analysis and Simulation) pp. 1005–1010 (1996).

L. Jaulin, M. Kieffer, O. Didrit, E. Walter. Applied Interval Analysis, with Examples in Parameter and State Estimation, Robust Control and Robotics. Springer-Verlag, London (2001).

J. J. Leonard and H. F. Durrant-Whyte. Dynamic map building for an autonomous mobile robot. International Journal of Robotics Research 11(4) (1992).

F. Lydoire and P. Poignet. Nonlinear predictive control using constraint satisfaction. In 2nd International Workshop on Global Constrained Optimization and Constraint Satisfaction (COCOS), pp. 179–188 (2003).

M. D. Marco, A. Garulli, S. Lacroix and A. Vicino. A set theoretic approach to the simultaneous localization and map building problem. In Proceedings of the 39th IEEE Conference on Decision and Control, vol. 1, pp. 833–838, Sydney, Australia (2000).

D. Meizel, O. Lévêque, L. Jaulin and E. Walter. Initial localization by set inversion. IEEE transactions on robotics and Automation 18(6), 966–971 (2002).

D. Meizel, A. Preciado-Ruiz and E. Halbwachs. Estimation of mobile robot localization: geometric approaches. In M. Milanese, J. Norton, H. Piet-Lahanier and E. Walter, editors, Bounding Approaches to System Identification, pp. 463–489. Plenum Press, New York, NY (1996).

J. Porta. Cuikslam: A kinematics-based approach to slam. In Proceedings of the 2005 IEEE International Conference on Robotics and Automation, pp. 2436–2442, Barcelona (Spain) (2005).

T. Raissi, N. Ramdani and Y. Candau. Set membership state and parameter estimation for systems described by nonlinear differential equations. Automatica 40, 1771–1777 (2004).

S. Thrun and M. Montemerlo. The GraphSLAM algorithm with applications to large-scale mapping of urban structures. International Journal on Robotics Research 25(5/6), 403–430 (2005).

P. H. Vinas, M. A. Sainz, J. Vehi and L. Jaulin. Quantified set inversion algorithm with applications to control. Reliable computing 11(5), 369–382 (2006).

TRAJECTORY PLANNING USING OSCILLATORY CHIRP FUNCTIONS APPLIED TO BIPEDAL LOCOMOTION

Fernando Juan Berenguer Císcar
Escuela Superior Politécnica, Universidad Europea de Madrid,C/ Tajo s/n, Villaviciosa de Odón, Spain
fjuan.berenguer@uem.es

Félix Monasterio-Huelin Maciá
E.T.S.I. de Telecomunicación,Universidad Politécnica de Madrid, C/ Ciudad Universitaria s/n, Madrid, Spain
felix.monasteriohuelin@upm.es

Keywords: Trajectory planning, bipedal walking robot, chirp functions.

Abstract: This work presents a method for planning sinusoidal trajectories for an actuated joint, so that the oscillation frequency follows linear profiles, like trapezoidal ones, defined by the user or by a high level planner. The planning method adds a cubic polynomial function for the last segment of the trajectory in order to reach a desired final position of the joint. We apply this planning method to an underactuated bipedal mechanism which gait is generated by the oscillatory movement of its tail. Using linear frequency profiles allow us to modify the speed of the mechanism and to study the efficiency of the system at different speed values.

1 INTRODUCTION

Trapezoidal and linear velocity profiles are widely used in trajectory planning for mobile robots and robot manipulators. The necessary procedure for defining this kind of trajectories can be found in many robotic textbooks (Spong, 1989, Sciavicco, 1996, Craig, 2006). In other robotic areas, as in the case of walking machines and nonholonomic locomotion systems, the robot joints execute oscillatory motions, and some planning methods are based on using sinusoidal trajectories with constant frequencies (Morimoto, 2006, Sfakiotakis, 2006, Murray, 1993). In (Berenguer, 2006) we presented an underactuated bipedal mechanism that is able to walk using only one actuator that moves a tail following a sinusoidal trajectory. The displacement velocity of these systems depends on the oscillation frequencies together with other parameters. The planning method presented here provides continuous sinusoidal joint trajectories that follow desired piecewise-linear frequency profiles and generate smooth variation of the systems speed.

On the other hand, swept sinusoids (chirps) are usually used for identifying and modelling actuators and mechanisms (McClung, 2004, Leavitt, 2006). In this work, we estimate the optimal stride frequency of a biped by means of analyzing the step length at different frequencies, during the execution of a trajectory generated by the proposed planning method.

This paper is organized as follows. Next section presents an initial example problem and shows perhaps a common beginner's mistake in the way of solving this problem. The correct problem's solution is also provided in this section. Section III sets out the planning problem in a general form and presents the proposed solution method. Section IV shows the application of this method to a bipedal model and how we can use the results for analyzing the efficiency of the model at different speeds. Finally, section V presents conclusions and future work.

2 AN INITIAL EXAMPLE

Suppose we want to generate a sinusoidal trajectory with unit amplitude for a robot joint. The joint is initially at its central position (0rad) with zero velocity, and will start to oscillate with an increasing frequency. The joint must reach a frequency of πrad/s at instant t=10s and keep this value during 20 seconds. Finally, the joint must reduce its frequency to zero and achieve the central position again at instant t=40s. Figure 1 shows this trapezoidal frequency profile and a continuous joint trajectory q(t) that follows it and finishes at zero position.

As a first solution to this problem, a beginner might propose the expression in (1) for the joint trajectory q(t). This is a sine function and we can check that the function ω(t) follows the trapezoidal profile.

$$q(t) = \sin(\omega(t)t) \quad \text{where} \quad \omega(t) = \begin{cases} \pi\dfrac{t}{10} & 0 \le t < 10 \\ \pi & 10 \le t < 30 \\ \pi\dfrac{(40-t)}{10} & 30 < t \le 40 \end{cases} \tag{1}$$

Figure 2 shows this function (1) and of course this is not the correct solution. We can see the reason for this result by analyzing the time derivative $\dot{q}(t)$ given by (2).

$$\dot{q}(t) = \begin{cases} 2\pi\dfrac{t}{10}\cos(\pi\dfrac{t^2}{10}) & 0 \le t < 10 \\ \pi\cos(\pi t) & 10 \le t < 30 \\ \pi\dfrac{(40-2t)}{10}\cos(\pi\dfrac{(40-t)}{10}t) & 30 \le t \le 40 \end{cases} \tag{2}$$

First, at instant t=10s, the left value of $\dot{q}(t)$ is twice as much as the right value, and we can see this effect in the slope change of q(t) in figure 2. We find a similar result at time t=30s, where there is a discontinuity in $\dot{q}(t)$ from π to -2π. It is also unexpected that $\dot{q}(t)$ increases during the last trajectory segment and at t=40s, when ω(t)=0rad/s and we expected zero velocity, its value is -4π rad/s.

The right solution for this example problem is (3), where θ(t) is the phase of the sinusoidal function, and its time derivative $\dot{\theta}(t)$ also follows the trapezoidal profile in figure 1.

$$q(t) = \sin(\theta(t)) \quad \text{where} \quad \theta(t) = \begin{cases} \pi\dfrac{t^2}{20} & 0 \le t < 10 \\ \pi t + \pi & 10 \le t < 30 \\ \pi\dfrac{(40-t)^2}{20} & 30 \le t \le 40 \end{cases} \tag{3}$$

Using (3) the final value of q(t) is the desired zero value. In a general problem, a desired final value of q(t) can't be reached if we use only linear functions in the profile and impose time instants and frequency values. We will see that we need another degree of freedom using, for example, a last quadratic function in the profile, to obtain a desired final value for the sinusoidal trajectory.

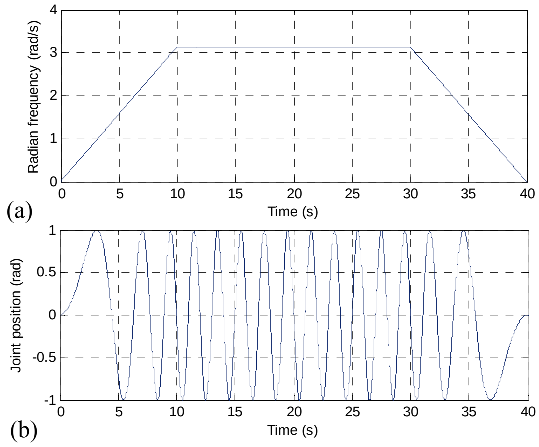


Figure 1: (a) Desired trapezoidal frequency profile and (b) desired trajectory for an actuated robot joint.

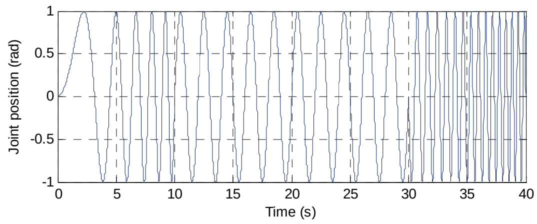


Figure 2: Graphical representation of function (1).

2.1 Basic Definitions

Given a sinusoidal function (sine or cosine) like (4), the argument θ(t) is the instantaneous phase and its time derivative is the instantaneous radian frequency. When θ(t) varies linearly with time, its time derivative is constant and its value is the radian frequency, as in the time interval [10s, 30s] in (3).

$$f(t) = A\sin(\theta(t)) \tag{4}$$

When the phase is quadratic, as in the first and last intervals in (3), the instantaneous radian frequency varies linearly between two values and f(t) is called a chirp function. So, the profile in figure 1 shows the instantaneous radian frequency of a sinusoidal function and represents the concatenation of chirp functions. We will consider here constant frequency sinusoids as a subset of chirp functions, with the same initial and final frequencies.

3 INTERPOLATION OF CHIRP FUNCTIONS

We now present a method for planning trajectories without discontinuities in the joint position and velocity by means of the concatenation of chirp functions. First we obtain the solution without a desired final position, and in section 3.1 we add this constraint to the problem and solve it using a final cubic phase function.

Problem statement: Given a set of N+1 time instants t_i (for i=1 to N+1), and a set of N+1 desired radian frequencies ω_i at each instant t_i, find a set of N chirp functions $f_i(t)$ so that their concatenation represents a continuous trajectory with amplitude A, initial value $q(t_1)=q_1$, and which instantaneous frequency interpolates the frequencies ω_i.

To solve this problem, we will use the function family given by (5).

$$f_i(t) = A\sin(\theta_i(t)) \quad \text{where}$$
$$\theta_i(t) = \begin{cases} 0 & t < t_i \\ a_i(t-t_i)^2 + b_i(t-t_i) + c_i & t_i \le t < t_{i+1} \\ 0 & t_{i+1} \le t \end{cases} \tag{5}$$

The problem centres on finding the coefficients of the phases $\theta_i(t)$, and it is basically the same problem of interpolating trajectories with linear velocity profiles.

The set of conditions that allow us to solve this problem is:

$$\begin{aligned} &\theta_1(t_1) = \arcsin(q_1 / A) = c_1 \\ &\theta_i(t_i) = \theta_{i-1}(t_i) = c_i \\ &\dot\theta_i(t_i) = \omega_i = b_i \\ &\dot\theta_i(t_{i+1}) = \omega_{i+1} = 2a_i(t_{i+1} - t_i) + b_i \end{aligned} \tag{6}$$

From these expressions, the values of a_i and b_i are directly calculated, and the c_i coefficients, that represent the initial phase in each profile's segment, must be calculated iteratively:

$$\begin{aligned} &a_i = \frac{\omega_{i+1} - \omega_i}{2(t_{i+1} - t_i)} \\ &b_i = \omega_i \\ &c_1 = \arcsin(q(t_1)/A) \\ &c_i = a_{i-1}(t_i - t_{i-1})^2 + b_{i-1}(t_i - t_{i-1}) + c_{i-1} \end{aligned} \tag{7}$$

The main problem of this solution is that we can not establish a desired final value of the joint position.

3.1 Concatenation of Chirp Functions with a Final Cubic Phase

Usually, a planned trajectory finishes with zero velocity, and in these cases it is interesting also to reach a desired joint position q_N. This condition adds a new constraint for selecting the last trajectory function $f_N(t)$ and therefore the phase $\theta_N(t)$ needs another degree of freedom, that is, we need to use a cubic function instead of a quadratic one:

$$f_N(t) = A\sin(\theta_N(t)) \quad \text{where}$$
$$\theta_N(t) =$$
$$= \begin{cases} 0 & t < t_N \\ k(t-t_N)^3 + l(t-t_N)^2 + m(t-t_N) + n & t_N \le t < t_{N+1} \\ 0 & t_{N+1} \le t \end{cases} \tag{8}$$

The set of conditions that $\theta_N(t)$ must satisfy is:

$$\begin{aligned} &q_N = A\sin(\theta_N(t_{N+1})) \\ &\theta_N(t_N) = \theta_{N-1}(t_N) \\ &\dot\theta_N(t_N) = \omega_N \\ &\dot\theta_N(t_{N+1}) = \omega_{N+1} \end{aligned} \tag{9}$$

The first of these conditions has many solutions, so we will find the solution corresponding to an almost linear profile, that is, the final phase value will be the nearest to the final value that we will obtain from (7) for i=N. To obtain the coefficients k, l, m and n, we use the next procedure:

1-. First, we calculate the final phase in the quadratic case using the coefficients a_N, b_N and c_N obtained from (7), and also the integer number of revolutions around the unit circle:

$$\theta_{quad} = a_N(t_{N+1} - t_N)^2 + b_N(t_{N+1} - t_N) + c_N \tag{10}$$

$$r = \text{floor}(\theta_{quad} / 2\pi) \tag{11}$$

2-. Next, we select an angle $\alpha \in [0, 2\pi)$, in the same side of the unit circle as θ_{quad}, that satisfies the first condition in (9).

$$\alpha = \begin{cases} \pi - \arcsin(q_N / A) & \cos(\theta_{quad}) < 0 \\ \arcsin(q_N / A) & \cos(\theta_{quad}) \ge 0 \end{cases} \tag{12}$$

If $\alpha<0$, we will add 2π ($\alpha=\alpha+2\pi$).

3-. The desired phase at t_{N+1} is then given by (13).

$$\theta_N(t_{N+1}) = \theta_{N+1} = \alpha + 2r\pi \qquad (13)$$

4-. Finally, we calculate the coefficients by means of (14). These expressions are obtained from (13) and the last three conditions in (9).

$$\begin{aligned} n &= \theta_{N-1}(t_N) \\ m &= \omega_N \\ l &= \frac{3(\theta_{N+1} - \theta_{N-1}(t_N)) - (2\omega_N + \omega_{N+1})(t_{N+1} - t_N)}{(t_{N+1} - t_N)^2} \\ k &= \frac{-2(\theta_{N+1} - \theta_{N-1}(t_N)) + (\omega_N + \omega_{N+1})(t_{N+1} - t_N)}{(t_{N+1} - t_N)^3} \end{aligned} \qquad (14)$$

4 APPLICATION TO AN UNDERACTUATED BIPEDAL MECHANISM

We now present experimental results from simulations where we apply this trajectory planning method to a bipedal walking model. This model is described in more detail in (Berenguer, 2007), and we now include a brief description.

4.1 Bipedal Model and Gait Descriptions

The walking model, shown in figure 3, consists of a light body, a tail connected to it and two legs. Each leg is formed by a parallel link mechanism and a flat rectangular foot. The tail, with an almost horizontal displacement, works as a counterbalance and controls the movement of the biped.

The joint connecting the tail to the body is actuated by an electric motor and it is the only actuated degree of freedom. Connecting the body to each leg are the top joints. Their rotation axis is normal to the frontal plane, so they allow the mechanism to raise a foot while both feet remain parallel to the ground. These top joints are passive joints with negligible friction. Finally, each parallel link mechanism has four joints, and we consider that in one of these joints (the ankle joint) there is a spring with friction. Due to the characteristics of the parallel link mechanism, these four joints represent only one passive degree of freedom for each leg of the mechanism. In summary, the model has eleven joints, four passive degrees of freedom and one actuated degree of freedom.

We now describe how the mechanism can walk when the tail moves side to side in an oscillating way. We start by supposing that the biped is at an equilibrium position with the tail in its central position (Fig.4.a). Both ankle springs hold the weight of the mechanism and it stays almost vertical.

When the tail moves to a lateral position of the mechanism, its mass acts as a counterbalance and produces the rise of one of the feet (Fig.4.b). Then only one spring holds the body, so the stance leg falls forward and the swing leg moves forward as a pendulum until the foot contacts the ground (fig.4.c).

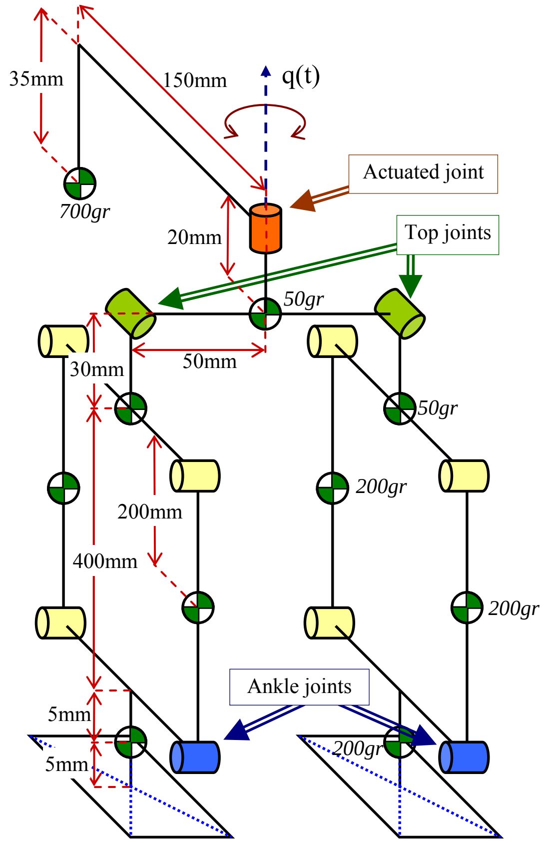


Figure 3: Model of the biped mechanism.

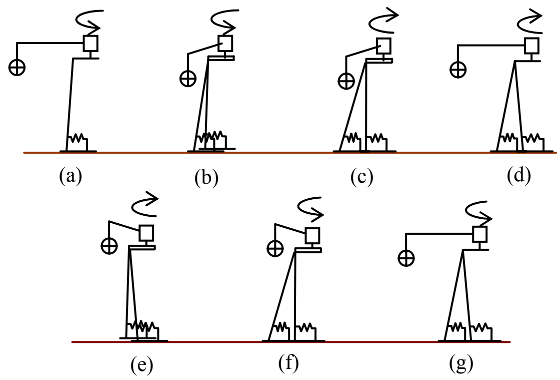


Figure 4: Phases during a stride.

During the new double support phase, the tail moves to the other side and the ankle springs move the body backwards (fig.4.d). When the tail reaches the other side, the second foot rises and a new step is generated (fig.4.e). In this single support phase, the spring of the foot that is in the ground produces enough torque to take the body forward again. This second step finishes with a new contact of the swing leg with the ground (fig.4.f).

Figure 4.g represents the last instant of this initial stride, and the starting point of a new one or the final configuration of a completed trajectory. We can see that if the tail stops, the system will stay in a steady configuration with no energy cost.

4.2 Example of Trajectory Generation

We want to design a trajectory that allows us to evaluate and analyze the biped behaviour at three different oscillation frequencies. The frequency profile must achieve these frequencies in a linear way, and keep them during some periods, so we can suppose a quasi-periodic gait at the end of each constant frequency segment. The desired oscillation amplitude is 1.5 rad and the radian frequencies and time instants are:

$$\begin{aligned} \omega_i[\mathrm{rad/s}] &= \{0 \quad 0.5 \quad 0.5 \quad 1 \quad 1 \quad 1.5 \quad 1.5 \quad 0\} \\ t_i[\mathrm{s}] &= \{0 \quad 20 \quad 70 \quad 90 \quad 140 \quad 160 \quad 200 \quad 300\} \end{aligned} \tag{15}$$

Using (7) we obtain q(t) with a linear profile shown in figure 5.

$$\begin{aligned} & q(t) = 1.5\sin(\theta(t)) \\ & \theta(t) = \theta_i(t) \quad \text{at} \quad t_i \le t < t_{i+1} \\ & \theta_1(t) = 0.0125t^2 \\ & \theta_2(t) = 0.5(t-20)+5 \\ & \theta_3(t) = 0.0125(t-70)^2 + 0.5(t-70)+30 \\ & \theta_4(t) = (t-90)+45 \\ & \theta_5(t) = 0.0125(t-140)^2 + (t-140)+95 \\ & \theta_6(t) = 1.5(t-160)+120 \\ & \theta_7(t) = -0.0075(t-200)^2 + 1.5(t-200)+180 \end{aligned} \tag{16}$$

As we can see in figure 6, this solution provides the final joint position q(300)= -0.76rad. If the desired final tail position is q=0rad, it will be necessary to apply the procedure in section 3.1. The solution in this case is the same as in (16) but with the last phase $\theta_7(t)$ given by:

$$\begin{aligned} \theta_7(t) = {} & 1.062\times10^{-6}(t-200)^3 + \\ & + 7.659\times10^{-3}(t-200)^2 + 1.5(t-200)+180 \end{aligned} \tag{17}$$

Figure 5 shows the frequency profiles of both solutions. We can see a most linear segment in the last time interval for the second solution which practically overlaps the first solution's profile. Figure 6 shows both trajectories nearly overlapping during the last time interval, with the same number of oscillations but different final value.

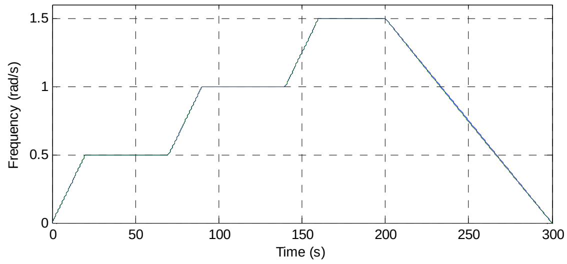


Figure 5: Instantaneous frequency profiles.

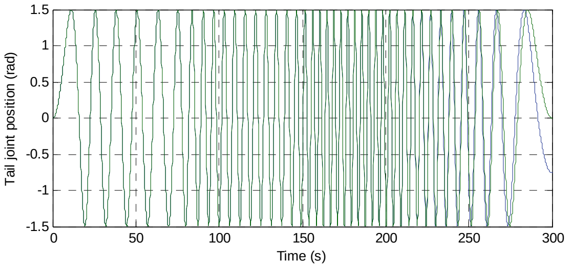


Figure 6: Trajectory of the tail joint.

4.3 Evaluation of the Mechanism Behaviour

The bipedal mechanism walks with a forward speed which is proportional to the stride length and frequency. The stride frequency is the same as the tail oscillation frequency and the stride length depends on the tail frequency and also on other parameters of the model. If we fix the values of these other parameters, the speed and energy consumption of the mechanism will depend in a non linear way on the stride frequency. Using linear frequency profiles we can analyze this dependency and estimate a near optimal joint frequency for an established set of model parameters.

Figure 7 shows the distance covered by the biped and the mechanical energy required by the tail joint during the execution of the last trajectory defined in section 4.2. The relative small amplitude oscillations are due to the forward and backward oscillation of the body during walking. The mechanical energy has been calculated by the integration of the absolute value of the product between the angular velocity of the joint and the required torque. We observe an important difference in the walking speed for the first and second oscillation frequencies, and a much smaller variation for the second and third ones, while the power consumption varies significantly for these last frequencies. So, we find a loss of efficiency when we increment the stride frequency.

Table 1 presents numerical data considering the last stride period just before a change in the oscillation frequency. We suppose that the gait is almost periodic during this last period. Figure 8 shows these values and also an estimation of the same magnitudes at different frequencies. This estimation is obtained from the last profile's segment, which covers all frequencies between 0 and 1.5rad/s, considering each half-oscillation as an approximation of half-period of a sinusoid.

As we can see, speed goes up quickly at low frequencies because stride length also grows. For frequencies greater than 1.04rad/s, stride length decreases with frequency and speed rises more slowly. We also notice that speed and power have similar behaviour (S-curve) before this frequency. After that, the power slope increases whereas speed slope decreases. We consider this 1.04rad/s frequency as a near optimal oscillation frequency for the actuated joint.

Table 1: Stride length, speed and mechanical power at three different stride frequencies during a stride.

Stride frequency (rad/s)	Stride period (s)	Stride length (m)	Speed (m/s) x10^{-3}	Power (W) x10^{-3}
0.5	12.57	0.0807	6.422	1.386
1.0	6.283	0.1992	31.704	10.707
1.5	4.188	0.1767	42.184	18.694

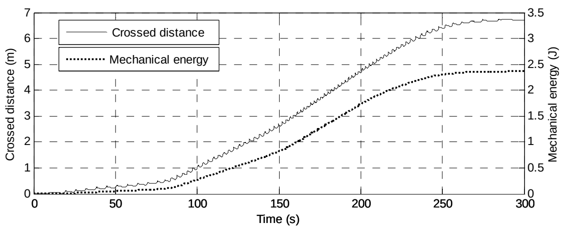


Figure 7: Crossed distance and mechanical energy during the trajectory execution.

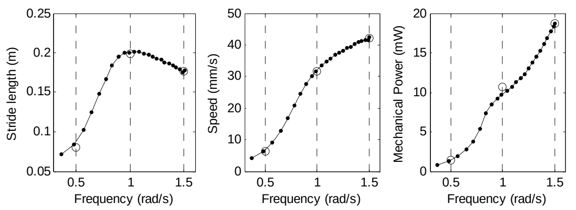


Figure 8: Stride length, mechanism speed and required mechanical power at different joint frequencies.

5 CONCLUSIONS

In this work we propose a method for planning oscillatory trajectories based on the concatenation of chirp functions. By means of adding a final cubic function, the joint can also reach a desired final position following a nearly linear frequency profile. Our aim is to apply this method to a bipedal robot that walks moving a tail in an oscillatory way.

This planning method allows us to study the gait efficiency at different stride frequencies during the design and adjusting phase. On the other hand, the implementation of this planner will allow a real prototype to select the forward speed as a function of the obstacles density, ground inclination or for optimization requirements.

REFERENCES

Spong, M. W., Vidyasagar, M., 1989, *Robot Dynamics and Control*, John Wiley & Sons.

Sciavicco, L., Siciliano, B., 1996. *Modeling and Control of Robot Manipulators*, The McGraw-Hill Comp., 1st edition.

Craig, J. J., 2006. *Introduction to robotics*, Prentice Hall, 3th edition.

Morimoto, J., Gen Endo, Jun Nakanishi, Hyon, S., Cheng, G., Bentivegna, D., Atkeson, C.G., 2006, "Modulation of simple sinusoidal patterns by a coupled oscillator model for biped walking", in *Proc. 2006 IEEE International Conference on Robotics and Automation*, USA, pp.1579-1584.

Sfakiotakis, M., Tsakiris, D. P., Vlaikidis, A., 2006, "Biomimetic Centering for Undulatory Robots", in *Proc. of the 1st IEEE Int. Conference on Biomedical Robotics and Biomechatronics*, pp.744- 749.

Murray, R. M., Sastry, S., 1993, "Nonholonomic motion planning: Steering using sinusoids", *IEEE Trans. on Automatic Control*, 38(5), pp.700-716.

McClung, A. J., Cham, J. G., Cutkosky, M. R., 2004, "Rapid Maneuvering of a Biologically Inspired Hexapedal Robot" in *ASME IMECE Symp. on Adv. in Robot Dynamics and Control*, USA.

Leavitt, J., Sideris, A., Bobrow, J. E., 2006, "High bandwidth tilt measurement using low-cost sensors", *IEEE/ASME Trans. on Mechatronics*, 11(3), pp.320-327.

Berenguer, F. J., Monasterio-Huelin, F., 2006, "Easy design and construction of a biped walking mechanism with low power consumption", in *Proc. of CLAWAR'06, the 9th Int. Conference on Climbing and Walking Robots*, Belgium, pp. 96-103.

Berenguer, F. J., Monasterio-Huelin, F., 2007, "Stability and Smoothness Improvements for an Underactuated Biped with a Tail", in *Proc. of the 2007 IEEE Int. Symposium on Industrial Electronics*, Vigo, Accepted.

MODELING AND OPTIMAL TRAJECTORY PLANNING OF A BIPED ROBOT USING NEWTON-EULER FORMULATION

David Tlalolini, Yannick Aoustin and Christine Chevallereau
Institut de Recherche en Communications et Cybernétique de Nantes
CNRS, Ecole Centrale de Nantes, Universit de Nantes, 1 rue la No, 44321 Nantes, France
David.Tlalolini-Romero@irccyn.ec-nantes.fr; Yannick.Aoustin@irccyn.ec-nantes.fr
Christine.Chevallereau@irccyn.ec-nantes.fr

Keywords: 3D biped, cyclic walking gait, parametric optimization, constraints.

Abstract: The development of an algorithm to achieve optimal cyclic gaits in space for a thirteen-link biped and twelve actuated joints is proposed. The cyclic walking gait is composed of successive single support phases and impulsive impacts with full contact between the sole of the feet and the ground. The evolution of the joints are chosen as spline functions. The parameters to define the spline functions are determined using an optimization under constraints on the dynamic balance, on the ground reactions, on the validity of impact, on the torques and on the joints velocities. The criterion considered is represented by the integral of the torque norm. The algorithm is tested for a biped robot whose numerical walking results are presented.

1 INTRODUCTION

The design of walking gaits for legged robots and particularly the bipeds has attracted the interest of many researchers for several decades. Due to the unilateral constraints of the biped with the ground and the great number of degrees of freedom, this problem is not trivial. Intuitive methods can be used to obtain walking gaits as in (Grishin et al., 1994). Using experimental data and physical considerations, the authors defined polynomial functions in time for a prototype planar biped. This method is efficient. However to build a prototype and to choose the appropriate actuators or to improve the autonomy of a biped, an optimization algorithm can lead to very interesting results. In (Rostami and Besonnet, 1998) the Pontryagin's principle is used to design impactless nominal trajectories for a planar biped with feet. However the calculations are complex and difficult to extend to the 3D case. As a consequence a parametric optimisation is a useful tool to find optimal motion.

The choice of optimisation parameters is not unique. The torques, the Cartesian coordinates or joint coordinates can be used. Discrete values for the torques defined at sampling time are used as optimization parameters in (Roussel et al., 2003). However it is necessary, when the torque is an optimised variable, to solve the inverse dynamic problem to find the joint accelerations and integrations are used to obtain the evolution of the reference trajectory in velocity and in position. Thus this approach require many calculation : the direct dynamic model is complex and many evaluations of this model is used in the integration process. In (Beletskii and Chudinov, 1977), (Bessonnet et al., 2002), (Channon et al., 1992), (Zonfrilli et al., 2002), (Chevallereau. and Aoustin, 2001) or (Miossec and Aoustin, 2006) to overcome this difficulty, the parametric optimization defines directly the reference trajectories of Cartesian coordinates or joint coordinates for 2D bipeds with feet or without feet. An extension of this strategy is given in this paper for a 3D biped with with twelve motorized joints. The dynamic model is more complex than for a 2D biped, so its computation cost is important in the optimisation process and the use of Newton-Euler method to calculate the torque is more appropriate than the Lagrange method usually used. Since the inverse dynamic model is used only to evaluate the torque for the constraints and criterion calculation, the number of evaluation of the torque can be limited. The desired motion is based on the solution of an optimal problem whose constraints depend on the nonlinear multibody system dynamics of the 12 DoF biped and physical contact constraints with the environment.

A half step of the cyclic walking gait is composed uniquely of a single support and an instantaneous double support that is modelled by passive impulsive equations. This walking gait is simpler that the human gait, but with this simple model the coupling effect between the motion in frontal plan and sagittal plane can be studied. A finite time double support phase in not considered in this work currently because for rigid modelling of robot, a double support phase can usually be obtained only when the velocity of the swing leg tip before impact is null. This constraint has two effects. In the control process it will be difficult to touch the ground with a null velocity, as a consequence the real motion of the robot will be far from the ideal cycle. Furthermore, large torques are required to slow down the swing leg before the impact and to accelerate the swing leg at the beginning of the single support. The energy cost of such a motion is higher than a motion with impact in the case of a planar robot without feet (Chevallereau. and Aoustin, 2001), (Miossec and Aoustin, 2006).

Therefore a dynamic model is calculated for the single phase. An impulsive model for the impact on the ground with complete surface of the foot sole of the swing leg is deduced from the dynamic model for the biped in double support phase. It takes into account the wrench reaction from the ground. This model is founded on the Newton Euler algorithm, considering that the reference frame is connected to a stance foot. The evolution of joint variables are chosen as a spline function of time instead of usual polynomial functions to prevent oscillatory phenomenon during the optimization process (see (Chevallereau. and Aoustin, 2001), (Saidouni and Bessonnet, 2003) or (L. Hu and Sun, 2006)). The coefficients of the spline functions are calculated as function of initial, intermediate and final configurations and initial and final velocities of the robot which are optimization variables. Taking into account the impact and the fact that the desired walking gait is cyclic, the number of optimization variables is reduced. The criterion considered is the integral of the torque norm. During the optimization process, the constraints on the dynamic balance, on the ground reactions, on the validity of impact, on the limits of the torques, on the joints velocities and on the motion velocity of the biped robot are taken into account. The paper is organized as follows. The 3D biped and its dynamic model are presented in Section II. The cyclic walking gait and the constraints are defined in Section III. The optimization parameters, optimization process and the criterion are discussed in Section IV. Simulation results are presented in Section V. Section VI contains our conclusion and perspectives.

2 MODELS OF THE STUDIED BIPED ROBOT

2.1 Biped Model

We considered an anthropomorphic biped robot with thirteen rigid links connected by twelve motorized joints to form a tree structure. It is composed of a torso, which is not directly actuated, and two identical open chains called legs that are connected at the hips. Each leg is composed of two massive links connected by a joint called knee. The link at the extremity of each leg is called foot which is connected at the leg by a joint called ankle. Each revolute joint is assumed to be independently actuated and ideal (frictionless). The ankles of the biped robot consist of the pitch and the roll axes, the knees consist of the pitch axis and the hips consist of the roll, pitch and yaw axes to constitute a biped walking system of two 2-DoF ankles, two 1-DoF knees and two 3-DoF hips as shown in figure 1. The action to walk associates single support phases separated by impacts with full contact between the sole of the feet and the ground, so that a model in single support, a model in double support and an impact model are derived.

2.2 Geometric Description of the Biped

To define the geometric structure of the biped walking system we assume that the link 0 (stance foot) is the base of the biped robot while link 12 (swing foot) is the terminal link. Therefore we have a simple open loop robot which geometric structure can be described using the notation of Khalil and Kleinfinger (Khalil and Dombre, 2002). The definition of the link frames is given in figure 1 and the corresponding geometric parameters are given in Table I. The frame R_0 coordinates, which is fixed to the tip of the right foot (determined by the width l_p and the length L_p), is defined such that the axis z_0 is along the axis of frontal joint ankle. The frame R_{13} is fixed to the tip of the left foot in the same way that R_0.

2.3 Dynamic Model in Single Support Phase

During the single support phase the stance foot is assumed to remain in flat contact on the ground, *i.e.*, no sliding motion, no take-off, no rotation. Therefore the dynamics of the biped is equivalent to an 12 DoF manipulator robot. Let $q \in \mathbb{R}^{12}$ be the generalized coordinates, where $q_1, ..., q_{12}$ denote the relative angles of the joints, $\dot{q} \in \mathbb{R}^{12}$ and $\ddot{q} \in \mathbb{R}^{12}$ are the velocity

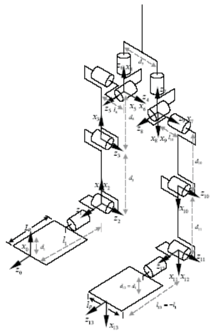


Figure 1: The multi-body model and link frames of the biped robot.

Table 1: Geometric parameters of the biped.

j	$a(j)$	α_j	θ_j	r_j	d_j
1	0	0	q_1	l_1	d_1
2	1	$\frac{\pi}{2}$	q_2	0	0
3	2	0	q_3	0	d_3
4	3	0	q_4	l_4	d_4
5	4	$-\frac{\pi}{2}$	$q_5-\frac{\pi}{2}$	0	0
6	5	$-\frac{\pi}{2}$	q_6	0	0
7	6	0	q_7	0	d_7
8	7	$\frac{\pi}{2}$	$q_8-\frac{\pi}{2}$	0	0
9	8	$-\frac{\pi}{2}$	q_9	0	0
10	9	0	q_{10}	$l_{10}=l_4$	$d_{10}=d_4$
11	10	0	q_{11}	0	$d_{11}=d_3$
12	11	$\frac{\pi}{2}$	q_{12}	0	0
13	12	0	q_{13}	$l_{13}=-l_1$	$d_{13}=d_1$

and acceleration vectors respectively. The dynamic model is computed using the Newton-Euler method (see (Khalil and Dombre, 2002)) represented by the following relation

$$\Gamma = \mathrm{f}(q,\dot{q},\ddot{q},F_t) \tag{1}$$

where $\Gamma \in \mathbb{R}^{12}$ is the joint torques vector and F_t is the external wrench (forces and torques), exerted by the swing foot on the ground. In single support phase $F_t = 0$ and in double support phase $F_t \neq 0$.

In order to denote the dynamic model under the Lagrange form

$$\Gamma = D_s(q)\ddot{q} + H_s \tag{2}$$

with

$$H_s = (C_s(q,\dot{q}) + G_s(q)) \tag{3}$$

the equation (1) is used. In such calculation the matrix D_s and the vector H_s are needed. $C_s \in \mathbb{R}^{12}$ represents the Coriolis and centrifugal forces and $G_s \in \mathbb{R}^{12}$ is the vector of gravity.

The matrix D_s is calculated by the algorithm of Newton-Euler, by noting from the relation (1), (M.W.Walker and D.E.Orin, 1982), that the i^{th} column is equal to Γ if

$$\dot{q} = 0, g = 0, \ddot{q} = e_i, F_t = 0$$

$e_i \in \mathbb{R}^{12\times 1}$ is the unit vector, whose elements are zero except the i^{th} element which is equal to 1.

The calculation of the vector H_s is obtained in the same way that D_s considering that $H_s = \Gamma$ if $\ddot{q} = 0$. Therefore, the dynamic model under the Lagrange form is denoted by the following matrix equations

$$\Gamma = D_s(q)\ddot{q} + H_s(q,\dot{q})$$

where $D_s \in \mathbb{R}^{12\times 12}$ is the symmetric definite positive inertia matrix.

To take easily into account the effect of the reaction force on the stance foot, it is interesting to add 6 coordinates to describe the situation of the stance foot. Newton variables are used for this link, thus its velocity is described by the linear velocity of frame $R_0 : V_0$ and angular velocity ω_0. Since the stance foot is assumed to remain in flat contact, the resultant ground reaction force/moment F_R and M_R are computed by using the Newton-Euler algorithm. $\omega_0 = \mathbf{0}$, $\dot{\omega}_0 = \mathbf{0}$ and $\dot{V}_0 = -g$ are the initial conditions of the Newton-Euler algorithm to take into account the effect of gravity. So, the equation (2) becomes

$$D(X)\dot{V} + C(V,q) + G(X) = D_\Gamma\Gamma + D_R R_{F_R} \tag{4}$$

where $X = [X_0, \alpha_0, q]^T \in \mathbb{R}^{18}$, X_0 and α_0 is the position and the orientation variables of frame R_0, $V = [{}^0V_0, {}^0\omega_0, \dot{q}]^T \in \mathbb{R}^{18}$ and $\dot{V} = [{}^0\dot{V}_0, {}^0\dot{\omega}_0, \ddot{q}]^T \in \mathbb{R}^{18}$. $D \in \mathbb{R}^{18\times 18}$ is the symmetric definite positive inertia matrix, $C \in \mathbb{R}^{18}$ represents the Coriolis and centrifugal forces, $G \in \mathbb{R}^{18}$ is the vector of gravity. $R_{F_R} = [F_R, M_R]^T \in \mathbb{R}^6$ is the ground reaction forces on the stance foot, calculated by the Newton-Euler algorithm, $D_\Gamma = [0_{6\times 12} \mid I_{12\times 12}]^T \in \mathbb{R}^{18\times 12}$ and $D_R = [I_{6\times 6} \mid 0_{12\times 6}]^T \in \mathbb{R}^{6\times 18}$ are constant matrices composed of 1 and 0.

In the optimization process, the torques and force are calculated with the Newton-Euler algorithm and not with the equation (4). The Newton-Euler is much more efficient from the computation point of view, (Khalil and Dombre, 2002).

2.4 Dynamic Model in Double Support Phase

In double support phase, only the forces and moments of interaction of the left foot with the ground have to be added. Then, the model (4) becomes

$$D(X)\dot{V} + C(V,q) + G(X) + D_f R_f = D_\Gamma\Gamma + D_R R_{F_R} \tag{5}$$

where $R_f \in \mathbb{R}^6$ represents the vector of forces F_{12} and moments M_{12} exerted by the left foot on the ground. This wrench is naturally expressed in frame R_{12} : ${}^{12}F_{12}$, ${}^{12}M_{12}$. The virtual work δW_{12} of this wrench is :

$$\delta W_{12} = {}^{12}F_{12}^{T\,12}d_{12} + {}^{12}M_{12}^{12}\delta_{12} \quad (6)$$

where ${}^{12}d_{12}$ represents an infinitesimal virtual displacement of the link 12 and ${}^{12}\delta_{12}$ represents an infinitesimal virtual angular displacement. The relation between these virtual displacements, ${}^{12}d_{12}$ and ${}^{12}\delta_{12}$, and the virtual joints displacement δq_{12} are the same that between the velocities ${}^{12}V_{12}, {}^{12}\omega_{12}$ and $\dot{q}_{12}$.

Usually the velocities of link 12 can be expressed as

$$\begin{bmatrix} V_{12} \\ w_{12} \end{bmatrix} = \begin{bmatrix} V_0 + w_0 \times {}^0P_{12} \\ w_{12} \end{bmatrix} + J_{12}\dot{q} \quad (7)$$

where ${}^0P_{12}$ is the vector linking the origin of frame R_0 and the origin of frame R_{12} expressed in frame R_0, $J_{12} \in \mathbb{R}^{6\times 12}$ is the Jacobian matrix of the robot, $J_{12}\dot{q}$ represents the effect of the joint velocities on the Cartesian velocity of link 12. The velocities V_{12} and w_{12} must be expressed in frame R_{12}, thus we write (7):

$$\begin{bmatrix} {}^{12}V_{12} \\ {}^{12}w_{12} \end{bmatrix} = \begin{bmatrix} {}^{12}A_0 & -{}^{12}A_0^0\hat{P}_{12} \\ 0_{3\times 3} & {}^{12}A_0 \end{bmatrix} \begin{bmatrix} {}^0V_0 \\ {}^0w_0 \end{bmatrix} + {}^{12}J_{12}\dot{q} \quad (8)$$

where ${}^{12}A_0 \in \mathbb{R}^{3\times 3}$ is the rotation matrix, which defines the orientation of frame R_0 with respect to frame R_{12}. Term ${}^0\hat{P}_{12}$ is the skew-symmetric matrix of the vector product associated with vector ${}^0P_{12}$.

$${}^0\hat{P}_{12} = \begin{bmatrix} 0 & -P_z & P_y \\ P_z & 0 & -P_x \\ -P_y & P_x & 0 \end{bmatrix}$$

Defining matrix $D_f \in \mathbb{R}^{18\times 6}$ as the concatenation of two matrices such that $D_f = [T \ |{}^{12}J_{12}]^T$, where ${}^{12}J_{12} \in \mathbb{R}^{6\times 12}$ is the Jacobian matrix of the robot and $T \in \mathbb{R}^{6\times 6}$ equals

$$T = \begin{bmatrix} {}^{12}A_0 & -{}^{12}A_0^0\hat{P}_{12} \\ 0_{3\times 3} & {}^{12}A_0 \end{bmatrix} \quad (9)$$

Then, the linear and angular velocities of the swing foot in frame R_{12} is :

$$\begin{bmatrix} {}^{12}V_{12} \\ {}^{12}w_{12} \end{bmatrix} = D_f^T V \quad (10)$$

Then D_f can be defined by applying the virtual principle on the second leg. However in order to compute the matrix D_f, it is necessary, either to calculate the matrix ${}^{12}J_{12}$ jacobian by a traditional method, by taking into account the equation (9), or to calculate this matrix by the algorithm of Newton-Euler, by noting from relation (5) that the i^{th} column is equal to $D_\Gamma \Gamma + D_R R_{F_R}$ if

$$\dot{V} = \mathbf{0}, V = \mathbf{0}, g = 0 \text{ and } R_f = e_i$$

$e_i \in \mathbb{R}^{6\times 1}$ is the unit vector, whose elements are zero except the i^{th} element which is equal to 1.

2.5 Impact Equations for Instantaneous Double Support

When the swing foot touches the ground, an impact exists. In reality many possibilities can appear for an impact (partial contact with the sole on the ground, elastic deformations of the bodies and the ground). To simplify our study this impact is assumed to be instantaneous and inelastic with complete surface of the foot sol touching the ground. This means that the velocity of the swing foot touching the ground is zero after its impact. We assume that the ground reaction at the instant of impact is described by a Dirac delta-function with intensity I_{R_f}. Assuming that the previous stance foot is motionless before the impact and does not remains on the ground after the impact the dynamic model during the impact is (see (Formal'sky, 1982) and (M. Sakaguchi and Koizumi, 1995))

$$D(X)\Delta V = -D_f I_{R_f} \quad (11)$$

$$D_f^T V^+ = 0 \quad (12)$$

$$\begin{bmatrix} {}^0V_0^- \\ {}^0w_0^- \end{bmatrix} = \begin{bmatrix} 0_{3\times 1} \\ 0_{3\times 1} \end{bmatrix} \quad (13)$$

where $\Delta V = (V^+ - V^-)$ is the change of velocity caused by the impact and V^+ (respectively V^-) denote the linear and angular velocity of the stance foot and also the joint velocities of the biped after (respectively before) the impact. These equations form a system of linear equations which solution allows to know the impulse forces and the velocity after the impact, thus they can be applied to the biped walking system.

3 DEFINITION OF THE WALKING CYCLE

Because biped walking is a periodical phenomenon our objective is to design a cyclic biped gait. A complete walking cycle is composed of two phases: a single support phase and a double support phase which is modeled through passive impact equations. The single support phase begins with one foot which stays on the ground while the other foot swings from the rear to the front. We shall assume that the double support phase is instantaneous, this means that when the swing leg touches the ground the stance leg takes off. There are two facets to be considered for this problem. The definition of reference trajectories and the method to determine a particular solution of it. This section is devoted to the definition of reference trajectories. The optimal process to choose the best solution of parameters, allowing a symmetric half step, from the point of view of a given criterion will be described in the next section.

3.1 Cyclic Walking Trajectory

Since the initial configuration is a double support configuration, the both feet are on the ground, the twelve joint coordinates are not independent. Because the absolute frame is attached to the right foot we define the situation of the left foot by $(y_{lf}, z_{lf}, \phi_{lf})$ and the situation of the middle of the hips $(x_h, y_h, z_h, \theta_h)$, both expressed in R_0 frame. (y_{lf}, z_{lf}) is the coordinate, in the horizontal plane, of the left foot position, ϕ_{lf} denotes the left foot yawing motion, (x_h, y_h, z_h) is the hip position and θ_h defines the hip pitching motion. The values of the joint variables are solution of the inverse kinematics problem for a leg, which may also be considered as a 6-link manipulator. The problem is solved with a symbolic software, (SYMORO+, see (Khalil and Dombre, 2002)).

In order to deduce the final configuration, we impose a symmetric role of the two legs, therefore from the initial configuration, the final configuration is deduced as:

$$q_{f_{DS}} = E q_{i_{DS}} \tag{14}$$

where $E \in \mathbb{R}^{12\times 12}$ is an inverted diagonal matrix which describes the legs' exchange.

Taking into account the impulsive impact (11)-(13), we can compute the velocity after the impact. Therefore, the velocity after the impact, $\dot{q}^+$, can be calculated when the velocity before the impact, $\dot{q}^-$, is known. The use of the defined matrix E allows us to calculate the initial velocity for the current half step as:

$$\dot{q}_i = E\dot{q}^+. \tag{15}$$

By this way the conditions of cyclic motion are satisfied.

3.2 Constraints

In order to insure that the trajectory is possible, many constraints have to be considered.

3.2.1 Magnitude Constraints on Position and Torque

- Each actuator has physical limits such that

$$|\Gamma_i| - \Gamma_{i,\max} \le 0, \quad \text{for } i = 1,\dots,12 \tag{16}$$

where $\Gamma_{i,\max}$ denotes the maximum value for each actuator.

$$|\dot{q}_i| - \dot{q}_{i,\max} \le 0, \quad \text{for } i = 1,\dots,12 \tag{17}$$

where $\dot{q}_{i,\max}$ denotes the maximum velocity for each actuator.

- The upper and lower bounds of joints for the configurations during the motion are:

$$q_{i,\min} \le q_i \le q_{i,\max}, \quad \text{for } i = 1,\dots,12 \tag{18}$$

$q_{i,\min}$ and $q_{i,\max}$ stands respectively for the minimum and maximum joint limits.

3.2.2 Geometrical Constraints in Double Support Phase

- The distance $d(hip, foot)$ between the foot in contact with the ground and the hip must remain within a maximal value, *i.e.*,

$$d(hip, foot) \le l_{hip}. \tag{19}$$

This condition must hold for initial and final configurations of the double support.

- In order to avoid the internal collision of both feet through the lateral axis the heel and the toe of the left foot must satisfy

$$y_{heel} \le -a \text{ and } y_{toe} \le -a \tag{20}$$

with $a > \frac{l_p}{2}$ and and l_p is the width of right foot.

3.2.3 Walking Constraints

- During the single support phase to avoid collisions of the swing leg with the stance leg or with the ground, constraints on the positions of the four corners of the wing foot are defined.
- We must take into account the constraints on the ground reaction $R_{F_R} = [R_{F_{Rx}}, R_{F_{Ry}}, R_{F_{Rz}}]^T$ for the stance foot in single support phase as well as impulsive forces $I_{R_f} = [I_{R_{fx}}, I_{R_{fy}}, I_{R_{fz}}]^T$ on the foot touching the ground in instantaneous double support phase. The ground reaction and impulsive forces must be inside a friction cone defined by the friction coefficient μ. This is equivalent to write

$$\sqrt{R^2_{F_{Ry}} + R^2_{F_{Rz}}} \le \mu R_{F_{Rx}} \tag{21}$$

$$\sqrt{I^2_{R_{fy}} + I^2_{R_{fz}}} \le \mu I_{R_{fx}} \tag{22}$$

The ground reaction forces and the impulsive forces at the contact can only push the ground but may not pull from ground, then the condition of no take off is deduced:

$$R_{f_x} \ge 0 \tag{23}$$

$$I_{R_{fx}} \ge 0. \tag{24}$$

- In order to maintain the balance in dynamic walking, the $ZMP \equiv CoP$, (Zero Moment Point equivalent to the Center of Pressure, see (Vukobratovic and Stepanenko, 1972), point must be within the support polygon, *i.e.*, the distance from CoP to support polygon is negative

$$d(CoP, SP) \leq 0, \tag{25}$$

where SP denotes the support polygon determined by the width l_p and the length L_p of the feet.

4 PARAMETRIC OPTIMIZATION

4.1 The Cubic Spline

To describe the joint motion by a finite set of parameters we choose to use for each joint a piecewise function of the form

$$\begin{aligned} q_i &= \varphi_i(t) = \begin{cases} \varphi_{i1}(t) & if \quad t_0 \leq t \leq t_1 \\ \varphi_{i2}(t) & if \quad t_1 \leq t \leq t_2 \\ \vdots & \\ \varphi_{in}(t) & if \quad t_{n-1} \leq t \leq t_n \end{cases} \\ i &= 1, \ldots 12 \end{aligned}$$

where $\varphi_k(t)$ are polynomials of third-order such that

$$\varphi_{ik}(\mathbf{a}_{ik}, t) = \sum_{j=0}^{3} a_{ikj}(t - t_{k-1})^j, \ k = 1, \ldots, n \ \forall t \in [t_0, t_n] \tag{26}$$

where a_{ikj} are calculated such that the position, velocity and acceleration are always continuous in $t_0, t_1, \ldots t_n$. We used $n = 3$, thus the motion is defined by a specified initial configuration, a final configuration in double support and two intermediate configurations in single support taking into account the initial and final velocity as boundary conditions.

4.2 Optimization Parameters

A parametric optimization problem has to be solved to design a cyclic bipedal gait with successive single and double support phases. This problem depends on parameters to prescribe the two intermediate configurations, q_{int1} and q_{int2}, and the final velocity $\dot{q}_f$ in the single support phase. Taking into account the conditions (14) and (15) the minimal number of parameters necessary to define the joint motion are:

1. Twenty-four parameters are needed to define the two intermediate configurations in single support phase, twelve parameters for the first intermediate configuration $q_{i,int1}$ and twelve parameters for the second intermediate configuration, $q_{i,int2}$ for $i = 1, \ldots, 12$.
2. The velocity before the impact is also prescribed by twelve parameters, $\dot{q}_i^-$ $(i = 1, \ldots 12)$.
3. The left foot yawing motion denoted by ϕ_{lf} and its position (y_{lf}, z_{lf}) in the horizontal plane as well as the situation of the middle of the hips defined by $(x_h, y_h, z_h, \theta_h)$ in double support phase are chosen as parameters.

Let us remark that to define the initial and final configurations in double support nine parameters are required however we define these configurations with only seven parameters. The two others parameters, orientation of the middle of the hips in frontal and transverse plane, are fixed to zero. The duration of a half step, T_s, is fixed arbitrarily.

4.3 Criterion

In the optimization process we consider, as criterion J_Γ, the integral of the norm of the torque divided by the half step length. In other words we are minimizing a quantity proportional to the energy required for a motion

$$J_\Gamma = \frac{1}{d} \int_0^{T_s} \Gamma^T \Gamma dt \tag{27}$$

where T_s is the time of the half step. This general form of minimal energy performance represents the losses by Joule effects to cover distance d.

4.4 Optimization Algorithm

Generally, many values of parameters can give a periodic bipedal gait satisfying constraints (17)-(24). A parametric optimization process, which objective is to minimize J_Γ under nonlinear constraints, is used to find a particular nominal motion. This optimization process can be formally stated as

$$\left. \begin{array}{ll} \text{minimize} & J_\Gamma(p) \\ \text{subject to} & g_i(p) \leq 0 \quad i = 1, 2, \ldots, l \end{array} \right\} \tag{28}$$

where p is the vector of parameters, $J_\Gamma(p)$ is the criterion to minimize with l constraints $g_i(p) \leq 0$ to satisfy. This constraints are given in section 3.2. The nonlinear constrained problem is solved using the Matlab function *fmincon*. This optimization function provide an optimization algorithm based on the Sequential Quadratic Programming (SQP). Therefore, this nonlinear optimization problem with forty-three variables: twenty-four for the two intermediate configurations in single support, twelve for the velocity before the impact and seven to solve the inverse kinematics problem, subject to the constraints given by (17)-(24), is solved numerically.

Table 2: Parameters of SPEJBL.

Physical Parameters	Mass (kg)	Length (m)
Torso	0.3967	$d_7 = 0.140$
Right Leg		
Hip	0.2604	linked to torso
Thigh	0.1224	$d_4 = 0.120$
Shin	0.0558	$d_3 = 0.120$
Ankle	0.1278	$d_1 = 0.042$
Foot	0.3045	$L_p = 0.178$

5 SIMULATION RESULTS

To validate our proposed method, we present the results of an optimal motion for the biped shown in figure 2. The desired trajectory was obtained by the optimization process presented in Section IV, with the minimization of the criterion (27) satisfying the constraints given by (17)-(24). The figure 4 shows the evolution of the optimal motion for a half step with duration, of a single support, which is equal to 0.58 s. For the simulation, we use the physical parameters of the SPEJBL[1]. The physical parameters of SPEJBL are collected in Table 2. Figure 2 shows the photo of SPEJBL and also the dimensional design drawn by VariCAD software.

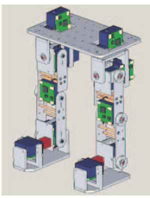


Figure 2: Dimensional drawing of SPEJBL.

The results shown have been obtained with $T_s = 0.58\ s$. The optimal motion is such that the step length is 0.366 m and the optimal velocity is 0.6323 m/s. These values are results of the optimization process.

The normal components of the ground reactions, in function of time, of the stance foot during one half step in single support are presented in figure 3. The average vertical reaction force is 20 N, which is coherent with the weight of the robot which the mass equals 2.1385 Kg. The chosen friction coefficient is 0.7.

The figure 4 shows the evolutions of joint variables $q_i(t)$ $i = 1,...,12$, defined by the third-order spline function presented in Section III, in the single

[1] SPEJBL is a biped robot designed in the Department of Control Engineering of the Technical University in Prague.

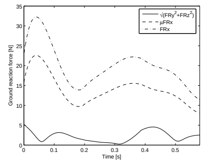
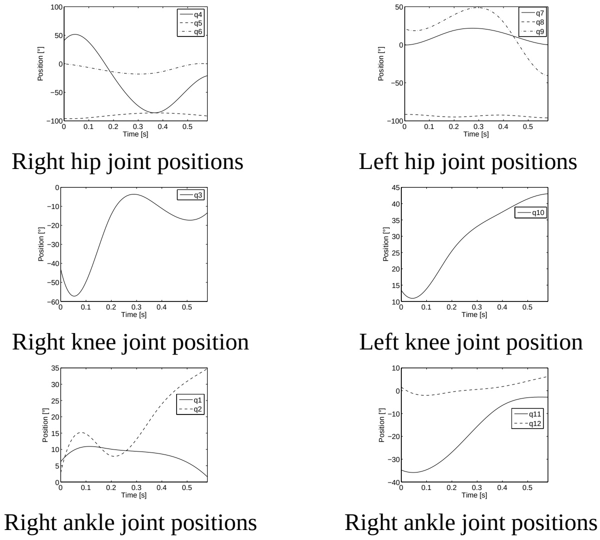


Figure 3: Normal components in the stance foot.



Right hip joint positions

Left hip joint positions

Right knee joint position

Left knee joint position

Right ankle joint positions

Right ankle joint positions

Figure 4: Evolution of joint positions.

support phase during one half step. Let us remark that the evolution of each joint variable depends on the boundary conditions ($\dot{q}_{i,ini}$, $\dot{q}_{i,fin}$ for $i = 1,...,12$) and also on the intermediate configurations ($q_{i,int1}$, $q_{i,int2}$ for $i = 1,...,12$) whose values are computed in the optimal process.

The figure 5 shows the CoP trajectory which is always inside the support polygon determined by $l_p = 0.11\ m$ and $L_p = 0.17\ m$., that is, the robot maintains the balance during the motion. Because the minimal distance between of CoP and the boundary of the foot is large, smaller foot is acceptable for this cyclic motion.

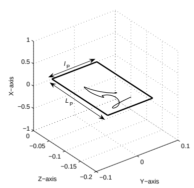


Figure 5: The evolution of CoP trajectory.

For a set of motion velocities, the evolution of J_Γ criterion is presented in figure 6. With respect to the evolution of J_Γ we can conclude that the biped robot

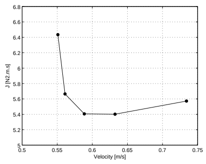


Figure 6: J_Γ in function of several motion velocities for the biped.

consumes more energy for low velocities to generate one half step. Due to the limitations of the joint velocities we could not obtain superior values to 0.73 m/s. The energy consumption increases probably for higher velocity (see (Chevallereau. and Aoustin, 2001)). The robot has been designed to be able to walk slowly, this walk require large torque and small joint velocities. Its design is also based on large feet in order to be able to use static walking, as a consequence the feet are heavy and bulky, thus the resulting optimal motion is close to the motion of a human with snowshoes.

6 CONCLUSION

Optimal joint reference trajectories for the walking of a 3D biped are found. A methodology to design such optimal trajectories is developed. This tool is useful to test a robot design or for the control of the robot. In order to use classical optimization technique, the optimal trajectory is described by a set of parameters: we choose to define the evolution of the actuated relative angle as spline functions. A cyclic solution is desired. Thus the number of the optimization variables is reduced by taking into account explicitly of the cyclic condition. Some inequality constraints such as the limits on torque and velocity, the condition of no sliding during motion and impact, some limits on the motion of the free leg are taken into account. Optimal motion for a given duration of the step have been obtained, the step length and the advance velocity are the result of the optimization process. The result obtained are realistic with respect to the size of the robot under study. Optimal motion for a given motion velocity can also be studied, in this case the motion velocity is consider as a constraint. The proposed method to define optimal motion will be tested on other prototype with dimension closer to human.

REFERENCES

Beletskii, V. V. and Chudinov, P. S. (1977). Parametric optimization in the problem of bipedal locomotion. *Izv. An SSSR. Mekhanika Tverdogo Tela [Mechanics of Solids]*, (1):25–35.

Bessonnet, G., Chesse, S., and Sardin, P. (2002). Generating optimal gait of a human-sized biped robot. In *Proc. of the fifth International Conference on Climbing and Walking Robots*, pages 717–724.

Channon, P. H., Hopkins, S. H., and Pham, D. T. (1992). Derivation of optimal walking motions for a bipedal walking robot. *Robotica*, 2(165–172).

Chevallereau., C. and Aoustin, Y. (2001). Optimal reference trajectories for walking and running of a biped. *Robotica*, 19(5):557–569.

Formal'sky, A. (1982). *Locomotion of Anthropomorphic Mechanisms*. Nauka, Moscow [In Russian].

Grishin, A. A., Formal'sky, A. M., Lensky, A. V., and Zhitomirsky, S. V. (1994). Dynamic walking of a vehicle with two telescopic legs controlled by two drives. *Int. J. of Robotics Research*, 13(2):137–147.

Khalil, W. and Dombre, E. (2002). *Modeling, identification and control of robots*. Hermes Sciences Europe.

L. Hu, C. Z. and Sun, Z. (2006). Biped gait optimization using spline function based probability model. *in Proc. of the IEEE Conference on Robotics and Automation*, pages 830–835.

M. Sakaguchi, J. Furushu, A. S. and Koizumi, E. (1995). A realization of bunce gait in a quadruped robot with articular-joint-type legs. *Proc. of the IEEE Conference on Robotics and Automation*, pages 697–702.

Miossec, S. and Aoustin, Y. (2006). *Dynamical synthesis of a walking cyclic gait for a biped with point feet*. Special issue of lecture Notes in Control and information Sciences, Ed. Morari, Springer-Verlag.

M.W.Walker and D.E.Orin (1982). Efficient dynamic computer simulation of robotics mechanism. *Trans. of ASME, J. of Dynamic Systems, Measurement and Control*, 104:205–211.

Rostami, M. and Besonnet, G. (1998). Impactless sagital gait of a biped robot during the single support phase. In *Proceedings of International Conference on Robotics and Automation*, pages 1385–1391.

Roussel, L., de Wit, C. C., and Goswami, A. (2003). Generation of energy optimal complete gait cycles for biped. In *Proc. of the IEEE Conf. on Robotics and Automation*, pages 2036–2042.

Saidouni, T. and Bessonnet, G. (2003). Generating globally optimised saggital gait cycles of a biped robot. *Robotica*, 21(2):199–210.

Vukobratovic, M. and Stepanenko, Y. (1972). On the stability of anthropomorphic systems. *Mathematical Biosciences*, 15:1–37.

Zonfrilli, F., Oriolo, M., and Nardi, T. (2002). A biped locomotion strategy for the quadruped robot sony ers-210. In *Proc. of the IEEE Conf. on Robotics and Automation*, pages 2768–2774.

REPUTATION BASED BUYER STRATEGY FOR SELLER SELECTION FOR BOTH FREQUENT AND INFREQUENT PURCHASES

Sandhya Beldona and Costas Tsatsoulis
Department of Electrical Engineering and Computer Science, Information and Telecommunication Technology Center
The University of Kansas, 2335 Irving Hill Road, Lawrence, KS 66045, USA
sbeldona@ittc.ku.edu, tsatsoul@ittc.ku.edu

Keywords: Autonomous agents, Learning, Distributed, Trust, Reputation, Ecommerce, Electronic Markets.

Abstract: Previous research in the area of buyer agent strategies for choosing seller agents in ecommerce markets has focused on frequent purchases. In this paper we present a reputation based buyer agent strategy for choosing seller agent in a decentralized, open, uncertain, dynamic, and untrusted B2C ecommerce market for frequent and infrequent purchases. The buyer agent models the reputation of the seller agent after having purchased goods from it. The buyer agent has certain expectations of quality and the reputation of a seller agent reflects the seller agent's ability to provide the product at the buyer agent's expectation level, and its price compared to its competitors in the market. The reputation of the seller agents and the price quoted by the seller agents are used to choose a seller agent to transact with. We compare the performance of our model with other strategies that have been proposed for this kind of market. Our results indicate that a buyer agent using our model experiences a slight improvement for frequent purchases and significant improvement for infrequent purchases.

1 INTRODUCTION

Our work considers decentralized, open, dynamic, uncertain and untrusted electronic market places with seller agents and buyer agents. The seller agents sell products and the quality and the price of product varies across them. The goal for the buyer agent (hereafter referred to as the buyer) is to purchase a product from a seller agent (hereafter referred to as the seller) who meets its expectations of quality and service and to purchase it at the lowest price possible in the market. At the same time the buyer wants to reduce its chances of interacting with dishonest and poor quality seller agents. In an open market, the sellers agents (hereafter referred to as sellers) and the buyers agents (hereafter referred to as buyers) can enter and leave the market anytime. In a dynamic market the players in the market need not exhibit the same behaviour all the time; the sellers can vary the price and the quality in various transactions. Untrusted market implies there could be dishonest sellers in the market. By uncertain market we mean that the buyers can gauge the quality of the product after actually receiving the product. There could be a onetime transaction between the buyer and the seller or multiple transactions between them. There is no limitation on the number of the sellers and the buyers in the market. These characteristics are typical of a traditional commerce market and hence we consider a similar environment for our electronic market.

It is not possible to pre-program an agent to operate under these conditions, or to know beforehand who the best seller for a buyer is, as new sellers are entering the market, the lowest priced seller may not necessarily be the best seller, and sellers could be lying. Agents have to be equipped with abilities to make the most rational decision based on all the information that they can gather. They should be able to learn from their past experiences.

Recent research has developed intelligent agents for ecommerce applications (A. Chavez & P. Maes, 1996), (A Chavez & D.Dreilinger & R.Guttman & P. Maes, 1997), (C. Goldman & S. Kraus & O.Shehory, 2001, p. 166-177), (R.B. Doorenbos & Etzioni & D. Weld, 1997, p. 39-48), (B. Krulwich, 1996, p. 257-263), (T. Tran, 2003), (T. Tran & R.

Cohen, 2004, Vol. 2, p. 828-835), (J.M. Vidal & E.H Durfee, 1996, p. 377-384). However, as Tran (T. Tran, 2003) summarizes, the agents in (R.B. Doorenbos & Etzioni & D. Weld, 1997, p. 39-48), (B. Krulwich, 1996, p. 257-263) are not autonomous, the agents in (A. Chavez & P. Maes, 1996), (A Chavez & D.Dreilinger & R.Guttman & P. Maes, 1997), (C. Goldman & S. Kraus & O.Shehory, 2001, p. 166-177), and (R.B. Doorenbos & Etzioni & D. Weld, 1997, p. 39-48), do not have learning abilities, the agents in (J.M. Vidal & E.H Durfee, 1996, p. 377-384). have significant computational costs, and the agents in (A. Chavez & P. Maes, 1996), (A Chavez & D.Dreilinger & R.Guttman & P. Maes, 1997), (C. Goldman & S. Kraus & O.Shehory, 2001, p. 166-177), (R.B. Doorenbos & Etzioni & D. Weld, 1997, p. 39-48), (B. Krulwich, 1996, p. 257-263), (J.M. Vidal & E.H Durfee, 1996, p. 377-384) do not have the ability to deal with deceptive agents. Tran and Cohen's (T. Tran & R. Cohen, 2004, Vol. 2, p. 828-835) , (T. Tran, 2003) work addressed these shortcomings by developing a strategy for the buying agents using reinforcement learning and reputation modelling of the sellers. However their model builds reputation slowly and the buyer has to interact with a seller several times before the seller is considered reputable. This model works well where the buyer has to make repeated transactions with the sellers during frequent purchases. The performance of this model deteriorates for infrequent purchases as the buyer has to purchase several times from a seller before making its decision about the seller. When the buyer is purchasing a product on an infrequent basis it needs to quickly identify reputed sellers.

We present reputation based modelling of a seller by the buyer which can work for frequent as well as infrequent purchases in a B2C ecommerce market. We compared the performance of the buying agents using our model, reinforcement learning (J.M. Vidal & E.H Durfee, 1996, p. 377-384) and reputation based reinforcement learning (T. Tran & R. Cohen, 2004, Vol. 2, p. 828-835), (T. Tran, 2003). Our results show that the buying agents using our model improved their performance slightly for frequent purchases and showed a significant improvement for infrequent purchases, making our approach better suitable for all kinds of buyers.

2 METHODOLOGY

We consider decentralized, open, dynamic, uncertain and untrusted electronic market places with buyers sellers. The buyers' model the sellers' reputation based on their direct interactions with them. The buyer has certain expectations of quality and the reputation of a seller reflects the seller's ability to provide the product at the buyer's expectation level, and its price compared to its competitors in the market. The buyer's goal is to purchase from a seller who will maximize its valuation of the product, which is a function of the price and quality of the product. At the same time it wants to avoid interaction with dishonest or poor quality sellers in the market. The reputation of the seller is used to weed out dishonest or poor quality sellers.

In this paper we use the following notation: Subscript represents the agent computing the rating. Superscript represents the agent about whom the rating is being computed. The information in the parenthesis in the superscript is the kind of rating being computed. For example, every time the buyer *b* purchases a product from the seller *s* , it computes a direct trust *(di)* rating $T_b^{s(di)}$ of the seller *s* by buyer *b*. The trust rating of seller *s* by buyer *b* is computed as shown in equation 1.

$$T_b^{s(di)} = \begin{cases} \frac{q_{act}}{q_{\exp}} - \left(\frac{p_{act} - p_{avg}}{p_{\max}} \right) if\ q_{act} \geq q_{\min}\ and\ p_{act} \geq p_{avg} & (a) \\ \frac{q_{act}}{q_{\exp}}\ if\ q_{act} \geq q_{\min}\ and\ p_{act} < p_{avg} & (b) \\ \frac{q_{act}}{q_{\exp}} - \left(\frac{p_{act} - p_{\min}}{p_{\max} - p_{\min}} \right) if\ q_{act} < q_{\min} & (c) \end{cases} \quad (1)$$

where q_{act} is the actual quality of the product delivered by the seller *s*, q_{exp} is the desired expected quality and q_{min} is the minimum quality expected by the buyer *b*. p_{act} is the price paid by the buyer *b* to purchase the product from the seller *s*. p_{min} is the minimum price quote, p_{max} is the maximum price quote received and p_{avg} is the average of the price quotes received by the buyer for this product.

The trust rating should be proportional to the degree the quality delivered by the seller meets the buyer's expectations and the price paid to purchase the product. If there are two sellers, *s1* and *s2*, who can meet the buyer's expectation for the quality of the product, and *s1*'s price is lower than *s2*, then *s1* should get a higher rating than *s2*. Similar to (T. Tran, 2003) and (T. Tran & R. Cohen, 2004, Vol. 2, p. 828-835) , we make the common assumption that it costs more to produce a higher quality product. So when considering the price charged by a seller, if the seller meets the buyer's minimum expectation for quality, and if the price is greater than the average price quoted, then the difference between the seller's price and the average price quoted is weighed against the maximum price quoted for that product (part (a) of the equation). On the other hand if the price of the seller is below the average price (which can happen if the other sellers are trying to maximize their profits or there are too many low quality sellers) then the rating for this seller is computed based on its quality alone (part (b) of the equation). If the seller's quality does not meet the buyer's expectation then the difference of seller's price and the minimum price quoted is compared to the difference between the maximum and the minimum price quoted to penalize the seller more severely (part (c) of the equation).

This model makes the assumption that the buyer b expects the highest quality and in the best case q_{act} can be equal to q_{exp} and it costs more to produce higher quality products. From the above equations it can been seen that $T_b^{s(di)}$ ranges from *[-1, 1]*. In the best case, b gets the expected quality at the lowest price and $T_b^{s(dimax)} = 1$. In the worst case $q_{act} = 0$ and b pays the maximum price quoted and $T_b^{s(dimin)} = -1$.

If the buyer has not interacted with the seller then $T_b^{s(di)} = 0$ for that seller and such a seller is referred to as a new seller.

Whenever the buyer b is evaluating a list of sellers for purchase decisions it computes $T_b^{s(diavg)}$, the average rating for each seller s from its past interactions. $T_b^{s(diavg)}$ is computed as the weighted mean of its past n recent interactions.

$$T_b^{s(diavg)} = \frac{1}{W}\sum_{i=1}^{n} w_i T_{b(i)}^{s(di)} \tag{2}$$

where

$$w_i = \frac{t_{cur}}{t_{cur} - t_i} \tag{3}$$

$$W = \sum_{i=1}^{n} w_i \tag{4}$$

where $T_{b(i)}^{s(di)}$ is the rating computed for a direct interaction using equation 1.Subscript i in parenthesis indicates the i^{th} interaction. w_i is the importance of the rating in computing the average. Recent ratings should have more importance. Hence the weight of a rating is inversely proportional to the difference between the time a transaction happened t_i to the current time t_{cur}.

The buyer has threshold values θ and ω for the direct trust ratings to indicate its satisfaction or dissatisfaction with the seller respectively. The threshold values θ and ω are set by the buyer and $\theta > \omega$ and θ and ω are in the range *[-1, 1]*. The buyer chooses sellers whose average direct trust rating is greater than or equal to θ and considers them to be reputable, does not choose sellers whose average direct trust rating is less than or equal to ω and considers them to be disreputable. It is unsure about sellers whose average direct trust ratings are between ω and θ and will consider them again only if there are no reputable or new sellers to consider. From the list of sellers who have submitted price bids, reputable sellers whose $T_b^{s(diavg)}$ is above the satisfaction threshold θ are identified as potential sellers. The buyer includes new sellers into the list of potential sellers to be able to quickly identify a good seller.

The buyer's valuation function for the product is a function of the price a seller is currently quoting and the quality that has been delivered in the past . For a seller with whom the buyer has interacted before, the quality is the average of the quality delivered in the past interactions. For a seller with whom the buyer has not interacted directly, the quality is set to the expected quality. From the list of potential sellers, the buyer chooses a seller who maximizes its product valuation function.

3 RELATED WORK

We compare our model to (T. Tran, 2003), (T. Tran & R. Cohen, 2004, Vol. 2, p. 828-835) and (J.M. Vidal & E.H Durfee, 1996, p. 377-384) as their and our work consider a similar market environment with autonomous buying agents who learn to identify seller agents to transact with. (J.M. Vidal & E.H Durfee, 1996, p. 377-384) use reinforcement learning strategy and (T. Tran, 2003) and (T. Tran & R. Cohen, 2004, Vol. 2, p. 828-835) use reinforcement learning with reputation modelling of sellers. Our model provides a different method of computing reputation and does not use reinforcement learning strategy.

Vidal and Durfee's (J.M. Vidal & E.H Durfee, 1996, p. 377-384) economic model consists of seller and buyer agents. The buyer has a valuation function for each good it wishes to buy which is a function of the price and quality. The buyer's goal is to maximize its value for the transaction. Agents are divided into different classes based on their modelling capabilities. 0-level agents base their actions on inputs and rewards received, and are not aware that other agents are out there. 1-level agents are aware that there are other agents out there, and they make their predictions based on the previous actions of other agents. 2-level agents model the beliefs and intentions of other agents. 0-level agents use reinforcement learning. The buyer has a function f for each good that returns the value that the buyer expects to get by purchasing the good at price p. This expected value function is learned using reinforcement learning as $f = f + \alpha(v - f)$ where α is the learning rate, initially set to 1 and reduced slowly to minimum value. The buyer picks a seller that maximizes its expected value function f. Our market model is extended into a more general one by having sellers offer different qualities and by the existence of dishonest sellers in the market. The buyers use the reputation of the sellers to avoid dishonest sellers and reduce their risks of purchasing low quality goods. The reputation of the sellers is learned based on direct interactions.

Tran and Tran and Cohen develop learning algorithms for buying and selling agents in an open, dynamic, uncertain and untrusted economic market (T. Tran, 2003) and (T. Tran & R. Cohen, 2004, Vol. 2, p. 828-835) . They use Vidal and Durfee's (J.M. Vidal & E.H Durfee, 1996, p. 377-384) 0-level buying and selling agents. The buying and selling agents use reinforcement learning to maximize their utilities. They enhance the buying agents with reputation modelling capabilities, where buyers model the reputation of the sellers. The reputation value varies from -1 to 1. A seller is considered reputable if the reputation is above a threshold value. The seller is considered disreputable if the reputation value falls below another threshold value. Sellers with reputation values in between the two thresholds are neither reputable nor disreputable. The buyer chooses to purchase from a seller from the list of reputable sellers. If no reputable sellers are available, then a seller from the list of non disreputable sellers is chosen. Initially a seller's reputation is set to 0. The seller's reputation is updated based on whether the seller meets the demanded product value. If the seller meets or exceeds the demanded product value then the seller is considered cooperative and its reputation is incremented. If the seller fails to meet the demanded product value then the seller is considered uncooperative and its reputation is decremented. This model builds reputation slowly. So the buyer has to interact with a seller several times before the reputation of the seller crosses the threshold value. This model works well where the buyer has to make repeated transactions with the sellers, but a buyer cannot utilize this model when making infrequent purchases.

4 EXPERIMENTS AND RESULTS

For our experiments we developed a multi-agent based simulation of an electronic market with autonomous buying agents, selling agents, and a matchmaker. The sellers upon entering the market register with a matchmaker (D. Kuokka & L. Harada , 1995) regarding the products that they can supply. When a buyer wants to purchase a product, it obtains a registered list of sellers selling this product from the matchmaker and sends a message to each of the sellers in the list to submit their bids for the product p. The sellers who are interested in getting the contract submit a bid which includes the price. The buyer waits for a certain amount of time for responses and then evaluates the bids received to choose a seller to purchase from.

The following parameters were set. The quality q sold across the sellers ranges from [10, 50] and varies in units of 1. The buyer expects a minimum quality of 40($q_{min} = 40$). The price of a product for an honest seller is $pr = q \pm 10\%q$. Like Tran (T. Tran, 2003) we make the assumption that it costs more to produce high quality goods. We also make the reasonable assumption that the seller may offer a discount to attract the buyers in the market or raise its price slightly to increase its profits. Hence the price of the product is set to be in the range of 90% -110% of the quality for an honest buyer. A dishonest buyer on the other hand may charge higher prices. The buyer's valuation of the product is a function of the quality and the price and for our simulation we set it as $3 * quality - price$. The buyer's valuation function reflects the gain, a buyer makes from having purchased a product from a seller. Each time a buyer purchases a product from a seller its product valuation is computed and we consider this as the buyer's gain for having purchased from that seller.

We compared the performances of four buyers .

1. *F&NFBuyer*: - This buying agent uses the buying strategy as described in our model. The buyer's desired expected quality is $q_{exp} = 50$. The acceptable quality for a buyer is from [40, 50]. The non acceptable quality is from [10-39]. The maximum price p_{max} quoted by honest seller would be 55 and the minimum price p_{min} quoted would be 9. The average price p_{avg} would be 32. The threshold values θ for a seller to be considered reputable and ω for a seller to be considered disreputable values can be computed as follows:

The buyer is expecting at least a quality of 40. In the worst case it can get this at the highest price that can be charged by a honest seller which would be 44. From equation 1(a) the trust rating for that seller would be

$$\frac{40}{50} - \left(\frac{44-32}{55}\right) = 0.581 \quad (5)$$

so we set $\theta = 0.58$. For new sellers the trust rating is set to 0. These buyers should not come under the category of disreputable sellers. So we set the threshold value for a seller to be considered unacceptable as -0.1. So $\omega = -0.1$

2. *Tran Buyer:* - This buying agent uses the buying strategy as described in Tran and Cohen [8]. The threshold for seller to be considered reputable is set to 0.5 and for seller to be considered disreputable is set to -0.9 as described in their work.
3. *RL Buyer:*- This buying agent uses a reinforcement learning strategy as described for 0-level buying agent in Vidal and Durfee [9].
4. *Random Buyer:*- This buying agent chooses a buyer randomly.

We populated the market with 12 sellers belonging to one of the six categories with the price and quality properties as shown (two agents per category):

1. *Honest Acceptable (HA):* - Each seller offers a quality in the range [40-50]. The price is between 90-110% of the quality they are selling.
2. *Honest Not Acceptable (HNA):* - Each seller offers a quality in the range[10-39]. Their price is between 90 -110% of the quality they are selling.
3. *Overpriced Acceptable (OPA):*- Each seller offers a quality in the range [40-50]. The price is between 111-200% of the quality they are selling.
4. *Overpriced Not Acceptable (OPNA):* - Each seller offers a quality in the range [10-39]. Their price is between 111-200% of the quality they are selling.
5. *Inconsistent:* - Each seller offers a quality in the range [10-50]. The price is between 90-110% of the quality they are selling.
6. *Dishonest:* - This category of sellers in their first sale to a buyer offer acceptable quality q [40-50] charging a price $pr = q \pm 10\%q$. In their subsequent sales to that buyer they reduce the quality q to be in the range [10-25]. However their price still remains high. Price $pr = q1 \pm 10\%q1$ where $q1$ is in the range [40 -50].

The data from the experiments was collected over 100 simulations. In each simulation, each buying agent conducted 500 transactions. In each transaction they purchased product p by querying the seller list from the matchmaker, obtain price quotes from different sellers and utilize their buying strategy to choose a seller. We compared the performances of the various buying agents on the following parameters.

- How long it took them to learn to identify high quality low priced sellers. We want the buying agents to identify high quality sellers offering low prices as soon as possible. If the buyer is able to identify high quality sellers quickly then the same strategy can be used when making infrequent purchases.
- The average gain as the number of purchases of product p is increased. If the average is consistently high means that the buyer is interacting with high quality sellers offering low prices most often. If the average gain is high earlier on implies that the buyer has identified high quality low price sellers quickly.

Figures 1-3 show the gain versus transactions for each type of buyer (because of space considerations we are not showing the plot of the gain vs. a random buyer, since the gain simply constantly fluctuates):

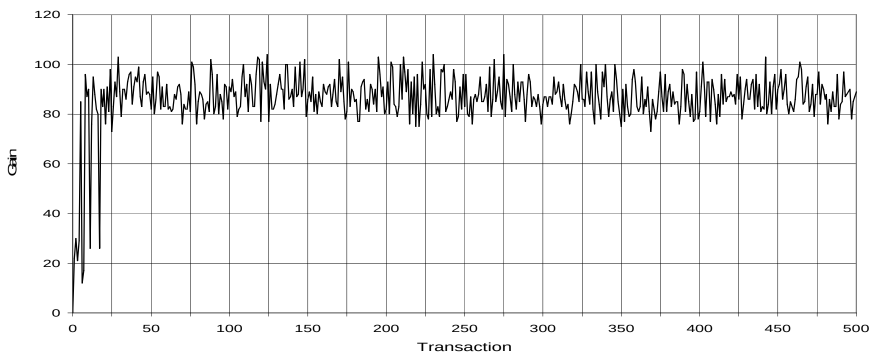


Figure 1: Gain Vs Transaction for a F&NF Buyer.

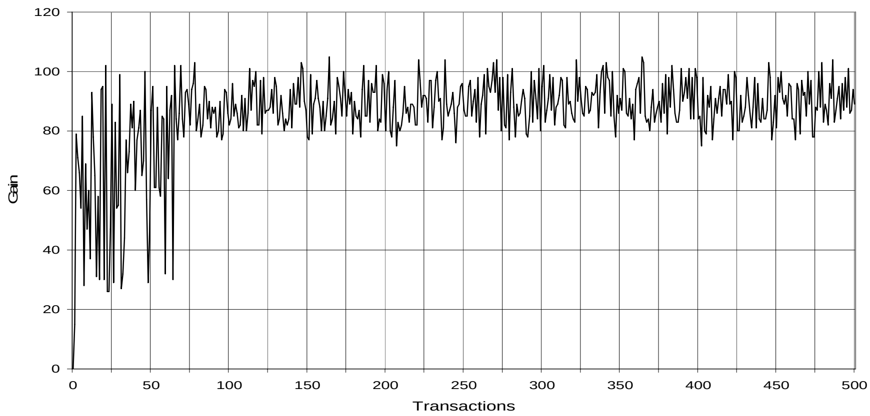


Figure 2: Gain Vs Transaction for a Tran Buyer.

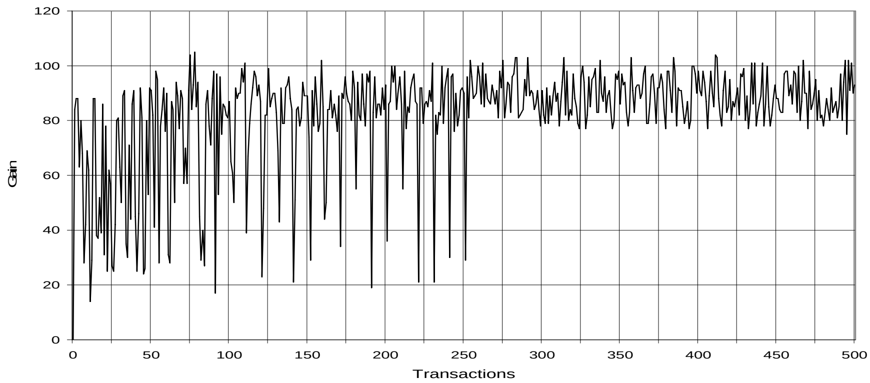


Figure 3 : Gain Vs Transaction for a RL Buyer.

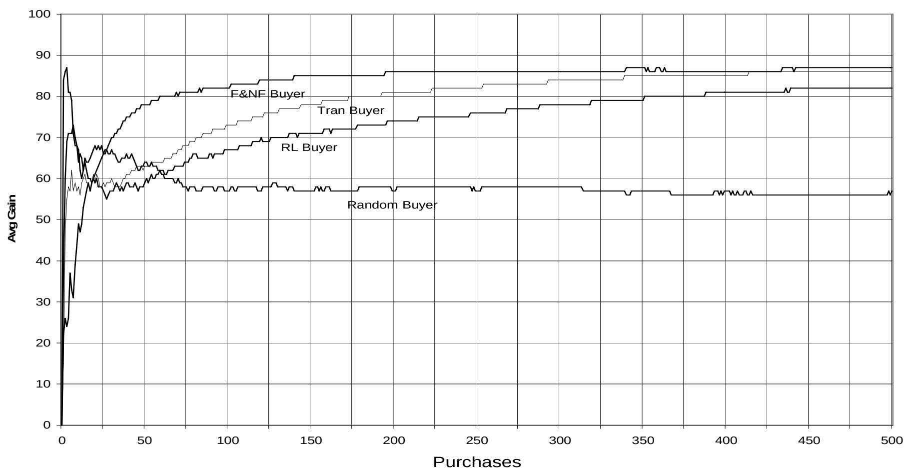


Figure 4: Average Gain versus Number of Purchases for different buyers.

Table1 shows the number of purchases made by a buyer from each seller type.

Table 1: Buyer seller interaction.

	HA	HNA	OPA	OPNA	INC	DIS
Rsk Buyer	488	2	2	2	2	4
Tran Buyer	451	7	23	5	8	6
RL Buyer	420	16	15	13	17	16
Random Buyer	86	88	82	83	69	92

Acceptable quality sellers can offer qualities anywhere between 40-50. The lowest gain from purchasing from a honest seller offering at the lowest end of good quality range and charging its highest price is 76 (3*40 – 44). When the gain from purchasing from a seller is 76 and above, it means the buyer is purchasing from a high quality low priced seller. From figures 1-3 it can be seen that F&NF Buyer, Tran Buyer and RL Buyer learn although at different rates to identify high quality low priced sellers. After having learned, they consistently interact with high quality low priced sellers. This is confirmed by the fact that highest number of purchases are made from honest acceptable sellers as shown in table 1. Random Buyers never learn and that is to be expected as they are choosing sellers randomly. F&NF Buyer learns to identify high quality low priced sellers very quickly in about 15 transactions or purchases. Tran Buyers take about 60 transactions to learn and RL Buyer learns in about 250 transactions. If the buyers were to purchase the product infrequently then the F&NF Buyer strategy would work better than the RL Buyer or Tran Buyer strategy as it requires the least number of transactions to learn.

Figure 4 shows the average gain versus the number of purchases for different buyers.

In the beginning, average gains are fluctuating as the buyers employing a non-random strategy are learning and Random Buyer is choosing sellers randomly. F&NF Buyer is the quickest to learn and its average gain raises sharply earlier on compared to the other two learning agents. As RL Buyer takes a long time to learn, its average gain at the end is still lower than the F&NF or Tran Buyer. Since Random Buyer purchases randomly from various types of sellers, its average is consistently the lowest. In the first half of the figure 4 it can be seen that when the purchases are fewer, the average gain for the F&NF Buyer, once its learning phase is completed, is higher than the other buying agents. So, if the buyers were to purchase the product infrequently, then the F&NF Buyer strategy works better than the RL or Tran Buyer strategy. As the number of purchases increases, F&NF Buyer still has the highest average gain with the Tran Buyer's average gain coming very close to it at very high number of purchases.

5 CONCLUSIONS AND FUTURE WORK

We presented a model for a buyer to maintain the seller reputation and strategy for buyers to choose sellers in a decentralized, open, dynamic, uncertain and untrusted multi-agent based electronic markets. The buyer agent computes a seller agent's reputation based on its ability to meet its expectations of product, service, quality and price as compared to its competitors. We show that a buying agent utilizing our model of maintaining seller reputation and buying strategy does better than buying agents employing strategies proposed previously for frequent as well as for infrequent purchases. For future work we are looking at how the performance of buying agent can be improved for extremely infrequent purchases.

REFERENCES

A. Chavez, and P. Maes, 1996, Kasbah: An Agent Marketplace for Buying and SellingGoods. *In Proceedings of the 1st Int. Conf. on the Practical Application of Intelligent Agents and Multi-Agent Technology*, London.

A Chavez, D. Dreilinger, R. Guttman. And P. Maes, 1997, A real-Life Experiment in Creating an Agent Marketplace, *In Proceedings of the Second International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology.*

C. Goldman, S. Kraus and O. Shehory, 2001, Equilibria strategies: for selecting sellers and satisfying buyers**,** *Lecture Notes in Artificial Intelligence*, Vol. 2182, M. Klusch and F. Zambonelli (Eds.), Springer, 166-177.

R.B Doorenbos, Etzioni, and D. Weld, 1997, A Scalable Comparison-Shopping Agent for the World Wide Web, *In Proceedings of the First International Conference on Autonomous Agents*, 39-48.

B. Krulwich, 1996, The bargainfinder agent:comparision price shopping on the Internet, *Bots, and other Internet Beasties,* J. Williams, editor, 257-263, Macmillan Computer Publishing.

D. Kuokka and L. Harada, 1995, Supporting Information Retrieval via Matchmaking, *Working Notes of the AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments*.

T. Tran, 2003, Reputation-oriented Reinforcement Learning Strategies for Economically-motivated Agents in Electronic Market Environment, *Ph.D. Dissertation*, University of Waterloo.

T. Tran and R. Cohen, 2004, Improving User Satisfaction in Agent-Based Electronic Marketplaces by Reputation Modeling and Adjustable Product Quality, *Proc. of the Third Int. Joint Conf. on Autonomous Agents and Multi Agent Systems (AAMAS-04),* Volume 2, 828-835.

J.M. Vidal and E.H. Durfee, 1996, The impact of Nested Agent Models in an Information Economy, *Proc. of the Second Int. Conf. on Multi-Agent Systems,* 377-384.

BLENDING TOOL PATHS FOR G^1-CONTINUITY IN ROBOTIC FRICTION STIR WELDING

Mikael Soron and Ivan Kalaykov
Department of Technology, Örebro University, Fakultetsgatan, Örebro, Sweden
mikael.soron@tech.oru.se, ivan.kalaykov@tech.oru.se

Keywords: Robotics, friction stir welding, path planning.

Abstract: In certain robot applications, path planning has to be viewed, not only from a motion perspective, but also from a process perspective. In 3-dimensional Friction Stir Welding (FSW) a properly planned path is essential for the outcome of the process, even though different control loops compensate for various deviations. One such example is how sharp path intersection is handled, which is the emphasis in this paper. We propose a strategy based on Hermite and Bezier curves, by which G^1 continuity is obtained. The blending operation includes an optimization strategy in order to avoid high second order derivatives of the blending polynomials, yet still to cover as much as possible of the original path.

1 INTRODUCTION

Being a quite recent welding process, Friction Stir Welding (FSW), has been announced 'the next big thing' in the welding community since its release on the market a decade ago. Since its invention at The Welding Institute (TWI) in 1991 (patent filed in 92 (Thomas et al., 1991)), it has been declared a superior welding technique, in term of quality and repeatability, compared to traditional fusion welding techniques, for joining aluminium alloys.

The high quality of the FS weld is a result from a non-melting welding process without the use of filler material. And by the means of quality, it is defined through elimination (or close to) of shrinkage, porosity and distortions, which are quality issues in traditional fusion welding (Chao et al., 2003; Lomolino et al., 2005).

From an application point of view, the quality of the weld and the robustness of the process, has appealed certain types of industries with extreme demands on the results of the joining process. Two examples are the aerospace industry and in nuclear waste management (Cederqvist and Andrews, 2003), which both has an obvious zero tolerance for error. But the application does not necessarily have to be extreme to benefit from FSW, as a typical application is joining extruded aluminium profiles.

But regardless of the application area, the process is constrained by a few factors. One being the large forces and torques needed to achieve a good result. Depending on the type of material used and its thickness, a certain amount of force is applied to produce heat (by friction). These forces normally range from a few kN up to 200 kN, which calls for heavy duty machinery to support the process. And as a result from applying large forces, the object needs to be rigidly clamped in a fixture and supported by a backingbar. If the object is capable of supporting the applied forces using its internal structure, the backingbar may not be needed.

The issue of self support can in some cases be eliminated through engineering. That is, by designing the pieces properly for FSW and selecting a location of joining where it is most suited. The machinery issue, on the other hand, can not be solved with one single track. The extreme range in forces, calls for multiple solutions. A machine capable of applying 200 kN, is most certainly oversized for application in the lower region of FSW. And another issue is of course flexibility as well as the dextrous workspace of the machine. A machine capable of applying 200 kN in one direction does not necessarily have the power to apply its forces in a arbitrary direction, if even the possibility to reach it, which limits the application and lowers the flexibility.

As a complement in the lower region of FSW (from a force perspective), industrial robots have been introduced. The 3-dimensional workspace of the robot as well as the relative low cost, are indeed appealing and several research project has aimed to develop a robotic solution. One, using a parallel designed robot (Strombeck et al., 2000), while the other attempts made used a serial designed robot (Smith, 2000; Soron and Kalaykov, 2006).

Even though the robots used in these applications were the ones having the greatest pay-load capabilities, the robot applications are limited in terms of material, thickness and speed due to the lack of capability to produce forces greater than approximately 10 kN. Along with the ability to apply forces, comes necessity to handle positional errors due to compliance. Both in the main direction (into the object) as well as in the plane perpendicular to the main direction.

To handle the compliance issues (especially in the main direction) force control is normally implemented in all FSW machines, robot as well as standard machines. In robotics, a standard solution is a force/position hybrid control (Raibert and Craig, 1981), enabling position based control in one or more direction, while achieving a desired force value in the other. A solution that fits FSW, since a path may be followed having a constant contact force with the material.

2 PATH GENERATION FOR FSW

Path generation has for decades been a well investigated topic for application areas such as milling (Elber and Cohen, 1993; Choy and Chan, 2003) and cutting (Wings and Jütter, 2004). To create a high defined path in such applications without an off-line programming (or CAM) tool, can be considered a time consuming operation, if not impossible in many cases.

The resemblance between mentioned processes and FSW is in many areas high, where:

- In all processes a tool is in contact with an object.
- The tool must follow a pre-defined path (or contour) with high accuracy.
- The tool's orientation must be defined at all time also often with high accuracy.

To achieve a satisfying result in such process, using an industrial robot as manipulator instead of an NC machine, a set of new motion control algorithms are essential. Typically, guidance by (3D) vision or force/torque sensing is used to compensate for the lack of positional accuracy or to gain robustness (Kim et al., 2003; Motta et al., 2001).

As mentioned earlier, force control is often used in robotic FSW applications to handle compliance issues. But also when compliance is not in play, as with traditional FSW machines, force control can increase the success rate by handling the occurrence of small deviations in the object's surface as well as indirect control the heat generation (Venable et al., 2004).

In FSW, one of the main reasons for implementing a robot application is the support of a 3-dimensional workspace. This implies that the main use of robots in FSW should be in application containing complex weld paths. At least, these are the application that benefits from a robotic solution. And as complex weld paths are in focus, so becomes the need to create those, preferably as simple as possible.

Another interesting aspect is the fact that one normally gains robustness, in force controlled motions, by having an accurate reference path. Ill-defined paths obviously need a higher amount of corrections, which the control system might be incapable of handling. And as (more) errors are introduced, new ones may arise as a result of the old.

The use of CAD objects in the path generation process is a well known technology, especially in NC applications. In robotics, on the other hand, the emphasis has been more in the area of factory modeling. The ability to visualize the robot cell (and the robot's motion), measuring cycle times as well avoiding collisions, are features supported in most of today's off-line programming (OLP) tools. But the features to model and improve a path, based on underlying CAD data and the application at hand, are often modest. Simple features, such as path blending algorithms and tool orientation control, are often hidden within more general path planning algorithms, if included at all. Even though, they are essential for many of today's robot applications, including FSW.

On most robotic systems, a path smoothing operation exists. Such operations may be defined as zones (Norrlöf, 2003) in which the control system is allowed to re-author the path in order to reach motion continuity. Such implementation are definitely helpful, e.g. when executing way-point motions, but generally not useful from a path planning perspective. Even as the FSW process is bounded to the motion of the robot, the process constraints are not accounted for in the motion controller, leaving the operator to pure guesses at the path designing stage.

3 PATH BLENDING TECHNIQUES

Blending of curves or surfaces are well known techniques when joining simple primitives to construct complex shapes. Instead of using only one equation to express a complex shape, which may lead to a high degree polynomial, it becomes natural to use several, less complex sub-shapes. And in order to maintain G^n continuity, we apply blending techniques. In terms of constructing a robot path, it becomes even more obvious since the continuity is directly linked to the ability to execute the motion. Also, in order to generate motion commands for the robot, we are restricted to use simple path primitives such as lines and arcs.

To represent a curve, with the lowest possible complexity, still offering good enough flexibility we use cubic polynomials. Expressed on a parametric form we get:

$$C(u) = (x(u), y(u), z(u)) \tag{1}$$

where $u \in [0,1]$ and

$$\begin{aligned} x(u) &= a_x u^3 + b_x u^2 + c_x u + d_x \\ y(u) &= a_y u^3 + b_y u^2 + c_y u + d_y \\ z(u) &= a_z u^3 + b_z u^2 + c_z u + d_z \end{aligned} \tag{2}$$

giving the compact form as:

$$C(u) = \mathbf{a}u^3 + \mathbf{b}u^2 + \mathbf{c}u + \mathbf{d} \tag{3}$$

The tangent vector at u of the curve C is obtained by differentiating $C(u)$ as:

$$\frac{dC(u)}{du} = \left(\frac{dx(u)}{du}, \frac{dy(u)}{du}, \frac{dz(u)}{du}\right), \tag{4}$$

or as in the case with a cubic polynomial:

$$\frac{dC(u)}{du} = 3\mathbf{a}u^2 + 2\mathbf{b}u + \mathbf{c} \tag{5}$$

which are the parametric derivatives.

3.1 Hermite Blending

A cubic Hermite curve is defined based on the segments two end-points and the tangent vectors at those points. Recalling the equations 3 and 5 we obtain the following four equations:

$$\begin{aligned} C(0) &= \mathbf{d} \\ C(1) &= \mathbf{a} + \mathbf{b} + \mathbf{c} + \mathbf{d} \\ C'(0) &= \mathbf{c} \\ C'(1) &= 3\mathbf{a} + 2\mathbf{b} + \mathbf{c} \end{aligned} \tag{6}$$

as we have substituted u for 0 and 1 respectively. Solving the equation for the four unknowns we get:

$$\begin{aligned} \mathbf{a} &= 2C(0) - 2C(1) + C'(0) + C'(1) \\ \mathbf{b} &= -3C(0) + 3C(1) - 2C'(0) - C'(1) \\ \mathbf{c} &= C'(0) \\ \mathbf{d} &= C(0) \end{aligned} \tag{7}$$

And by substituting the algebraic coefficients, we get:

$$\begin{aligned} C(u) = &(2u^3 - 3u^2 + 1)C(0) \\ &+(-2u^3 + 3u^2)C(1) \\ &+(u^3 - 2u^2 + u)C'(0) \\ &+(u^3 - u^2)C'(1) \end{aligned} \tag{8}$$

In a blending algorithm we may select $C(0)$ and $C(1)$ from the two intersecting curves to join them with a C^1 continuity. In such case, the only thing affecting the output of the curve is the location of those end-points. This may of course lead to unwanted phenomena in the practical case due to high second order derivatives. But if we release the C^1 continuity in order to affect the blending curve by changing the magnitude of the tangent vectors, we get another variable, k. By doing so, we still have geometric first order continuity, G^1, and a more flexible algorithm for blending. We then form the blending function as:

$$\begin{aligned} C(u) = &F_1(u)\mathbf{p}_0 + F_2(u)\mathbf{p}_1 \\ &+F_3(u)k\mathbf{t}_0 + F_4(u)k\mathbf{t}_1 \end{aligned} \tag{9}$$

where the F_n are the Hermite basis functions (Martens, 1990) substituted from equation 8, the $\mathbf{p}_n$ are the intersecting control points and $\mathbf{t}_n$ are the tangent vector at $\mathbf{p}_n$.

3.2 Bezier Blending

Yet another type of Spline curve, often used in blending operation, is the Bezier curve(Bezier, 1986). In difference to the Hermite curve, which interpolates all its control points, Bezier curves only interpolate the first and last. The usage of the tangents at the end-points is similar to the Hermite, but the basis functions of a Bezier curve are based on a Bernstein polynomial as:

$$C(u) = \sum_{i=0}^{n} \mathbf{p}_i \frac{n!}{i!(n-i)!} u^i (1-u)^{n-i} \tag{10}$$

for $u \in [0,1]$.

In the case of smoothing a sharp intersection, we now use three control points, giving the Bezier equation:

$$C(u) = (1-u)^2\mathbf{p}_0 + 2u(1-u)\mathbf{p}_1 + u^2\mathbf{p}_2 \tag{11}$$

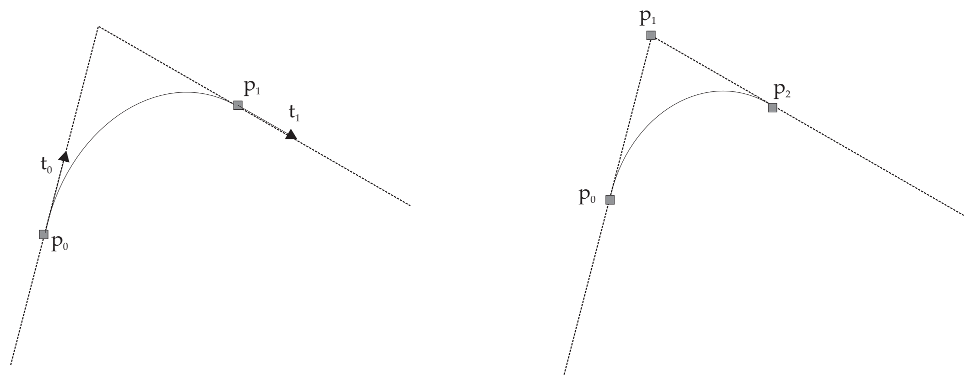


Figure 1: On the left a Hermite smoothing using two control points and its tangents. On the right a Bezier smoothing based on three control points.

As with the Hermite curve, the key to influence the behavior of the curve is through the basis functions. With Bezier curve we use the expression of weights at the control points. The weight is generalized through specifying multiple-coincident points at a vertex (control point), by which we pull the curve closer to that control point. This feature is considered desirable, due to its straight forward implementation, making easier to predict the output of the curve, than in e.g. the Hermite curve. Another important feature of the Bezier curve is that it is contained within its convex hull.

4 IMPLEMENTATION

In our path generation system we implement both Hermite blending as well as Bezier blending, to allow two different approaches towards the same goal. The objective is of course to create G^1 continuous path for the robot to execute, with only a limited interaction with operator/programmer.

In both cases we allow the user to define a minimum and maximum distance from the intersection point, where the blending function should be applied. The idea behind having an interval in which the blending is performed is to allow the system to optimize its blending performance. In the case with the Hermite curve we also allow an interval for the magnification of the tangent vector from $[0,k]$, while the Bezier is allowed to put weight on the intersecting control point, $\mathbf{p}_1$.

Since neither Hermite or Bezier curves have any corresponding motion implementation on the robot's control system, the blending motion needs to be executed as a circular motion (or rather as a sequence of consecutive linear/circular motions). Therefore, at each iteration of the blending optimization, the curve is converted into circular path segments, which endpoints tangent's are evaluated as well as the segments second order derivative. To avoid a too time consuming operation, we introduce discrete steps on which we perform the optimization.

4.1 Blending Sequence

The following sequence is used in the Hermite blending operation:

1. Set initial variables for magnification ($m = 1$), error ($\varepsilon_0 = \infty$), optimization tolerance (μ) and distance interval ($d \in [d_{min}, d_{max}]$)
2. Define the intersection to blend through original path segments, $C_0(u)$ and $C_1(u)$
3. If $d < d_{max}$ exit
4. Extract control points at $\mathbf{p}_0 = C_0(1-d)$ and $\mathbf{p}_1 = C_1(d)$
5. Calculate the tangents, $\mathbf{t}_0$ and $\mathbf{t}_1$, at $\mathbf{p}_0$ and $\mathbf{p}_1$
6. Create the blending curve, C, and calculate highest, $|C''|$.
7. If $\varepsilon_i < \varepsilon_{i-1}$ increase m jump to 5.
8. If $\varepsilon > \mu$ reset m increase d and jump to 3.
9. Exit

By applying this strategy we iterate until the tangent error is less than the optimization tolerance. If the distance interval is zero, $d_{min} = d_{max}$ we only optimize locally on the magnification factor.

When using the Bezier method we end up with a similar approach, namely:

1. Set initial variables for weight ($w = 0$), error ($\varepsilon_0 = \infty$), optimization tolerance (μ) and distance interval ($d \in [d_{min}, d_{max}]$)

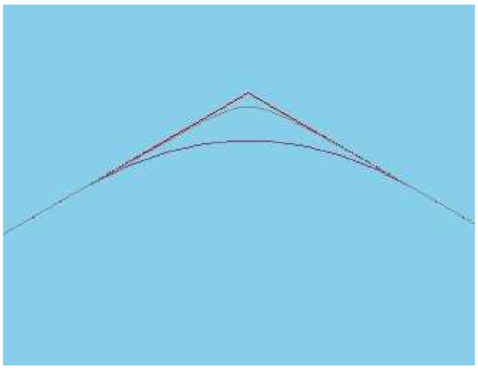


Figure 2: The resulting curves after blending.

2. Define the intersection to blend through original path segments, $C_0(u)$ and $C_1(u)$
3. If $d < d_{max}$ exit
4. Extract control points at $\mathbf{p}_0 = C_0(1-d)$, $\mathbf{p}_1 = C_1(0)$ and $\mathbf{p}_2 = C_1(d)$
5. Create the blending curve, C, and calculate highest, $|C''|$.
6. If $\varepsilon_i < \varepsilon_{i-1}$ increase m jump to 4.
7. If $\varepsilon > \mu$ reset m increase d and jump to 3.
8. Exit

In this approach we still iterate through an inner and an outer loop. But instead of evaluating the inner loop on magnification we evaluate the weight, w.

One remark should be that the Bezier curve has its convex hull property, which does not allow it to expand outside volume of the control points. This is however not the case with the Hermite, which might exhibits undesirable characteristics, such as loops and cusps (Mortenson, 2006).

5 SIMULATION

In this experiment we compare the output of our two strategies in a path modeling environment. We will not declare one better than the other, but simply visualize the output curves using the same input path segments. By using the same input parameters for the common variables, such as distance interval and tolerance level.

The two input segments are defined in the xy plane through three control points as:

$$\begin{aligned} \mathbf{p}_1 &= [0,0,0] \\ \mathbf{p}_2 &= [0,100,0] \\ \mathbf{p}_3 &= [100,100,0] \end{aligned}$$

where $\vec{s}_1 = \mathbf{p}_2 - \mathbf{p}_1$ and $\vec{s}_2 = \mathbf{p}_3 - \mathbf{p}_2$. The two segments $\vec{s}_1$ and $\vec{s}_2$ correspond to two parameterized curves, denoted $C_1(u)$ and $C_2(u)$ with $u \in [0,1]$.

As input parameters to our blending algorithm we choose:

- A bending distance interval ranging from $d_{min} = 10$ to $d_{max} = 50$.
- A tangent magnification factor of $k_{max} = 5$ (for Hermite).
- A weight of $w_{max} = 5$ (for Bezier).

The resulting intersection blending of the two segments, $\vec{s}_1$ and $\vec{s}_2$, is shown in Figure 2. Reviewing the two operations, one may notice that the Bezier algorithm, due to its convex hull property, deviates less from the original path than the Hermite algorithm. The limiting factor in the Bezier case is rather the rapid increase of the second order derivative. In the case of the Hermite operation, the change of the magnification factor pulls the curve closer to (or further away from) the intersecting control point. But in this case, overexceeding the limits causes an overshot of the curve. The visual feedback in such case, is a helping factor in the path planning stage.

One may conclude the simulation with the following remarks:

- The Bezier algorithm allows less deviation from the original path, but the second order derivatives needs to be monitored. One may also mention the straight-forward solution to manipulate the curve by adding weights at control points.
- The Hermite solution have a more complex approach to manipulation, but have on the other hand a visual response when overexceeding limits.

6 CONCLUSIONS

In this paper we have suggested two methods on how to blend path segments to create G^1 continuous paths. There certainly exists numerous of way to do this, but the lack of implementations calls for such studies. In our solution we provide the following:

- Two different implementations, Bezier or Hermite blending.
- Allowing the user to define the location (or location interval) of the blending function.
- A local optimization based on vector magnitude (Hermite) or weight (Bezier).
- A global optimization on the interval (if existing).

One issue that occurs in this type of implementations is that the blending function needs to be converted into executable motion, since spline motions rarely exists in the control system of the robot. One could probably create better implementations through subdividing the splines until a certain level of accuracy is obtained, but this is, however, not within the scope of this implementation.

The idea of intersection blending enlightens only a portion of all issues involved in path generation. It is, however, an essential issue in order to achieve good weld result in 3-dimensional FSW, since the process output is depending on a continuous motion of the welding tool.

ACKNOWLEDGEMENTS

The first author's research is financed by the KK-foundation and ESAB AB Welding Equipment. Their support is highly appreciated. The first author is grateful for the technical support provided by the project members ESAB AB Welding Equipment, ABB Robotics and the research centre for Applied Autonomous Sensor System (AASS) at Örebro University.

REFERENCES

Bezier, P. (1986). *The Mathematical Basis of the UNISURF CAD System*. Butterworth-Heinemann, Newton, MA, USA.

Cederqvist, L. and Andrews, R. E. (2003). A weld that last for 100000 years:fsw of copper canisters. In *CD-ROM Proceedings of the 4th International Symposium on Friction Stir Welding, Park City, Utah*.

Chao, Y. J., Qi, X., and Tang, W. (2003). Heat transfer in friction stir welding - experimental and numerical studies. *Manufactoring Science and Engineering*, 125:138–145.

Choy, H. and Chan, K. W. (2003). Modeling cutter swept angle at cornering cut. *International Journal of CAD/CAM*, 3(1):1–12.

Elber, G. and Cohen, E. (1993). Tool path generation for freeform surface models. In *SMA '93: Proceedings of the Second Symposium on Solid Modeling and Applications*, pages 419–428.

Kim, S. I., Landers, R. G., and Ulsoy, A. G. (2003). Robust machining force control with process compensation. *Manufacturing Science and Engineering*, 125:423–430.

Lomolino, S., Tovo, R., and Santos, J. D. (2005). On the fatigue behaviour and design curves of friction stir butt-welded al alloys. *International Journal of Fatigue*, 27(3):305–316.

Martens, J. B. (1990). The hermite transform-theory. *IEEE Trans ASSP*, 38(9):1595–1606.

Mortenson, M. E. (2006). *Geometric Modeling*. Industrial Press Inc., 200 Madison Avenue, New York 10016-4078.

Motta, J. M. S. T., de Carvalhob, G. C., and McMaster, R. S. (2001). Robot calibration using a 3d vision-based measurement system with a single camera. *Robotics and Computer-Integrated Manufacturing*, 17(6):487–497.

Norrlöf, M. (2003). On path planning and optimization using splines. Technical report, Tekniska Högskolan.

Raibert, M. H. and Craig, J. J. (1981). Hybrid position/force control of manipulators. *ASME J. Dynamic Meas. Control*, (102):126–133.

Smith, C. B. (2000). Robotic friction stir welding using a standard industrial robot. *2nd Friction Stir Welding International Symposium*.

Soron, M. and Kalaykov, I. (2006). A robot prototype for friction stir welding. In *CD-ROM Proceedings of the IEEE International Conference on Robotics, Automation and Mechatronics, Bangkok, Thailand*.

Strombeck, A. V., Shilling, C., and Santos, J. F. D. (2000). Robotic friction stir welding - tool technology and applications. *2nd Friction Stir Welding International Symposium*.

Thomas, W. M., Nicholas, E. D., Needham, J. C., Church, M. G., Templesmith, P., and Dawes, C. J. (1991). International Patent Application no. PCT/GB92/02203 and GB Patent Application no. 9125978.9.

Venable, R., Colligan, K., and Knapp, A. (2004). Heat control via torque control in friction stir welding. *Nasa Tech Briefs*.

Wings, E. and Jütter, B. (2004). Generating tool paths on surfaces for a numerically controlled calotte cutting system. *Coputer-Aided Design*, 36:325–331.

INCLUSION OF ELLIPSOIDS

Romain Pepy and Eric Pierre
Institut d'Électronique Fondamentale
Université Paris XI
Orsay, France
pepy@ief.u-psud.fr

Keywords: Path planning, uncertainty, ellipsoid.

Abstract: We present, in this paper, a ready-to-use inclusion detection for ellipsoids. Ellipsoids are used to represent configuration uncertainty of a mobile robile. This kind of test is used in path planning to find the optimal path according to a safety criteria.

1 INTRODUCTION

Nowadays, path planning for mobile robots has taken a new dimension. Due to many failures when experimenting the following of geometrical paths, determined by the first generation of planners (Latombe, 1991), with real robots, searchers concluded that those too simple paths were no longer enough. Planning method must now guarantee that the paths proposed are safe, i.e. that the robot will be able to follow a path without any risk of failure, or at least in warranting a high success rate. To achieve this goal, some parameters must be considered : uncertainties of the model used (not-so-perfect mapping, inaccuracy of the sensors, slipping of the robot on the floor, etc.).

Collision detection is very important in mobile robotic and furthermore when trying to find a safe path in an uncertain-configuration space. Thus, searchers tend to integrate evolved collision detection in their planners. Thus, after having used circular disk to approximate the shape taken by the mobile robot, more and more searchers use elliptic disks as they offer a better accuracy.

Used in the context of *safe* path planning (Pierre and Lambert, 2006; Lozano-Pérez and Wesley, 1979; Gonzalez and Stentz, 2005; Pepy and Lambert, 2006), ellipsoids allow to approximate the shape of the set of positions where the mobile *could be* (ellipsoids thus take the mobile robot's geometry and the uncertainties on its position into account). The the Safe A* with Towers of Uncertainty (SATU*) planner (Pierre and Lambert, 2006) is one of those safe path planner that use ellipsoid to approximate the shape of the set of positions where the mobile could be. Ellipsoids are used in the SATU* to perform collision detection between the mobile robot and its environment. However, the authors of the SATU* have also proposed a new mean of organising the performing of the planner so that the ellipsoids can be used to detect very early beginning of useless paths. In order to achieve this goal, inclusion detection must be performed between two ellipsoids (that correspond to two different ways to come to the same position). In this paper, we are going to present an algebraic method using the resultant of Sylvester (Lang, 1984) to solve this problem. The SATU* algorithm (Alg. 1) has already been presented in (Pierre and Lambert, 2006) and three tests of inclusion of uncertainties (lines 7, 26 and 36) are used. However the authors did not explain how they implemented those tests nor give the algorithms used. As the model of uncertainties used in the SATU* corresponds to an ellipsoid in 3 dimensions, this test of inclusion of uncertainties can be seen as a test of inclusion of ellipsoids.

In the present paper, we are going to propose an algorithm of test of inclusion of ellipsoids. In a first part, the uncertain configuration space will be described. Then, Sylvester's resultant will be used to defined a ready for use inclusion detection test.

Algorithm 1 (SATU*).

```
CLOSE ← ∅
OPEN ← NodStart
while OPEN≠ ∅ do
  Nod ← Shortest_f*_Path(OPEN)
  CLOSE ← CLOSE + Nod
  OPEN ← OPEN - Nod
  if Base(Nod) = Base(NodGoal) and uncertainty(Nod)
  ⊆ uncertainty(NodGoal) then
    RETURN Success
  end if
  NEWNODES ← Successors(Nod)
  for all NewNod of NEWNODES do
    if NewNod ∉ OPEN,CLOSE then
      g(NewNod) ← g(Nod)+cost(Nod,NewNod)
      f*(NewNod) ←
      g(NewNod)+h(NewNod,NodGoal)
      build(NEWTOWER,base(NewNod))
      AddLevel(NEWTOWER,NewNod)
      OPEN ← OPEN+NewNod
      parent(NewNod) ← Nod
    else
      TOWER ← ExtractTower(base(NewNod))
      level ← -1
      repeat
        AddLevel ← false
        level ← level+1
        LevelNod ← ExtractNode(TOWER,level)
        if (g(NewNod) ≥ g(LevelNod) and uncer-
        tainty(LevelNod) ⊈ uncertainty(NewNod)) or
        g(NewNod) < g(LevelNod) then
          AddLevel ← true
        end if
      until level ≠ TopLevel(TOWER) and Ad-
      dLevel=true
      if AddLevel=true then
        level ← insert(NewNod,TOWER)
        OPEN ← OPEN + NewNod
        parent(NewNod) ← Nod
        UpperNods ←
        nodes(UpperLevels(TOWER,level))
        for all uppernod of UpperNods do
          if uncertainty(NewNod) ⊆ uncer-
          tainty(uppernod) then
            remove(TOWER,uppernod)
            OPEN ← OPEN - uppernod
          end if
        end for
      end if
    end if
  end for
end while
RETURN NoSolution
```

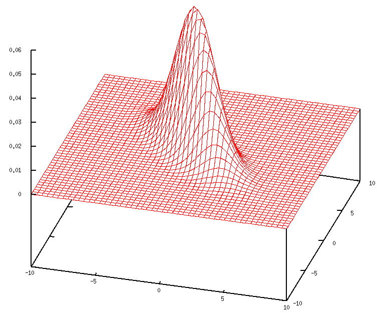


Figure 1: Example of $p_{(x,y)}.\,(x,y)$.

2 SHAPE OF UNCERTAINTIES

SATU* uses the Kalman filter (Kalman, 1960) to determine the uncertainties associated with the estimated positions of the mobile robot. (Smith and Cheeseman, 1986) gave a method to interpret the results of this filter. Indeed, thanks to this filter, we know at each time the covariance matrix $\mathbf{C}$ representing the uncertainties on the position $\mathbf{x}$ of the mobile robot. As we are only interested in the position of the mobile robot (in x and y), our random vector $\mathbf{x}$ varies in $\mathbb{R}^2$. As the Kalman filter uses only gaussian random vectors, the density of probability $p_{\mathbf{x}}(\mathbf{x})$ of the vector $\mathbf{x}$ is

$$p_{\mathbf{x}}(\mathbf{x}) = \frac{1}{2\pi\sqrt{\det \mathbf{C}}} e^{-\frac{1}{2}(\mathbf{x}-\hat{\mathbf{x}})^T \mathbf{C}^{-1}(\mathbf{x}-\hat{\mathbf{x}})},$$

where $\hat{\mathbf{x}}$ represents the estimated state vector.

We know the covariance matrix $\mathbf{C}$, which is given by

$$\mathbf{C} = \begin{bmatrix} \sigma_x^2 & \rho\sigma_y\sigma_x \\ \rho\sigma_y\sigma_x & \sigma_y^2 \end{bmatrix}, \tag{1}$$

where σ_x^2 represents the variance of x, σ_y^2 the variance of y and ρ is the correlation coefficient of x and y.

In translating the coordinate frame to the known point $\hat{\mathbf{x}}$, the probability density function is

$$p_{x,y}(x,y) = \frac{1}{2\pi\sqrt{\det \mathbf{C}}} e^{-\frac{1}{2(1-\rho^2)}\left[\frac{x^2}{\sigma_x^2}+\frac{2\rho xy}{\sigma_x\sigma_y}+\frac{y^2}{\sigma_y^2}\right]}. \tag{2}$$

An example of drawing of this probability density function is given figure 1.

Our goal is to determine the set of positions where the mobile robot can be for a given confidence threshold p. This corresponds in finding an areo of isodensity contours such that

$$(\mathbf{x}-\hat{\mathbf{x}})^T \mathbf{C}^{-1}(\mathbf{x}-\hat{\mathbf{x}}) = k^2,$$

where k is a constant. The relation between k and p is given by $k = \sqrt{-2\ln(1-p)}$.

Thus, we can determine the equation of the ellipsoid (centered on the known point $\hat{\mathbf{x}}$) where the robot mobile may be with the given probability p:

$$Ax^2 + Bxy + Cy^2 = k^2, \tag{3}$$

with

$$\begin{cases} A &= \frac{1}{(1-\rho^2)\sigma_x^2}, \\ B &= \frac{2\rho}{(1-\rho^2)\sigma_x\sigma_y}, \\ C &= \frac{1}{(1-\rho^2)\sigma_y^2}. \end{cases} \tag{4}$$

This equation describes an ellipsoid. A classical result allows us to write this equation in function of the parameters a, b and φ of the ellipsoid. Considering that a is the half major axis of the ellipsoid, b its half minor axis and φ its orientation, we have

$$\begin{cases} \varphi &= \frac{1}{2}\arctan\left(\frac{B}{A-C}\right), \\ a &= \sqrt{\frac{2k^2}{A+C-\sqrt{A^2+C^2-2AC+B^2}}}, \\ b &= \sqrt{\frac{2k^2}{A+C+\sqrt{A^2+C^2-2AC+B^2}}}. \end{cases} \tag{5}$$

Using (4) and (5), we can determine the parameters of the ellipsoid in function of the covariance matrix given by the EKF.

We are going to work with this ellipsoid in order to find a test of inclusion of ellipsoids. Then, a simple test on σ_θ^2 will allow us to complete the test of inclusion of 3-D ellipsoids.

3 DEFINITIONS AND NOTATIONS

The equation of an ellipsoid E_i with a half major axis a_i and a half minor axis $b_i < a_i$, centered on the origin of the frame of reference and with no orientation is given by

$$\frac{x'^2}{a_i^2} + \frac{y'^2}{b_i^2} = 1, \tag{6}$$

considering x' and y' the coordinates of the ellipsoid.

To determine the equation of the centered ellipsoid of orientation φ_i, the use of a simple rotation matrix of angle $-\varphi_i$ is enough. The new coordinates of the ellipsoid are named x and y. Equation (6) then becomes

$$A_i x^2 + B_i y^2 + C_i xy - 1 = 0 \tag{7}$$

where

$$\begin{cases} A_i &= \frac{(b_i\cos\varphi_i)^2 + (a_i\sin\varphi_i)^2}{(a_ib_i)^2} \\ B_i &= \frac{(b_i\sin\varphi_i)^2 + (a_i\cos\varphi_i)^2}{(a_ib_i)^2} \\ C_i &= \frac{\left(b_i^2 - a_i^2\right)\sin(2\varphi_i)}{(a_ib_i)^2}. \end{cases} \tag{8}$$

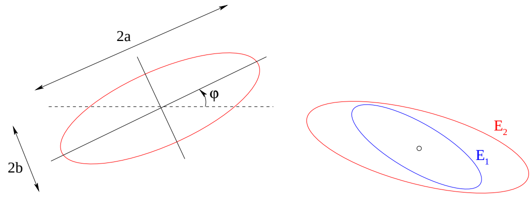


Figure 2: Ellipsoid's parameters.

It represents the equation of a centered ellipsoid of half major axis a_i, half minor axis b_i, and of orientation φ_i (figure 2). We will call this ellipsoid $E_i\left(a_i, b_i, \varphi_i\right)$.

In this paper, we will use two ellipsoids $E_1\left(a_1, b_1, \varphi_1\right)$ and $E_2\left(a_2, b_2, \varphi_2\right)$.

4 INCLUSION TEST

To ensure an ellipsoid E_1 is included in an ellipsoid E_2(see figure 2), we just need to perform two tests. We must check there is no intersection between E_1 and E_2 and then that E_1 lies within E_2.

4.1 Sylvester's Resultant

Checking if two ellipsoids intersect is the same as finding the tuples (x, y), which are solutions of both E_1 and E_2. The use of the resultant allows to find those tuples.

Definition Let $P = u_0X^p + \cdots + u_p$ and $Q = v_0X^q + \cdots + v_q$ be two polynoms of degrees $p > 0$ and $q > 0$ respectively. The *resultant* of two polynomials P and Q is the determinant of the *Sylvester's matrix* of P and Q, which is the square matrix of size $p+q$:

$$\begin{pmatrix} u_0 & u_1 & \cdots & \cdots & \cdots & u_p & 0 & \cdots & 0 \\ 0 & u_0 & u_1 & \cdots & \cdots & \cdots & u_p & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & u_0 & u_1 & \cdots & \cdots & \cdots & u_p \\ v_0 & v_1 & \cdots & \cdots & v_q & 0 & \cdots & \cdots & 0 \\ 0 & v_0 & v_1 & \cdots & \cdots & v_q & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & v_0 & v_1 & \cdots & \cdots & v_q \end{pmatrix}.$$

Theorem Let $P = u_0X^p + \cdots + u_p$ and $Q = v_0X^q + \cdots + v_q$ be two polynomials with their coefficient in a field $\mathbb{K}$, so that $u_0 \neq 0$ and $v_0 \neq 0$. Let $\overline{\mathbb{K}}$ be an algebraically closed field[1] containing $\mathbb{K}$. The two following properties are equivalent:

1. The polynomials P and Q have a common root in $\overline{\mathbb{K}}$.
2. P and Q's resultant is null.

The demonstration is given in (Lang, 1984).

4.2 Intersection Test

Theorem 4.1 indicates that two polynomials have at least one solution in the algebraic closure of the field in which the coefficients are defined if the resultant of those polynomials is nil. As the equations of the ellipsoids are polynomials (bilinear), we can use the resultant to determine the intersection point of the ellipsoids. The coefficients of the polynomials (ellipsoids) are defined on $\mathbb{R}$, the use of the Sylvester's resultant gives the common roots of the polynomial in an algebraically closed field containing $\mathbb{R}$, for example $\mathbb{C}$. We will then need to check if the roots are in $\mathbb{R}$, as we just need *real* intersections. To define the Sylvester's matrix, we need to rewrite the equation of the ellispoid in function of only one parameter. Let's choose (randomly) x.

$$(7) \Leftrightarrow Ax^2 + (Cy)x + By^2 - 1 = 0,$$

which gives, for the equations of E_1 and E_2:

$$\begin{cases} A_1x^2 + (C_1y)x + B_1y^2 - 1 &= 0 \\ A_2x^2 + (C_2y)x + B_2y^2 - 1 &= 0. \end{cases} \tag{9}$$

Sylvester's matrix of the polynomials is then:

$$S = \begin{pmatrix} A_1 & C_1y & B_1y^2 - 1 & 0 \\ 0 & A_1 & C_1y & B_1y^2 - 1 \\ A_2 & C_2y & B_2y^2 - 1 & 0 \\ 0 & A_2 & C_2y & B_2y^2 - 1 \end{pmatrix}.$$

The determinant of S is

$$|S| = py^4 + qy^2 + r, \tag{10}$$

with

$$\begin{aligned} p &= -2A_1A_2B_1B_2 + (A_1B_2 - A_2B_1)C_1C_2 \\ &\quad + A_1B_1C_2^2 + A_2B_2C_1^2 + A_1^2B_2^2 + A_2^2B_1^2 \\ q &= 2A_1A_2(B_1 + B_2) + (A_2 - A_1)C_1C_2 \\ &\quad - 2A_1^2B_2 - 2A_2^2B_1 - A_1C_2^2 - A_2C_1^2 \\ r &= (A_2 - A_1)^2. \end{aligned}$$

The change of variables $Y = y^2$ allows us to rewrite the determinant:

$$|S| = pY^2 + qY + r.$$

Algorithm 2 ComputeEquation(a, b, φ).

1: $A \leftarrow ((b\cos\varphi)^2 + (a\sin\varphi)^2)/(ab)^2$
2: $B \leftarrow ((b\sin\varphi)^2 + (a\cos\varphi)^2)/(ab)^2$
3: $C \leftarrow ((b^2 - a^2)\sin(2\varphi))/(ab)^2$
4: RETURN (A, B, C)

Algorithm 3 Determinant$(A_1, B_1, C_1, A_2, B_2, C_2)$.

1: $p \leftarrow -2A_1A_2B_1B_2 + (A_1B_2 - A_2B_1)C_1C_2 + A_1B_1C_2^2 + A_2B_2C_1^2 + A_1^2B_2^2 + A_2^2B_1^2$
2: $q \leftarrow 2A_1A_2(B_1 + B_2) + (A_2 - A_1)C_1C_2 - 2A_1^2B_2 - 2A_2^2B_1 - A_1C_2^2 - A_2C_1^2$
3: $r \leftarrow (A_2 - A_1)^2$
4: RETURN (p, q, r)

We are looking for the roots of $|S|$ in $\mathbb{R}$. If the discriminant of the resultant (which is $q^2 - 4rp$) is strictly negative, there is no roots in $\mathbb{R}$. If r is nil, we can directly conclude that there is at least one intersection point because there exists at least one real root ($y = 0$).

If the discriminant of $|S|$ is positive or nil, there is at least one root in $\mathbb{R}$. We conclude that the ellipsoids have at least one intersection point in $\mathbb{C}$ (y may be real, x could be complex). We then need to calculate the values of y that gives an intersection point, then check that the corresponding values of x are ni $\mathbb{R}$ too. Roots of 10 are given by:

$$\begin{cases} y_{1,2} &= \pm\frac{-q-\sqrt{\Delta}}{2p} \\ y_{3,4} &= \pm\frac{-q+\sqrt{\Delta}}{2p} \end{cases} \tag{11}$$

Using 11, (9) becomes:

$$(A_1 - A_2)x^2 + y(C_1 - C_2)x + y^2(B_1 - B_2) = 0, \tag{12}$$

which discriminant is

$$\Delta = y^2(C_1 - C_2)^2 - 4y^2(A_1 - A_2)(B_1 - B_2). \tag{13}$$

If $\Delta \geq 0$, then equation (12) has real roots. We conclude that the global system as a tuple (x, y) of real solutions. Thus, ellipsoids have at least one intersection point. On the contrary, if equation (12) has no real root in x, then the ellipsoids do not intersect.

4.3 Test of Length

If the ellipsoids do not intersect, then it means that one of them is fully included in the other (as they have the same center). To test if it is the first ellipsoid that is included in the second, we just need to check that the half major axis of the first ellipsoid, a_1, is smaller

[1] A field $\overline{\mathbb{K}}$ is algebraically closed if every polynomial of $\overline{\mathbb{K}}[X]$ is split on $\overline{\mathbb{K}}$, i.e. product of polynomials of degree 1.

Algorithm 4 ComputeRoots(p,q,Δ).

$y1 \leftarrow (-q-\sqrt{\Delta})/(2p)$
$y3 \leftarrow (-q+\sqrt{\Delta})/(2p)$
RETURN $(y_1,-y_1,y_3,-y_3)$

Algorithm 5 InclusionTest$(a_1,b_1,\varphi_1,a_2,b_2,\varphi_2)$.

$(A_1,B_1,C_1) \leftarrow$ ComputeEquation(a_1,b_1,φ_1)
$(A_2,B_2,C_2) \leftarrow$ ComputeEquation(a_2,b_2,φ_2)
$(p,q,r) \leftarrow$ Determinant$(A_1,B_1,C_1,A_2,B_2,C_2)$
$\Delta \leftarrow q^2-4rp$
if $\Delta \geq 0$ **then**
  $(y_1,y_2,y_3,y_4) \leftarrow$ ComputeRoots(p,q,Δ)
  for all $y \in \{y_1,y_2,y_3,y_4\}$ **do**
    $\Delta_y \leftarrow y^2\,(C_1-C_2)^2-4y^2\,(A_1-A_2)\,(B_1-B_2)$
    if $\Delta_y \geq 0$ **then**
      RETURN NO INCLUSION
    end if
  end for
end if
if $(a_1 \geq a_2)$ or $(b_1 \geq b_2)$ **then**
  RETURN NO INCLUSION
end if
RETURN INCLUSION

than the half major axis of the second ellipsoid, a_2. We can, following the same scheme, check that the half minor axis of the first ellipsoid b_1 is smaller than the half minor axis of the second ellipsoid, b_2.

4.4 Algorithm

The complete algorithm of inclusion of two ellipsoids which have the same center is given alg. 5. Sub-functions ComputeEquation (Alg. 2), Determinant (Alg. 3) and ComputeRoots (Alg. 4) refer respectively to the equations (8), (10) and (11).

5 CONCLUSION

In this article, we presented a method of test of inclusion of ellipsoids when those ellipsoids have the same center. This test of inclusion is necessary in the use of some path planners such as the SATU* planner (Pierre and Lambert, 2006). The proposed method is very easily implementable on computer and have a constant and low complexity.

REFERENCES

Gonzalez, J. P. and Stentz, A. T. (2005). Planning with uncertainty in position: An optimal and efficient planner. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS '05)*.

Kalman, R. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45.

Lang, S. (1984). *Algebra*. Addison-Wesley.

Latombe, J. C. (1991). *Robot Motion Planning*. Kluwer Academic Publisher.

Lozano-Pérez, T. and Wesley, M. A. (1979). An algorithm for planning collision-free paths among polyhedral obstacles. *Commun. ACM*, 22(10):560–570.

Pepy, R. and Lambert, A. (2006). Safe path planning in an uncertain-configuration space using rrt. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5376–5381, Beijing, China.

Pierre, E. and Lambert, A. (2006). Path planning for car-like robots using towers of uncertainties. In *Proceedings IEEE International Conference on Information and Communication Technologies: from Theory to Applications*, pages 754–759, Damascus, Syria.

Smith, R. and Cheeseman, P. (1986). On the representation and estimation of spacial uncertainty. *International Journal of Robotics Research, pp 56-68, 5(4).*

MINIMUM COST PATH SEARCH IN AN UNCERTAIN-CONFIGURATION SPACE

Eric Pierre and Romain Pepy
Institut d'Électronique Fondamentale
Université Paris XI
Orsay, France
pepy@ief.u-psud.fr

Keywords: Path planning, uncertainty, A* algorithm.

Abstract: The object of this paper is to propose a minimum cost path search algorithm in an uncertain-configuration space and to give its proofs of optimality and completeness. In order to achieve this goal, we focus on one of the simpler and efficient path search algorithm in the configuration space : the A* algorithm. We then add uncertainties and deal with them as if they were simple dof (degree of freedom). Next, we introduce towers of uncertainties in order to improve the efficiency of the search. Finally we prove the optimality and completeness of the resulting algorithm.

1 INTRODUCTION

Today's researches on mobile robots focalize on a specific aspect: the autonomous navigation of a robot. Navigating a robot generaly involves the use of a path planner, which determines a path in a graph built from the environment. However, where path planner taking into account geometric models of the environment work well in simulation, the implementation of such algorithms on real robots leads in most case to failure to follow the path planned (Latombe, 1991). Thus, searchers have decided to introduce the concept of uncertainty, in order to generate a path that would work even with uncertain data. This field of research has leaded to various works (Fraichard and Mermond, 1998; Lazanas and Latombe, 1995; Pepy and Lambert, 2006).
The present paper is the following of the work in (Lambert and Gruyer, 2003), which proposes a path planner based on the A* algorithm ((Hart et al., 1968; Nilsson, 1988; Russel and Norvig, 1995)) that takes uncertainties into account. Using a simulator of exteroceptive and proprioceptive sensors and an Extended Kalman Filter to process the sensors information, the algorithm finds a *safe* path for a mobile robot in a known-but-imperfect indoor environment despite the uncertainties associated with the lack of precision of the sensors. Unfortunately, the authors do not prove the optimality and completeness of the algorithm they propose.
The goal of the present paper is firstly to generalize the algorithm given in (Lambert and Gruyer, 2003) to any problem based on planning a path in a graph with taking into account uncertainties on the graph's parameters. Secondly, it is to prove the optimality and completeness of this generalized algorithm.
In section 2, we build the mathematical structure needed to explain the A* algorithm and to prepare the proofs. Using the mathematical bases proposed in the previous section, section 3 presents the A* while taking the uncertainties into account, which we will call A* in an Uncertain Configuration Space (AUCS*). Finally, section 4 proposes a second and optimized version of this algorithm (called Safe A* with Towers of Uncertainties, or SATU*) using a new concept: the towers of uncertainties (Lambert and Gruyer, 2003).

2 THE A* ALGORITHM

2.1 Overview

The famous A* algorithm (algorithm 1) is one of the simpler and efficient path planner in a graph. This algorithm has first been proposed in (Hart et al.,

1968), and is based on the Dijkstra algorithm (Dijsktra, 1959) on which a heuristic function has been added.
The algorithm searches the graph in order to find a path that goes from a start node to a goal node. When we deal with a path planner, it is important to know its behaviour, especially about two particular properties :

- Optimality: When the algorithm chooses a path to go from the start node to the goal node, this path must be the optimal one (given some criteria of optimality).
- Completeness: Given that the goal is reachable, the algorithm must reach it in a finite time.

If the path planner we deal with verifies those properties, we know for sure that a path will eventually be found (given that the goal node is reachable) and that this path will be the best one, according to a given criteria.

2.2 Spaces Used

We want to find a path in a **n**-dimension configuration space $\mathcal{X}$. The **n** dimensions represent the degree of freedom (dof) of the system. We have to deal with two subspaces of $\mathcal{X}$, one representing the *free*-configuration space (called $\mathcal{X}_{free}$), where we can move, and the other representing the *collision*-configuration space (called $\mathcal{X}_{collision}$), where the movements are impossible.

2.3 Structure of the Graph

We discretize $\mathcal{X}_{free}$ into an infinite number of nodes. Let $\mathcal{N}$ be the set of those nodes. They are characterised by **n** parameters. We can build a directed graph (the nodes are linked together by one-way edges) in $\mathcal{X}_{free}$, which follows those properties :

Property 1 *There is a minimum positive edge cost between any two nodes (i.e., there is a given positive* ε *so that the cost between any two nodes is greater than* ε*)*

Property 2 *The number of edges starting from a node is finite.*

Property 3 *The edge cost between any two nodes in a path is finite.*

When a graph possesses the first two properties, it is called a *locally finite graph* (in a given and finite area of the space, there is a finite number of paths between any two nodes) (Nilsson, 1988).

Algorithm 1 A*

```
CLOSE ← ∅
OPEN ← NodeStart
while OPEN≠ ∅ do
  Node ← Shortest_f*_Path(OPEN)
  CLOSE ← CLOSE + Node
  OPEN ← OPEN - Node
  if Node = NodeGoal then
    return (Success,Node)
  end if
  NEWNODES ← Successors(Node) ∉ CLOSE
  for all NewNode of NEWNODES do
    if NewNode ∉ OPEN or g(NewNode) > g(Node)+cost(Node,NewNode) then
      g(NewNode) ← g(Node)+cost(Node,NewNode)
      f*(NewNode) ← g(NewNode)+h(NewNode,NodeGoal)
      parent(NewNode) ← Node
      if NewNode ∉ OPEN then
        OPEN ← OPEN+NewNode
      end if
    end if
  end for
end while
return NoSolution
```

2.4 Paths of the Graph

- let $\mathcal{P}$ be the set of all the paths beginning at the start node and exploring the configuration space. An important characteristic is that no path in $\mathcal{P}$ can loop (a path cannot go through a node twice).
- let $\mathcal{P}_{goal}$ be the set of all the paths beginning at the start node and ending on the goal node.
- let $\mathcal{P}_{\overline{goal}}$ be the set of all the paths beginning at the start node and that do not reach the goal node. We have

$$\mathcal{P}_{\overline{goal}} = \mathcal{P} \setminus \mathcal{P}_{goal}. \tag{1}$$

An important thing to keep in mind is that the A* works incrementally with $\mathcal{P}$. It tests paths of $\mathcal{P}$ node by node and does not know if those paths belong to $\mathcal{P}_{goal}$ or $\mathcal{P}_{\overline{goal}}$ until it reaches their final node.

2.5 Parameters of the Algorithm

2.5.1 Criteria of Optimality

Each path has an associated cost, which is the cumulated costs of every edge in the path. Let $f(c_i)$ be this cost, for all $c_i \in \mathcal{P}$.

Let us introduce, $g(c_i, n)$, which represents the length of the part of the path c_i going from the start node to the node n, $n \in c_i$. If c_i is finite, we can observe that

$$f(c_i) = g(c_i, \text{finalnode}(c_i)). \tag{2}$$

We want to find a shortest path, i.e. a path $c \in \mathcal{P}_{goal}$ that verify

$$f(c) \leq f(c_i) \quad \forall c_i \in \mathcal{P}_{goal}. \tag{3}$$

Finding such a path is our *criteria of optimality*.

2.5.2 Rule to Expand the Graph

As it is well known, the A* works with a path-independant heuristic function $h(n_j, n_k)$ that estimates the unknown length between two nodes n_j and n_k. We can then work with an estimated total length $f^*(c_i, n)$ rather than the real (unknown) $f(c_i)$, such that

$$f^*(c_i, n) = g(c_i, n) + h(n, \text{goalnode}). \tag{4}$$

2.6 Proofs of the Algorithm

Both the proof of optimality and of completeness have been given in (Nilsson, 1988). We won't detail them further here.

3 THE A* IN AN UNCERTAIN CONFIGURATION-SPACE

3.1 The Concept of Uncertainties

The uncertainties represent the fact that in the real world, one cannot precisely know the real model of the systems. Thus, we can determine the errors induced by all the models used and we can calculate the uncertainties associated to each node (in $\mathcal{N}$) of each path (in $\mathcal{P}$). Let us consider that the uncertainties are fully described in a **m**-dimension space. Consequently, a full uncertain configuration is now given by $\mathbf{n}+\mathbf{m}$ parameters: **n** for the coordinates in the space $\mathcal{X}$ and **m** for the coordinates in the uncertainties-space. Such an uncertain configuration will be called an *extended node*.
In this section, we are going to present a way to run a simple A* on a new graph based on extended nodes, in considering that the uncertainties can be managed as additional degree of freedom.

3.2 Spaces Used

Let $\mathcal{X}^e$ be the uncertain configuration space, which is $\{\mathbf{n}+\mathbf{m}\}$-dimension space and fully include $\mathcal{X}$ (we can then consider that we deal with $\{\mathbf{n}+\mathbf{m}\}$ dof, even if some of them are uncertainties). Knowing the uncertainties associated to a given node, we can build a new $\mathbf{n}+\mathbf{m}$ dimensions free-configuration space $\mathcal{X}^e_{free}$ taking into account those incertainties.
Let $\mathcal{X}^e_{collision}$ be characterised by

$$\mathcal{X}^e_{collision} = \mathcal{X}^e \backslash \mathcal{X}^e_{free}. \tag{5}$$

Our goal is now to find a safe path, which is a path that never go in $\mathcal{X}^e_{collision}$ (considering the uncertainties of every extended node).

3.3 Structure of the Graph

In the previous case (see section 2.3 above) we already introduced a graph based on $\mathcal{N}$. However, we do not want to work with the set of nodes $\mathcal{N}$, as it does not take the uncertainties into account. Let us introduce a new set of nodes, called $\mathcal{U}$. The nodes in this set will then be *extended* nodes. The properties applying on $\mathcal{N}$ apply also on $\mathcal{U}$ (the three properties presented in section 2.3 applie on this new graph). Thus, our new graph is locally finite.
Another property may be pointed out:

Property 4 *An infinite number of extended node can be based on the same node.*

3.4 Paths of the Graph

We can now introduce the paths generated through the *extended nodes* of $\mathcal{U}$.

- let $\mathcal{G}$ be the set of all the paths beginning at the start *extended node* and exploring the $\{\mathbf{n}+\mathbf{m}\}$-dimension space configuration $\mathcal{X}^e_{free}$. The criteria of optimality presented in section 2.5 applies fully here.
- let $\mathcal{G}_{goal}$ be the set of all the paths beginning at the start *extended node* and reaching the goal *extended node*. A direct consequence is that $\mathcal{G}_{goal} \subset \mathcal{G}$.
- let $\mathcal{G}_{\overline{goal}}$ be the set of all the paths beginning at the start *extended node* and that do not reach the goal *extended node*. We have

$$\mathcal{G}_{\overline{goal}} = \mathcal{G} \setminus \mathcal{G}_{goal}. \tag{6}$$

- g, f, f^* and h represent respectively the length from the start node to a fixed node in a path, the total length of a path, the estimated length of a path and the heuristic of a path in the $\{\mathbf{n}+\mathbf{m}\}$-dimension space.

What we finally search is the *shortest path*, which is the shortest path c in $\mathcal{G}$. This path must verify

$$f(c) \leq f(c_i) \quad \forall c_i \in \mathcal{G}. \tag{7}$$

3.5 The Algorithm

It is now possible to implement the A*, which we will call A* in an Uncertain Configuration-Space (AUCS*), directly on the newly described $\{\mathbf{n}+\mathbf{m}\}$-dimension graph. The AUCS* is exactly the same algorithm than the A*, the only difference is the graph on which it performs. The goal node is then defined by $\mathbf{n}+\mathbf{m}$ parameters, the first $\mathbf{n}$ corresponding to the location we want to reach and the $\mathbf{m}$ other parameters being the uncertainties we want to have at this location.
As the algorithm is exactly the same (despite it performs on a different graph), we do not give it again. See algorithm 1.

3.6 Proofs of the Algorithm

3.6.1 Proofs of Optimality

Despite the algorithm performs on an improved graph, its principal characteritics are unchanged. The heuristic and calculus of length have the same properties (g and h could eventually be identical to those used in the previous section), and regarding this model the AUCS* still find the shortest path. □

3.6.2 Proofs of Completeness

There again, the only cases the AUCS* could not be complete is when there is an infinite number of paths such as their lengths are lower than the found path's or when a path with a length lower than the found path is infinite.
The proof is equivalent to the one determined by (Nilsson, 1988): the properties of the graph built are the same than in the previous case (it is locally finite), this is enough to lead to the same conclusion ; the AUCS* algorithm is complete. □

3.7 Working in a Finite Graph

In the previous sections, we considered the graphs were infinite and built in advance, before the algorithm starts. However, building an infinite graph is impossible for a (necessarily limited) computer. Thus, some choices must be done if we want to be able to run the algorithm on a computer. For example, we could define a sub-space of $\mathcal{X}$ where to run the algorithm. Of course, this implies no path not being entirely contained in $\mathcal{X}$ can be found. This is generally not a problem if the sub-space is cleverly defined (for example, if we want to find a path leading from a room to another in a building, we could restrain our space to the building itself). This is enough to ensure the new graph will be finite:

Property 5 *A graph generated in a finite sub-space of $\mathcal{X}$ and following properties 1 and 2 (section 2.3) is finite.*

Proof: Property 1 ensures that any two nodes must have a given minimum positive edge cost. Consequently, a finite sub-space can only be filled with a finite number of nodes. Property 2 ensures that only a finite numbers of edges can go from a node: the graph is finite. □
Thus, we can work on a pre-built sub-graph of $\mathcal{X}$. However in the case of $\mathcal{X}^e$, the problem is slightly different. Property 4 implies there could be an infinite number of extended nodes on each node. Of course, it is not possible to build such a graph with a computer. Thus, we are going to work on a dynamically built graph: from $\mathcal{P}$, the extended nodes are dynamically added in the new graph $\mathcal{G}$ when they are reached.
Proofs of optimality and completeness ensure that the graph stays finite, as the dynamically building of the extended nodes ends when the algorithm reaches the goal node, which has been proven happens in a finite time.
The only change in algorithm 1 is that line 10, *Successors* function dynamically creates the nodes if they do not already exist. This implies that, given a fully-described extended node and the base of the node where to go, *Successors* must be able to calculate the $\mathbf{m}$ last parameters of the node to reach.
We now have an implementable algorithm in an uncertain configuration space, which does not need infinite memory.

4 THE A* WITH TOWERS OF UNCERTAINTIES

4.1 The Towers of Uncertainties

In section 3.7, we showed that we could work on a finite dynamically built graph. However, finding a way to reduce the need of memory needed would be very helpful. (Lambert and Gruyer, 2003) has determined a method (described below) that reduces the field of search (reducing the memory needed) and reorganizing the way to record the data. This method involves the use of *towers of uncertainties*, and only works on dynamically built graphs (as presented section 3).
However, the graphs must be more constrained than those seen in the sections above.
Constraint 1: This constraint concerns the evolution

of the uncertainties between two nodes: this evolution follows a given function and this function is the same for every uncertainty. If the uncertainties of a given extended node include the uncertainties of another extended node (those extended nodes must be based on the same node), then the pattern ensures that this property will be kept in the following nodes of the paths (when the paths follow the exact same nodes).
Constraint 2: If the function (collision-detection) giving the configuration-space where the path belongs ($\mathcal{X}_{free}$ or $\mathcal{X}_{collision}$) declares the path in $\mathcal{X}_{collision}$ for a given uncertainty, then it will declare a new path following the same (non-extended) nodes with wider uncertainties in $\mathcal{X}_{collision}$ too.

When the AUCS* searches the graph, it generates various beginning of paths. Given that numerous extended nodes can be at the same position (see property 4 above), it is perfectly possible to have two or more different paths reaching the same position (same node, but different extended nodes).
Without knowing what comes in the future, could not we already discard some of those paths?
The criteria of optimality we want to minimize is the length of the path. What we search is the smallest path. However, some paths expanded at some step of the algorithm may not reach the goal (for example, every path built from this path does not belong to $\mathcal{X}^{e}_{free}$). Thus, the current smallest path could be in $\mathcal{X}^{e}_{collision}$ without the algorithm knowing it (the part being in collision with the environment having not been reached by the algorithm yet) and a longer path could be in $\mathcal{X}^{e}_{free}$. The immediate consequence is that we should keep both of them during the search.
What about if the longer path has also wider uncertainties? Then it is clear that if the smallest path does belong to $\mathcal{G}_{\overline{goal}}$, the longer path belongs to $\mathcal{G}_{\overline{goal}}$ too (this is ensured by constraint 2 presented above, in section 4.1, as from this node on, the exact same paths will be tried by both of the paths). In this case, we could discard the longer path. On the other hand, if the smallest path is in $\mathcal{G}_{free}$, then we do not need the longer path anymore, even if it is in $\mathcal{G}_{free}$ too, as its length is by definition longer. We could also discard it in this case.
Consequently, we can *always* discard the longer path with the wide uncertainties.
This property considerably reduces the complexity of the algorithm, as it detects earlier useless paths. In order to make the detection of the useless paths the quicker possible, (Lambert and Gruyer, 2003) proposes the use of *towers of uncertainties* in an algorithm we call SATU* (Safe A* with Towers of Uncertainties).
Each extended node is divided in two entities:

- *A base* : Given by the node of the extended node (the **n** first parameters).
- *A level* : The uncertainties associated to the node and function of the path leading up there (the **m** following parameters).

So, we can now dynamically (while the algorithm performs) build a tower of uncertainties, with a common base, and as many levels as the number of extended nodes (with the same common base) opened at this point. The levels are placed following some rules:

- The levels are placed in an increasing order of their length.
- Given a level, if another level under it has uncertainties that completely fits in its own, then the given level is removed (its length is bigger and its uncertainties are wider than the level's under it).

A direct consequence of the description above is that the memory needed for the SATU* is lower than for the AUCS*, even if we consider that the AUCS* build the graph dynamically too: for a given number n of extended nodes which have the same base, the needed memory will be $(\mathbf{n}+\mathbf{m})n$ in the AUCS* and only $\mathbf{n}+n.\mathbf{m}$ in the the SATU*.

4.2 Description of the Algorithm

SATU* is given in algorithm 2. The first part of the algorithm is the same as algorithm 1: two lists are created and initialized, a loop allows to expand extended node after extended node (selecting the extended node with the smallest f), a test verify if the goal extended node has been reached, otherwise the successors of the current extended node are selected and opened.
For each of those successors, a first test (line 12) checks if the base of the successor is already in CLOSE or OPEN. If it is not, then when can create a new tower on this base, add a first level with the uncertainties associated with the successor and add the base of the node in OPEN (without forgetting to calculate f and to store the parent of the node in order to be able to find the path when the algorithm is finished).
If the base of the successor belongs either to CLOSE or OPEN, we need to check if the successor may enter the tower or not.
In order to do that, the algorithm must compare each one of the levels of the tower with the successor's uncertainties and g. Line 20 and 21, the algorithm selects the tower and initialized an index which will represent the value of the current level being checked. Line 22 begins the loop that will compare the current level extracted and the successor.
Line 25, the current level to compare is extracted from

Algorithm 2 (SATU*)

```
1:  CLOSE ← ∅
2:  OPEN ← NodeStart
3:  while OPEN≠ ∅ do
4:    Node ← Shortest_f*_Path(OPEN)
5:    CLOSE ← CLOSE + Node
6:    OPEN ← OPEN - Node
7:    if Base(Node) = Base(NodeGoal) and uncertainty(Node) ⊆ uncertainty(NodeGoal) then
8:      return Success
9:    end if
10:   NEWNODES ← Successors(Node)
11:   for all NewNode of NEWNODES do
12:     if NewNode ∉ OPEN,CLOSE then
13:       g(NewNode) ← g(Node)+cost(Node,NewNode)
14:       f*(NewNode) ← g(NewNode)+h(NewNode,NodeGoal)
15:       build(NEWTOWER,base(NewNode))
16:       AddLevel(NEWTOWER,NewNode)
17:       OPEN ← OPEN+NewNode
18:       parent(NewNode) ← Node
19:     else
20:       TOWER ← ExtractTower(base(NewNode))
21:       level ← -1
22:       do
23:         AddLevel ← false
24:         level ← level+1
25:         LevelNode ← ExtractNode(TOWER,level)
26:         if (g(NewNode) ≥ g(LevelNode) and uncertainty(LevelNode) ⊈ uncertainty(NewNode)) or g(NewNode) < g(LevelNode) then
27:           AddLevel ← true
28:         end if
29:       while level ≠ TopLevel(TOWER) and AddLevel=true
30:       if AddLevel=true then
31:         level ← insert(NewNode,TOWER)
32:         OPEN ← OPEN + NewNode
33:         parent(NewNode) ← Node
34:         UpperNodes ← nodes(UpperLevels(TOWER,level))
35:         for all uppernode of UpperNodes do
36:           if uncertainty(NewNode) ⊆ uncertainty(uppernode) then
37:             remove(TOWER,uppernode)
38:             OPEN ← OPEN - uppernode
39:           end if
40:         end for
41:       end if
42:     end if
43:   end for
44: end while
45: return NoSolution
```

the tower.
Line 26, a test is performed. We may insert the successor in the tower if either its g is greater or equal than the level's and its uncertainties are not included in the level's or if its g is lower than the level's. Thus, if those conditions are met, we keep the successor and try to compare it to the next level. If, just once, those conditions are not met, then we can discard the successor.
Line 30, if we may insert the successor in the tower, we do it.
Line 31, we insert the successor in the tower in the increasing order of level's g.
we then add the successor in OPEN, for it to be expanded later.
Line 34, we select the nodes with a greater g than the successor's, and check if they can be kept (if their uncertainties are included in the successor's, then we can discard them). This is done in removing them from the tower and from OPEN lines 37 and 38.

4.3 Proofs of the Algorithm

4.3.1 Proof of Optimality

As shown in section 4, the SATU* works as a AUCS*. The addition of the tower of uncertainties only allows to discard very early a path that is either in $\mathcal{G}_{\overline{goal}}$ or not the smallest path in $\mathcal{G}_{free}$. Thus, regarding the discovery of the smallest path, it has the exact same behaviour than a AUCS*, which proves that it respects the criteria of optimality. The algorithm therefore ensures that its found path c verifies

$$g(c) \leq g(c_i) \quad \forall c_i \in \mathcal{G}_{free}. \tag{8}$$

□

4.3.2 Proof of Completeness

Given that the optimality of the SATU* has been proven above and that the graphs used have the same properties than for the AUCS*, we can directly conclude that the SATU* is complete (as the graphs used are locally finite). □

5 CONCLUSION

In this paper, we firstly presented a simple way to run an A* algorithm in an uncertain-configuration space (AUCS*) by considering the uncertainties as simple degree of freedom of the system. This characteristic allows the add of uncertainties in any path planner algorithm that respects the necessary properties.

Secondly, we introduced a new path planner, the SATU* algorithm, working in an uncertain-configuration space, strongly based on the A* algorithm that limits the memory needed. We then gave the needed proofs of optimality and completeness associated. Some further work will focalize on the calculation and comparison of the AUCS* and SATU* complexities and on the experimental tests using a real mobile robot.

REFERENCES

Dijsktra, E. (1959). A note on two problems in connexion with graphs. In *Numerische Mathematik*, volume 1, pages 269–271.

Fraichard, T. and Mermond, R. (May 1998). Path planning with uncertainty for car-like robots. In *IEEE Int. Conf. on Robotic and Automation*, pages 27–32.

Hart, P., Nilsson, N., and Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. In *IEEE Transaction on Systems Science and Cybernetics, SSC-4(2)*, pages 100–107.

Lambert, A. and Gruyer, D. (September 14-19, 2003). Safe path planning in an uncertain-configuration space. In *IEEE International Conference on Robotics and Automation*, pages 4185–4190.

Latombe, J. C. (1991). *Robot Motion Planning*. Kluwer Academic Publisher.

Lazanas, A. and Latombe, J. C. (1995). Motion planning with uncertainty : a landmark approach. In *Artificial Intelligence 76(1-2)*, pages 287–317.

Nilsson, N. J. (1988). *Principles of artificial intelligence*. Morgan Kaufman Publisher inc.

Pepy, R. and Lambert, A. (2006). Safe path planning in an uncertain-configuration space using rrt. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5376–5381, Beijing, China.

Russel, S. J. and Norvig, P. (1995). *Artificial Intelligence: A modern Approach*, pages 92–101. Prentice Hall, Inc.

FORWARD KINEMATICS AND GEOMETRIC CONTROL OF A MEDICAL ROBOT
Application to Dental Implantation

Richard Chaumont, Eric Vasselin and Dimitri Lefebvre
GREAH of LE HAVRE University, 25 rue Philippe LEBON. 76058 LE HAVRE Cede, France
richard.chaumont436@univ-lehavre.fr, eric.vasselin@ univ-lehavre.fr, dimitri.lefebvre@ univ-lehavre.fr

Keywords: Surgical robotics, geometric modelling, inverse geometric modelling, dynamic modelling, non linear system, identification.

Abstract: Recently, robotics has found a new field of research in surgery in which it is used as an assistant of the surgeon in order to promote less traumatic surgery and minimal incision of soft tissue. In accordance with the requirements of dental surgeons, we offer a robotic system dedicated to dental implants. A dental implant is a mechanical device fixed into the patient's jaw. It is used to replace a single tooth or a set of missing teeth. Fitting the implant is a difficult operation that requires great accuracy. This work concerns the prototype of a medical robot. Forward and inverse kinematics as dynamics are considered in order to drive a control algorithm which is as accurate and safe as possible.

1 INTRODUCTION

Computer-assisted dental implantology is a multidisciplinary and complex topic that includes medical imagery, robotics and computer vision (Langlotz, & *al*, 2000 ; Nikou & *al*, 2000). The fitting of a dental implant is currently the only technique suitable to permanently restore the teeth. For this purpose, specific surgery has been recently developed. Such operations require great accuracy. Moreover, the spread of this type of surgery justifies the extension and use of new techniques (Taylor, 1994). This research and development work focuses on the medical robotics applied to dental implantology. The main contributions of this article are to discuss the forward kinematics and kinematics uncertainties, and also to provide a geometric control for the orientation of the drill.

2 MEDICAL ROBOTICS

For the last twenty years, new technologies have been used to improve surgical operations so that medical research and engineering improvements are closely linked today. On one hand, data processing, computer vision and medical imaging are used in an intensive way in operation rooms. On the other hand, the three principles of robotics-perception, reasoning and action - have been adapted for medical and surgery issues (Lavallee & *al*, 1995). The main goal is to bring together the fundamental principles of robotics and computer vision in order to assist the surgeon in daily therapeutic operations. The aims of medical robotics are to provide less traumatic investigation systems, to provide simulation tools, and, finally, to provide tools that are easier and more flexible to use.

The aim of dental implantology is to use bones and implants in order to provide prosthetic support. The main advantage in comparison with a conventional prosthesis is that dental implantology doesn't mutilate healthy teeth. At the time tooth extraction is completed, the fitting of a dental implant allows the consolidation of the prostheses. The main difficulty is to place the implants correctly. That is the reason why conventional prosthesis is still prefered to dental implantology in many cases.

Dental implants guarantee the patient better comfort but can also reduce overall cost owing to their longevity and lack of inherent complications in comparison with classic prostheses. For difficult cases (completely toothless patients, weak density bones, multiple implants, etc), dental surgeons are confronted with a complex operation. Over the years, the main difficulty concerned the integration of the bone-implant.

This problem has been solved by technical improvements and surgical advances (equipment, implant shapes, surgical protocol, etc). According to the opinion of many clinicians, the difficulty is henceforth to improve fitting techniques. The position and orientation of implants must take into account biomechanical and anatomical constraints (Dutreuil, 2001) involving three main criteria: mastication, phonetics and aesthetics. In particular, the problems to be solved are :

- How to adjust the implant position in the correct axis.
- How to optimize the relative position of two adjacent implants.
- How to optimize the implant position according to the bone density.
- How to conduct minimal incision of soft tissue.

Our answer to these questions is image guided surgery (Etienne & *al*, 2000). This solution uses an optical navigation system with absolute positioning in order to obtain position and orientation of the surgeon's tool in real time with respect to the patient's movements. The operation is planned on the basis of scanner data or x-ray images for simple clinical cases.

Figure 1: Navigation system.

The technique consists in initializing the superimposing of patient data and data derived form a set of specific points attached to the patient's jaw (Granger, 2003). The patient's jaw is then analyzed in real time by the navigation system. The Dental View navigation machine guides the surgeon via the image during the operative phase.

However, clinical tests show that supplementary assistance is necessary to help the surgeon during the drilling phase in order to fulfil precision requirements. For this purpose, we developed a surgical robot which controls orientation during the drilling phase.

Our robot is a semi-active mechanical device. It has a passive arm and a motorized wrist with three degrees of freedom (dof) that are not convergent (i.e. not a spherical wrist). The basis is passive, that is to say it is not motorized and can be manipulated by the surgeon like an instrument. The aim of the controller is to guide the surgeon so that it will respect the scheduled orientation.

3 FORWARD KINEMATICS

Drill orientation is characterized by 3 dof RotY, RotX and RotZ and depends also on contra angle ca (Fig. 3). The structure is represented in Fig.2.

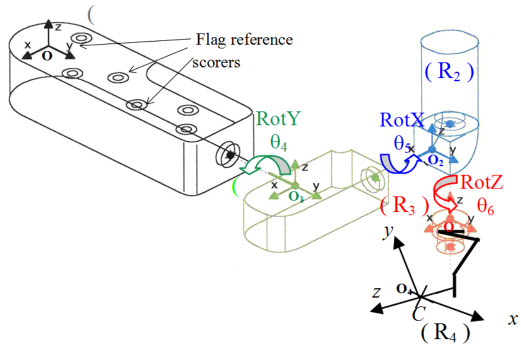


Figure 2: Robot axis.

3.1 Notations

We use homogeneous transformations to describe position and orientation from one link to another. Let us consider the matrix M :

$$M = \begin{bmatrix} R & T \\ P & Q \end{bmatrix} \qquad (1)$$

with P = [0 0 0], Q = [1] is a homothetic coefficient equal to 1 (orthogonal transformation); R is a orthogonal rotation matrix; and T is a translation matrix.

For simplicity, calculations are not given in detail. Notations will be represented above Fig.3.

Lxo, Lyo, Lzo : distance x, y, z between computer vision coordinate frame and 4^{th} joint coordinate frame.

Lx5, Lz6 : distance x, z between 4^{th} joint and 5^{th} joint.

Lyca, Lzca : distance y, z between 5 th joint and effector.

Lztool : drill length.

θ_4, θ_5, θ_6 : wrist joint variables.

Ca: contra angle.

ε_i: orthogonal uncertainty between joints.

l_i: length uncertainty between links.

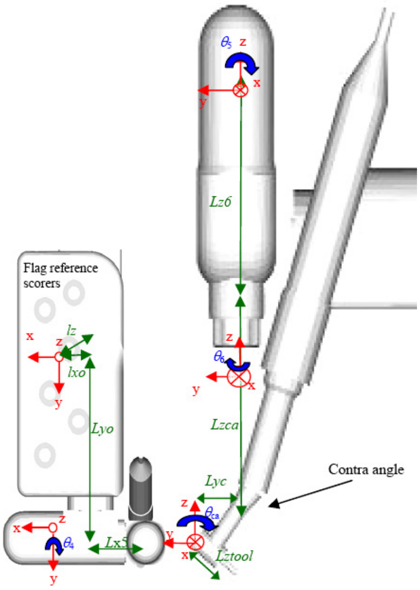


Figure 3: Robot parameters and axes.

3.2 Kinematic Uncertainties

First, we are going to determine the effector position in the ideal case, without considering uncertainties related to length and orthogonality links. We use homogeneous transformations to change the coordinate frame attached to a joint to the coordinate frame attached to the next one. We obtain 6 matrices that change the coordinate frames according to (2) :

$$A_1=\begin{pmatrix}1&0&0&Lxo\\0&1&0&Lyo\\0&0&1&Lzo\\0&0&0&1\end{pmatrix}\quad A_2=\begin{pmatrix}\cos\theta_4&0&\sin\theta_4&0\\0&1&0&0\\-\sin\theta_4&0&\cos\theta_4&0\\0&0&0&1\end{pmatrix}\quad A_3=\begin{pmatrix}1&0&0&Lx5\\0&\cos\theta_5&-\sin\theta_5&0\\0&\sin\theta_5&\cos\theta_5&0\\0&0&0&1\end{pmatrix}$$
$$A_4=\begin{pmatrix}\cos\theta_6&-\sin\theta_6&0&0\\\sin\theta_6&\cos\theta_6&0&0\\0&0&1&Lz6\\0&0&0&1\end{pmatrix}\quad A_5=\begin{pmatrix}1&0&0&0\\0&\cos(ca)&-\sin(ca)&0\\0&\sin(ca)&\cos(ca)&0\\0&0&0&1\end{pmatrix}\quad A_6=\begin{pmatrix}1&0&0&0\\0&1&0&Lyca\\0&0&1&Lzca\\0&0&0&1\end{pmatrix}\quad(2)$$

Ideal position in flag coordinate frame results from the preceding matrices :

$$V_{ideal}(R_0)=A_1\times A_2\times A_3\times A_4\times A_5\times A_6\times V(R_4)\qquad(3)$$

with V(R$_4$) = [0 0 1 0]^T to get effector orientation for the “z” axis and V(R$_4$) = [0 0 0 1] ^{T} to get effector position.

$$A_1^{'}=\begin{pmatrix}1&0&0&Lxo\pm l_{xo}\\0&1&0&Lyo\pm l_{yo}\\0&0&1&Lzo\pm l_{zo}\\0&0&0&1\end{pmatrix}$$

$$A_2^{'}=\begin{pmatrix}1&0&0&0\\0&\cos\pm\varepsilon_1&-\sin\pm\varepsilon_1&0\\0&\sin\pm\varepsilon_1&\cos\pm\varepsilon_1&0\\0&0&0&1\end{pmatrix}\times\begin{pmatrix}\cos\theta_4&0&\sin\theta_4&0\\0&1&0&0\\-\sin\theta_4&0&\cos\theta_4&0\\0&0&0&1\end{pmatrix}\times\begin{pmatrix}\cos\pm\varepsilon_2&-\sin\pm\varepsilon_2&0&0\\\sin\pm\varepsilon_2&\cos\pm\varepsilon_2&0&0\\0&0&1&0\\0&0&0&1\end{pmatrix}$$
$$A_3^{'}=\begin{pmatrix}1&0&0&Lx5\pm l_{x5}\\0&\cos\theta_5&-\sin\theta_5&0\\0&\sin\theta_5&\cos\theta_5&0\\0&0&0&1\end{pmatrix}\times\begin{pmatrix}\cos\pm\varepsilon_3&0&\sin\pm\varepsilon_3&0\\0&1&0&0\\-\sin\pm\varepsilon_3&0&\cos\pm\varepsilon_3&0\\0&0&0&1\end{pmatrix}\times\begin{pmatrix}\cos\pm\varepsilon_4&-\sin\pm\varepsilon_4&0&0\\\sin\pm\varepsilon_4&\cos\pm\varepsilon_4&0&0\\0&0&1&0\\0&0&0&1\end{pmatrix}$$
$$A_4^{'}=\begin{pmatrix}1&0&0&0\\0&\cos\pm\varepsilon_5&-\sin\pm\varepsilon_5&0\\0&\sin\pm\varepsilon_5&\cos\pm\varepsilon_5&0\\0&0&0&1\end{pmatrix}\times\begin{pmatrix}\cos\pm\varepsilon_6&0&\sin\pm\varepsilon_6&0\\0&1&0&0\\-\sin\pm\varepsilon_6&0&\cos\pm\varepsilon_6&0\\0&0&0&1\end{pmatrix}\times\begin{pmatrix}\cos\theta_6&-\sin\theta_6&0&0\\\sin\theta_6&\cos\theta_6&0&0\\0&0&1&Lz6\pm l_{z6}\\0&0&0&1\end{pmatrix}$$
$$A'_5=\begin{pmatrix}1&0&0&0\\0&1&0&Lyca\pm l_{yca}\\0&0&1&Lzca\pm l_{zca}\\0&0&0&1\end{pmatrix}$$
$$A_6^{'}=\begin{pmatrix}1&0&0&0\\0&\cos(ca)&-\sin(ca)&0\\0&\sin(ca)&\cos(ca)&0\\0&0&0&1\end{pmatrix}\times\begin{pmatrix}\cos\pm\varepsilon_5&0&\sin\pm\varepsilon_5&0\\0&1&0&0\\-\sin\pm\varepsilon_5&0&\cos\pm\varepsilon_5&0\\0&0&0&1\end{pmatrix}\times\begin{pmatrix}\cos\pm\varepsilon_6&-\sin\pm\varepsilon_6&0&0\\\sin\pm\varepsilon_6&\cos\pm\varepsilon_6&0&0\\0&0&1&0\\0&0&0&1\end{pmatrix}\qquad(4)$$

We will model actual position and orientation of the effector by considering uncertainties relating to link length (maximum tolerance of 0.1 mm) and also of frame orthogonal uncertainties (maximum tolerance 0.1 degree). Because of digital encoder resolution (200 000 points per revolution) and of gear reduction encoders uncertainties are negligible.

Moreover, we only consider uncertainties that concern the robot wrist (the arm is a passive mechanical structure). For every dof two uncertainty matrices are added according to the axes that are not articulated. Equation (4) is obtained as a consequence.

Actual effector position is given by (5):

$$V_{actual}(R_0)=A'_1\times A'_2\times A'_3\times A'_4\times A'_5\times A'_6\times V(R_4)\qquad(5)$$

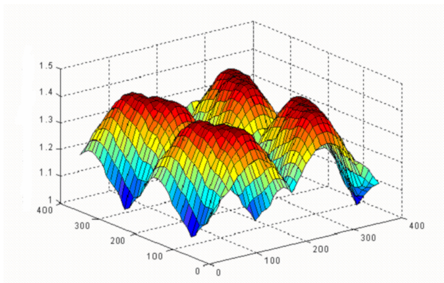


Figure 4: Magnitude of error function for θ_4 and θ_5.

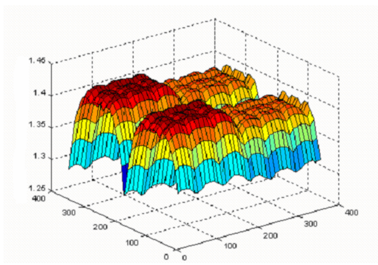


Figure 5: Magnitude of error function for θ_4 and θ_6.

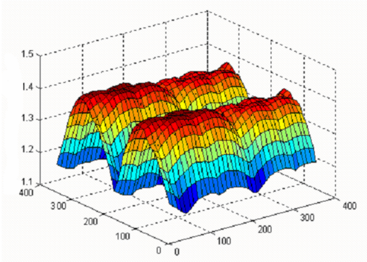


Figure 6: Magnitude of error function for θ_5 and θ_6.

From (4) and (5), we know the ideal position and orientation of the wrist as well as its actual position and orientation which take kinematic uncertainties into account. Therefore, we can obtain the position and orientation errors :

$$E(R_0) = V_{actual}(R_0) - V_{ideal}(R_0) \qquad (6)$$

Figs. 4 to 6 represent the maximal uncertainty magnitude (millimeter) generated by mechanical and assembly tolerances. One observes that :

- Uncertainty magnitude is always superior to 1 millimeter,
- Uncertainty magnitude can reach 1.4 millimeters for particular link positions.

In order to fulfil precision requirements uncertainties must be lower than 1 millimeter. Therefore, it is necessary to calibrate the robot wrist.

4 DYNAMICS

In this section we will propose a dynamic model for the axis of the robot. The axis control architecture is presented in Fig.7. Technical caracteristics of the electromechanical and electronic devices can be found in (Chaumont & *al*, 2006).

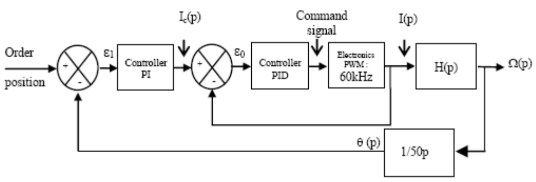


Figure 7: Axis control structure.

4.1 Identification of Electromechanical Device

Closed loop identification for electromechanical device is proposed in this section (Richalet, 1998). System output is the velocity and system input is the current. The robot axis has been placed so that inertia moment can be considered as constant whatever the orientation.

The time response (Fig. 8) presents a dissymmetry between the current generating the acceleration in comparison with the current generating the deceleration.

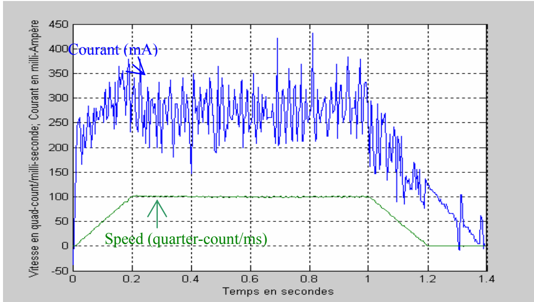


Figure 8: Protocol signature.

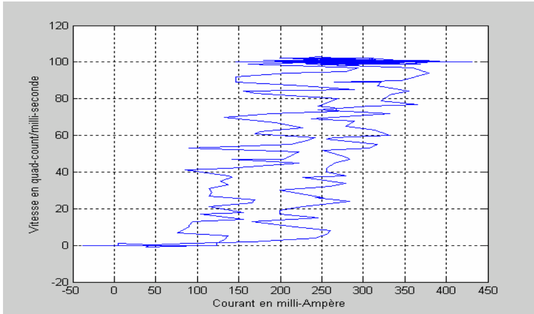


Figure 9: Hysteresis system.

The process is non linear. Fig. 9 represents input/output signature with a dead zone and a histereses. The electromechanical transfer function is represented in Fig.13.

4.2 Identification of Electronic Device

Electronic device input is the desired current and output is the actual current. It represents the electronic system part that is composed of the PWM and its controller.

Identification is achieved in closed loop. A survey of electronic control shows us that the transfer function is a second order overshoot with a stable zero.

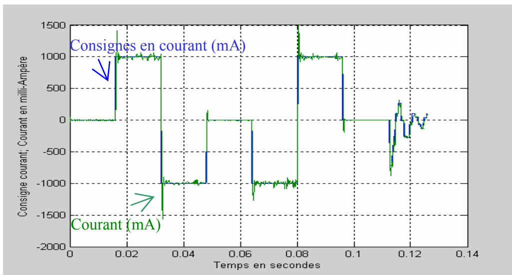


Figure 10: Protocol signature.

4.3 Discussion

The model is validated with the same desired current. Results obtained with the model and with the system are compared according to velocity kinetics (Fig. 11) and position (Fig. 12).

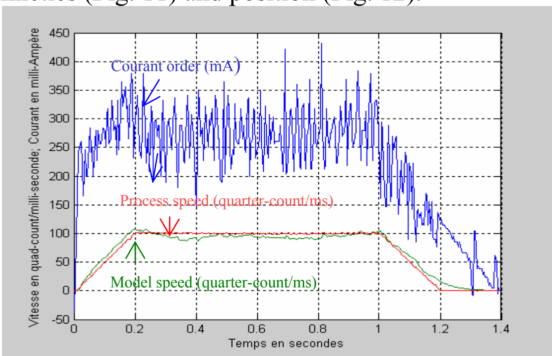


Figure 11: Velocity response.

Parameters

$\omega_0 = 730.97$; $\xi = 0.4258$; $K = 0.9955$; $k = 0.5969$; $T = 0.539$

Histereses size: 75 mA Dead zone size: 115 mA.

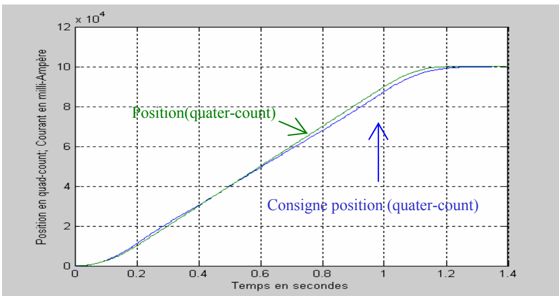


Figure 12: Position response.

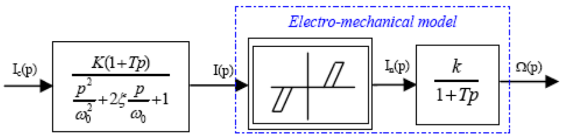


Figure 13: Axis module.

5 CONTROL DESIGN

Forward kinematics is given by equation (2). This model shows how to determine the effector's position and orientation in terms of the joint variables with a function "f". Inverse kinematics is concerned with the problem of finding the joint variables in terms of effector's position and orientation. Our aim is to control the orientation of the effector. The function "f" corresponds therefore to V_{ideal} (R_0) expressed by equation (3) with $V(R_4) =$ $[0\ \ 0\ \ 1\ \ 0]^T$ (uncertainties are not considered for control design).

Matrix V_{ideal} (R_0) leads to equations (7) to (9) :

$$Vx_0 = \sin(ca).\cos(\theta_4).\sin(\theta_6) - \sin(ca).\sin(\theta_4).\sin(\theta_5).\cos(\theta_6) + \sin(\theta_4).\cos(\theta_5).\cos(ca) \quad (7)$$

$$Vy_0 = -\cos(\theta_5).\cos(\theta_6).\sin(ca) - \sin(\theta_5).\cos(ca) \quad (8)$$

$$Vz_0 = -\sin(ca).\sin(\theta_4).\sin(\theta_6) - .\cos(\theta_4).\sin(\theta_5).\cos(\theta_6) + \cos(\theta_4).\cos(\theta_5).\cos(ca) \quad (9)$$

From equations (7), (8) and (9), expressions θ_4, θ_5 and θ_6 are determined.

$$\theta_6 = \operatorname{asin}\frac{\cos(\theta_4).Vx_0 - \sin(\theta_4).Vz_0}{\sin(ca)},\ \theta_{6a} = \pi - \theta_6 \quad (10)$$

$$\theta_5 = \operatorname{acos}\left(\frac{Vy_0}{\rho_1}\right) + a_1,\ \theta_{5a} = -\operatorname{acos}\left(\frac{Vy_0}{\rho_1}\right) + 2.a_1 \quad (11)$$

with : $\rho_1 = \sqrt{(\cos(\theta_6).\sin(ca))^2 + \cos^2(ca)}$

$$a_1 = \operatorname{atan}\left(\frac{\cos(ca)}{\cos(\theta_6).\sin(ca)}\right) + \pi \quad \text{if } -\cos(\theta_6).\sin(ca) < 0$$

$$\text{or } a_1 = \operatorname{atan}\left(\frac{\cos(ca)}{\cos(\theta_6).\sin(ca)}\right) \quad \text{if } -\cos(\theta_6).\sin(ca) > 0$$

$$\theta_{5b} = \operatorname{acos}\left(\frac{Vy_0}{\rho_2}\right) + a_2,\ \theta_{5c} = -\operatorname{acos}\left(\frac{Vy_0}{\rho_2}\right) + 2.a_2 \quad (12)$$

with : $\rho_2 = \sqrt{(\cos(\theta_{6a}).\sin(ca))^2 + \cos^2(ca)}$

$$a_2 = \operatorname{atan}\left(\frac{\cos(ca)}{\cos(\theta_{6a}).\sin(ca)}\right) + \pi \quad \text{if } -\cos(\theta_{6a}).\sin(ca) < 0$$

$$\text{or } a_2 = \operatorname{atan}\left(\frac{\cos(ca)}{\cos(\theta_{6a}).\sin(ca)}\right) \quad \text{if } -\cos(\theta_{6a}).\sin(ca) > 0$$

$$\theta_4 = \operatorname{acos}\frac{\sin(ca).\sin(\theta_6)}{\sigma} + b$$

$$\theta_{4a} = -\operatorname{acos}\frac{\sin(ca).\sin(\theta_6)}{\sigma} + 2.b$$

$$\theta_{4b} = \operatorname{acos}\frac{\sin(ca).\sin(\theta_{6a})}{\sigma} + b$$

$$\theta_{4c} = -\operatorname{acos}\frac{\sin(ca).\sin(\theta_{6a})}{\sigma} + 2.b \quad (13)$$

with : $\sigma = \sqrt{Vx_0^{\ 2} + Vz_0^{\ 2}}$

$$b = -\operatorname{atan}\left(\frac{Vz_0}{Vx_0}\right) - \pi \quad \text{if } Vx_0 < 0$$

$$b = -\operatorname{atan}\left(\frac{Vz_0}{Vx_0}\right) \quad \text{if } Vx_0 > 0$$

Equations (10) to (13) show that angle θ_6 depends on θ_4, that angles θ_4 and θ_5 depend on θ_6. A robot is "resolvable" if a unique solution exists for equation $\theta = f^{-1}(x)$. Our study suggest that our medical robot's wrist is not resolvable.

5.1 Reachable Workspace

Analysis of the resulting equations shows that the determination of the desired current according to a given set of joint variables θ_4, θ_5 and θ_6 is difficult.

Indeed, a given orientation has several solutions for joint variables. Fig. 14 illustrates these points by defining the reachable workspace. The method we propose consists in considering θ_4 as a constant parameter in order to work out joint variables θ_5 and θ_6 according to θ_4. The choice of θ_4 is motivated by optimization method.

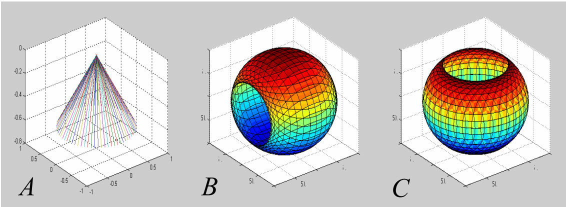


Figure 14: Reachable workspace.

5.2 Control Strategy

The controller must satisfy real time requirements. The sampling frequency is 20 Hz. Moreover, several applications such as artificial vision and data processing take place simultaneously. The controller inputs are :

- Data from artificial vision and image superimposition,
- Desired drill orientation,
- Actual angular values returned by digital encoders.

The algorithm determines the values of V_{xo}, V_{yo}, and V_{zo} that are vector components in the flag reference scorers of the robot. It determines θ_5 and θ_6 with respect to θ_4 and verifies that solutions are in the reachable workspace. If solutions are outside the reachable workspace, the algorithm increments θ_4 with 1° and recalculates θ_5 and θ_6. Incrementation is repeated until a solution is found in the reachable workspace.

6 FURTHER WORKS

The protocol used for identification will be applied in a generic way on the other axes in order to obtain a dynamic model of the robot. As a consequence, we will be able to simulate the robot's dynamic behaviour and to develop safe and efficient control design. On the other hand, our work will concern the following points :

- Accuracy, wrist calibration,
- Study of position / orientation decoupling,
- Trajectory planning, ergonomics.

This medical robot is an invasive and semi-active system. Therefore, an exhaustive study on reliability will also be necessary (Dombre, 2001) before starting clinical simulations and experimentations.

ACKNOWLEDGEMENTS

Authors thank the Doctor DERYCKE, a dental surgeon specialized in implantology, CHIEF EXECUTIVE OFFICER of DENTAL VIEW.

REFERENCES

Langlotz F, Berlemann U, Ganz R, Nolte LP., 2000. Computer-Assisted Periacetabular Osteotomy. Oper. *Techniques in Orthop*, vol 10, N° 1, pp. 14-19

Nikou C, Di Gioia A, Blackwell M, Jaramaz B, Kanade T., 2000. Augmented Reality Imaging Technology for Orthopaedic Surgery. Oper. *Techniques in Orthop*, vol 10, N° 1, pp. 82-86.

Taylor R.,1994. An Image Directed Robotic System for Precise Orthopaedic Surgery. *IEEE Trans Robot*, 10, pp. 261-275.

Lavallee S, Sautot P, Traccaz J, P. Cinquin P, Merloz P., 1995. Computer Assisted Spine Surgery : a technique for accurate transpedicular screw fixation using CT data and a 3D optical localizer. *Journal of Image Guided Surgery*.

Dutreuil J., 2001. Modélisation 3D et robotique médicale pour la chirurgie. *Thèse*, Ecole des Mines de Paris.

Etienne D, Stankoff A, Pennnec X, Granger S, Lacan A, Derycke R., 2000. A new approach for dental implant aided surgery. *The 4th Congresso Panamericano de Periodontologia*, Santiago, Chili.

Granger S., 2003. Une approche statistique multi-échelle au recalage rigide de surfaces : Application à l'implantologie dentaire. *Thèse*, Ecole des Mines de Paris.

Chaumont R, Vasselin E, Gorka M, Lefebvre D., 2005. Robot médical pour l'implantologie dentaire : identification d'un axe, *JNRR05*. Guidel, pp 304-305.

Richalet J., 1998. Pratique de l'identification. Ed. Hermes.

Dombre E., 2001. Intrinsically safe active robotic systems for medical applications. *1st IARP/IEEE-RAS Joint Workshop on Technical Challenge for Dependable Robots in Human Environments*, Seoul, Korea.

HIERARCHICAL SPLINE PATH PLANNING METHOD FOR COMPLEX ENVIRONMENTS

Martin Saska, Martin Hess and Klaus Schilling
University of Wuerzburg, Germany
Informatics VII, Robotics and Telematics
Am Hubland, Wuerzburg, Germany, 97074
{*saska,m.hess,schi*}*@informatik.uni-wuerzburg.de*

Keywords: Path planning, Mobile robots, PSO, Spline path, Hierarchical approach.

Abstract: Path planning and obstacle avoidance algorithms are requested for robots working in more and more complicated environments. Standard methods usually reduce these tasks to the search of a path composed from lines and circles or the planning is executed only with respect to a local neighborhood of the robot. Sophisticated techniques allow to find more natural trajectories for mobile robots, but applications are often limited to the offline case.

The novel hierarchical method presented in this paper is able to find a long path in a huge environment with several thousand obstacles in real time. The solution, consisting of multiple cubic splines, is optimized by Particle Swarm Optimization with respect to execution time and safeness. The generated spline paths result in smooth trajectories which can be followed effectively by nonholonomic robots.

The developed algorithm was intensively tested in various simulations and statistical results were used to determine crucial parameters. Qualities of the method were verified by comparing the method with a simple PSO path planning approach.

1 INTRODUCTION

Expanding mobile robotics to complicated environments increases the requirements on the control system of the robot. The desired task has to be executed by the robot as fast as possible in these scenarios. Standard path planning approaches very often just provide simple piecewise constant routes (Latombe, 1996),(Kunigahalli and Russell, 1994),(Franklin et al., 1985) or only local optimal solutions (Borenstein and Koren, 1991),(Khatib, 1986). Few methods that give continues and smooth solution (Azariadis and Aspragathos, 2005), (Nearchou, 1998), due to long computational time in order of minutes, cannot be used in real time applications.

All path planning methods try to find a path from an actual position S of the controlled robot to a desired goal position G, with regard to position and shape of known obstacles O. While these parameters stand as the inputs of the algorithm, the output can be either an optimal path from S to G or just a direction from the actual position respecting locally optimal trajectory. The key issue of the path planning for mobile robots is how to define the best trajectory. A common answer to this is that the optimal path should be similar to a path designed by a human operator. This vague definition can be expressed by a penalty function that is minimized during the planning. The simplest version of the function consists of two parts. While the first one evaluates a length of the path (or time needed to goal fulfilment), the second part ensures safety of the path (i.e. sufficient distance to obstacles). So finding an acceptable compromise between these requirements is the core problem of the path planning itself.

The novel approach presented in this paper offers a solution to the above mentioned problems. The output of the algorithm is a path, which consists of smoothly connected cubic splines. Spline paths are naturally executable by the robot and optimization of the speed profile can be easily done by modifications of the radius of the curvature. Global information about the workspace is subsumed, because the whole space of spline trajectories from S to G will be searched through during the optimization process. Therefore big clusters of obstacles as well as huge obstacles in the environment can be avoided immediately. This enables us to locate the final solution in the areas with lower density of obstacles at the beginning of the mission. Such a path may seem longer, due to necessary deviations from the ”bee-line”, but the final movement time can be shorter, because difficult maneuvers in the obstacle clusters are eliminated.

Particle Swarm Optimization (PSO) (Eberhart and Kennedy, 1995) was used for prospecting the optimal solution hereunder due to its relatively fast convergence, its global search character and its ability to avoid local minima. In this paper is supposed real time path planning and therefore the solution must be given in a split second. The complexity of PSO strongly depends on the dimension of the search space and on the needed swarm size. The number of iterations increase more than exponentially with the dimension. Accordingly a minimal length of the particle is required, but in complex environments a short string of cubic splines is not sufficient and collision free path in such limited space often does not exist.

In the proposed algorithm, that is based on a hierarchical approach, the PSO process is used several times. Each optimization proceeds in a subspace with a small dimension, but the total size of the final solution can be unlimited. The path obtained by this method is only suboptimal, but the reduction of the computational time is incomparable bigger. In addition the algorithm can be started without prior knowledge about the complexity of the environment, because the dimension of the search space is specified during the planning process. The dimension even may not be uniform in the whole workspace, but it is automatically adapted according to the number of collisions along the path, that is found by the hierarchically plunging.

The paper is organized as followed: A novel method reflecting special requirements for the mobile robots in a complex environment is presented in section 2. Section 2.1 provides information about the universal PSO approach. In section 2.2 adaptations of the PSO to our specific problem are described. After this experimental results are presented in section 3 followed by concluding words and plans for future work in section 4.

2 PATH PLANNING

The path planning approach presented in this paper is based on a searching through a space of states P. Each state P_i is represented by a vector $P_i = (P_{i,x}, P_{i,y}, P'_{i,x}, P'_{i,y})$, that denotes position and heading of the robot in the workspace. The final path of the robot is set by a transition $P_i \xrightarrow{T_i} P_{i+1}$ between each pair of neighboring states. T_i can be defined uniquely, if $T \equiv C^1$, where C^1 is the set of cubic splines with smooth first derivation (closely described in 2.2).

The system of all solutions Π is a vector $\Pi = < S, G, p, t, f(t) >$, where $S, G \in P$ are start and goal states of the robot and $p \subset P$ is a sequence of the states between S and G. $t \subset T$ is a string of the splines connecting S and G and $f(t)$ is the fitness function denoting the quality of the found path. The solution $\pi \in \Pi$ with minimum value of $f(t)$ should conform with the best path.

The crucial problem is to find π with minimum or nearly minimum value of the fitness function in the usually vast set Π. The complexity of the task grows with the size of vector p. Solutions with a small dimension of p can be found faster, but a collision free path may not exist in such limited space, if the workspace of the robot is too complicated.

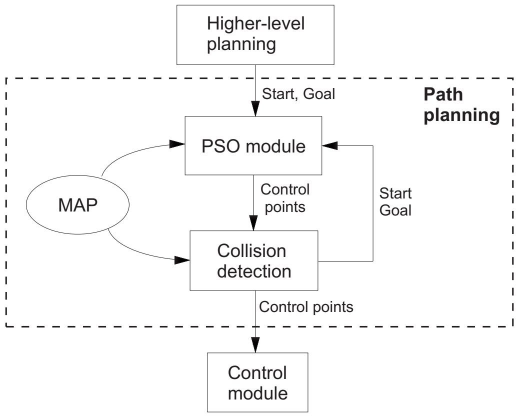


Figure 1: Schema of the presented hierarchical method.

In the hierarchical approach presented in this paper the optimization task is decomposed to multiple subtasks. The basic idea of the algorithm can be seen in Fig. 1. The actual state of the robot and the desired state generated by a higher planning module are the inputs for the Particle Swarm Optimization (described in 2.1). The path t which was found by PSO is checked in the collision detection module and a collision free solution is sent to the control module, where the path is executed. If a collision is detected, the string of splines t is divided and control points of each spline are used as new input for the PSO module. These points are put into the memory of the module sequently. For other optimization processes always the last added points are chosen. This ”LIFO” approach guarantees that the path close to the robot will be found as soon as possible. The remaining path will be found during the robot's movement. Therefore the time needed for the initial planning before the start of the mission is reduced several times.

An example of the path decomposition is shown in Fig. 2. Each vector p used in the optimization processes contains only two states P_i which correspond to three splines in the string. In this simplified example the hierarchical process is stopped always in the third level, independently of the number of collisions.

The input states in each particle are printed with dark background (e.g. S and G in the first level) whereas the optimized states are represented by letters in white rectangles (e.g. A and B in this level). The final path in the lowest level consists of 27 splines and it is characterized by 28 states of the robot. Such a path can be found as an extreme of the nonlinear fitness function in 104-dimensional space (26 optimized states multiplied by 4 numbers describing each state), which is difficult (or impossible) in real time applications. In the hierarchical approach the scanned space is 13 times reduced and the separate optimization subtask can be done in tens of iterations. Furthermore only three optimization processes (in the picture marked by $SABG$, $SCDA$, $SJKC$) are necessary for the beginning of the mission.

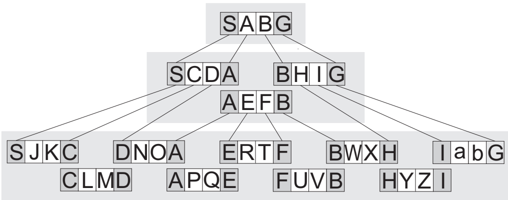


Figure 2: Example of the path decomposition.

2.1 Particle Swarm Optimization

The PSO method was developed for finding a global optimum of a nonlinear function (Kennedy and Eberhart, 1995),(Macas et al., 2006). This optimization approach has been inspired by the social behavior of birds and fish. Each solution consists of set of parameters and represents a point in a multidimensional space. The solution is called ”particle” and the group of particles (population) is called ”swarm”.

Two kinds of information are available to the particles. The first is their own experience, i.e. their best state and it's fitness value so far. The other information is social knowledge, i.e. the particles know the momentary best solution p_g of the group found during the evaluation process.

Each particle i is represented as a D-dimensional position vector $\overrightarrow{x}_i(q)$ and has a corresponding instantaneous velocity vector $\overrightarrow{v}_i(q)$. Furthermore, it remembers its individual best value of fitness function and position $\overrightarrow{q}_i$ which has resulted in that value.

During each iteration t, the velocity update rule (1) is applied on each particle in the swarm.

$$\begin{aligned}\overrightarrow{v}_i(q) = w\overrightarrow{v}_i(q-1) + \\ + \Phi_1(\overrightarrow{p}_i - \overrightarrow{x}_i(q-1)) + \\ + \Phi_2(\overrightarrow{p}_g - \overrightarrow{x}_i(q-1))\end{aligned} \quad (1)$$

The parameter w is called inertia weight and during all iterations decreases linearly from w_{start} to w_{end}. The symbols Φ_1 and Φ_2 are computed according to the equation

$$\Phi_j = \varphi_j \begin{pmatrix} r_{j1} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & r_{jD}, \end{pmatrix} \quad (2)$$

where $j = 1, 2$. The parameters φ_i are constants that weight the influence of the particles' own experience and of the social knowledge. In our experiments, the parameters were set to $\varphi_1 = 2$ and $\varphi_2 = 2$. The r_{jk}, where $k = 1 \ldots D$ are random numbers drawn from a uniform distribution between 0 and 1.

The position of the particles is then computed according to the update rule

$$\overrightarrow{x}_i(q) = \overrightarrow{x}_i(q-1) + \overrightarrow{v}_i(q). \quad (3)$$

If any component of $\overrightarrow{v}_i$ is less than $-V_{max}$ or greater than $+V_{max}$, the corresponding value is replaced by $-V_{max}$ or $+V_{max}$, respectively, where V_{max} is the maximum velocity parameter.

Influencing V_{max} in the simple PSO path planning method (Saska et al., 2006b) illustrated a strong correlation between the optimal value of the parameter and the space's size. This size is different for each optimization process in the hierarchical tree, because it is constrained by the position of the points S and G. Therefore the value of V_{max} is computed by the equation

$$V_{max} = \frac{||S - G||}{c_V}, \quad (4)$$

where c_V is a constant that has to be found experimentally.

The update formulas (1) and (3) are applied during each iteration and the values of p_i and p_g are updated simultaneously. The algorithm is stopped if the maximum number of iterations is reached or any other predefined stopping criteria is satisfied.

2.2 Particle Description and Evaluation

The path planning for a mobile robot can be realized by a search in the space of functions. We reduce this space to a sub-space which only contains strings of cubic splines. The mathematic notation of a cubic spline (Ye and Qu, 1999) is

$$g(t) \quad = \quad At^3 - Bt^2 + Ct + D, \quad (5)$$

where t is within the interval $< 0, 1 >$ and the constants A, B, C, D are uniquely determined by the boundary conditions $g(0) = P_0$, $g(1) = P_1$, $g'(0) = P'_0$ and $g'(1) = P_1$:

$$A = 2P_0 - 2P_1 + P_0' + P_1' \tag{6}$$
$$B = -3P_0 + 3P_1 - 2P_0' - P_1' \tag{7}$$
$$C = P_0' \tag{8}$$
$$D = P_0. \tag{9}$$

To guarantee continuity in the whole path, every two neighboring splines in the string must share one of their terminal states and therefore the actual and the desired states are defined by initial conditions. The total number of variables defining the whole path in 2D is therefore only $4(n-1)$, where n denotes the amount of splines in the string. The structure of the particles used in the optimization process is shown in Fig. 3.

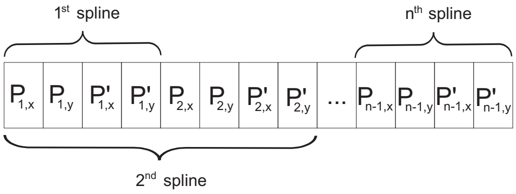


Figure 3: Structure of one particle composed of multiple spline parameters.

The best individual state and the best state achieved by the whole swarm is identified by the fitness function. The global minimum of this function corresponds to a smooth and short path that is safe (i.e. there is sufficient distance to obstacles).

Two different fitness functions were used in this hierarchical approach. The populations used in the last hierarchical level (the lowest big rectangle in figure 2) are evaluated by the fitness function

$$f_1 = f_{length} + \alpha f_{collisions}. \tag{10}$$

The particles in the other levels are evaluated by the extended fitness function

$$f_2 = f_{length} + \alpha f_{collisions} + \beta f_{extension}, \tag{11}$$

where the component $f_{extension}$ pushes the control points of the splines P_i to the obstacle free space. These points are fixed for the lower level planning and therefore collisions close to these points cannot be repaired. The function $f_{extension}$ is defined as

$$f_{extension} = \begin{cases} \delta^{-2}, & \text{if} \quad d_m < \delta \\ \delta^{-2} + p_{inside}, & \text{else} \end{cases} \tag{12}$$

where the constant p_{inside} strongly penalizes particles with points P_i situated inside of an obstacle (d_m is therefore sum of the robot's radius and radius of the obstacle) and δ is computed by

$$\delta = \min_{o \in O} \min_{P_i \in p} ||o - P_i||, \tag{13}$$

where O is the set of all obstacles in the workspace of the robot. The part f_{length} in both fitness functions corresponds to the length of the path which in 2D case can be computed by

$$f_{length} = \int_0^1 \sqrt{(g_x'(t))^2 + (g_y'(t))^2} dt. \tag{14}$$

The component $f_{collisions}$ penalizes the path close to an obstacle and it is defined by equation

$$f_{collisions} = \begin{cases} d^{-2}, & \text{if} \quad d_m < d \\ d^{-2} + p_{collision}, & \text{else} \end{cases} \tag{15}$$

where $p_{collision}$ penalizes paths with a collision. The penalizations should be in a true relation $p_{inside} >> p_{collision}$, because collisions far away from the control points can be corrected in the lower levels of planning. Parameter that denotes minimal distance of the path to the closest obstacle is computed by

$$d = \min_{o \in O} \min_{t \in <0;1>} ||g(t) - o)||. \tag{16}$$

Constants α and β in equations 10, 11 determine the influence of the obstacles on the final path.

3 RESULTS

The presented algorithm was intensively tested in two experiments. During the first one the parameters of the Particle Swarm Optimization were adjusted until an optimal setting was found. In the second experiment the approach was compared with the simple PSO path planning method that was published in (Hess et al., 2006).

The scenario that was used in both experiments was motivated by landscapes after a disaster. In such workspace there are several big objects (factories, buildings, etc.) with a high density of obstacles (wreckage) nearby. The rest of the space is usually almost free. In our testing situation we randomly generated 20 of such clusters with an uniform distribution. Around each center of the cluster 100 obstacles were positioned randomly followed by the uniform distribution of another 1000 obstacles in the complete workspace.

For the statistic experiments a set of 1000 randomly generated situations was generated. We chose a quadratic workspace with a side length of 1000 meters and circular obstacles with a radius of 4 meters.

Table 1: Amount of paths that intersect with an obstacle. The size of the test set is 1000 situations where the hierarchical algorithm was interrupted in the third level.

	V_{max}					
w_{st}	0.01	0.1	1	10	30	100
0.2	171	170	177	177	214	537
0.5	179	190	180	169	218	479
1	401	350	239	170	210	450
2	876	882	652	195	206	404
5	928	917	877	229	202	350

The prepared scenario was modified in order to get significant and transparent results: The radius of the obstacles was dilated by the robot radius and therefore the path outside the obstacles is considered as collision free. All obstacles around the robot's starting and goal position, that were closer than 10 times the robot' radius were removed from the scenario. Because the situations where these points lie in a cluster are unsolvable.

In the first experiment the influence of the parameters w_{start} and V_{max} on the optimization process was studied. All results were obtained with the following PSO constants: $w_{end} = 0.2$, $\varphi_1 = 2$, $\varphi_2 = 2$. The population was set to 30 particles and the computation was stopped after 30 iterations. The resulting trajectories were composed of 27 splines, because each vector p used in the optimization process corresponds to three splines and the hierarchical algorithm was interrupted in the third level as it is illustrated in Fig. 2. The interruption of the algorithm at a fixed time (even if no collision free path was found) is necessary for a reasonable comparison of the different parameter settings.

The amount of final paths with one or more collisions obtained by the experiments with different settings of w_{start} and c_V (equations (1) and (3)) are presented in table 1. More than 84 percents of the situations were solved flawlessly by the algorithm with the optimal values of the constants ($w_{start} = 0.5$ and $c_V = 3$) already in the third level of the hierarchical process. Looking at the table it can be seen that the fault increases with a deflection of both parameters from this optima. Too big values of c_V (resulting in small values of the maximum particle velocity) block the particles to search a sufficient part of the workspace and contrariwise small values of c_V change the PSO process to a random searching. Likewise too big values of the maximum inertia w_{start} facilitate the particles to leave the swarm and the searching process is again a random walk. Finally too small values of w_{start} push the particles to the best local solution and the population diversity is lost too early.

Table 2 depicts the mean fitness values and the corresponding covariances of the collision free paths

Table 2: Mean value and covariance of the best fitness values after 30 iterations.

	V_{max}			
w_{st}	0.1	1	10	30
0.2	1.40(1.2)	1.36(0.7)	1.26(0.2)	1.32(1.3)
0.5	1.42(1.5)	1.39(1.6)	1.25(0.1)	1.30(0.6)
1	1.82(6.2)	1.59(2.5)	1.27(0.2)	1.29(0.4)
2	2.81(23)	2.47(16)	1.30(0.3)	1.28(0.6)

Table 3: Number of paths which intersect with an obstacle found by the hierarchical and the simple PSO algorithm.

maximum level	I	II	III	IV	V
iterations	30	117	272	568	1103
hierarchical	923	490	159	87	65
simple($n = 2$)	809	592	481	457	411
simple($n = 3$)	923	634	505	443	405
simple($n = 4$)	992	821	645	599	526

obtained in the experiment described above. These numbers provide a comparison of the mean quality of the paths for different combinations of PSO settings within the meaning of the required features. The optimal value of w_{start} is the same as in table 1, but the optimal constant c_V is bigger than before. The stricter restriction of the maximum velocity forces the particles to search closer to the best member, that may not be collision free at the moment. The final path then can be smoother and shorter, but the ability to overcome this local minima and to find a collision free path is lower.

An example of a randomly generated situation with a solution found by the algorithm using the optimal parameters is shown in Fig. 4. Looking at Fig. 5 and 6, the decreasing amount of collisions in the path at different hierarchical levels can be observed. The solution found in the first step (dotted line) contains 21 collisions along the whole path, but the control points are situated far away from the obstacles and therefore the big clusters are avoided. Only 4 collisions are contained in the corrected path at the second level (dashed line) and the final solution (solid line) after three hierarchical optimizations is collision free.

In the second experiment the usage of the hierarchical method better agrees with a real application. The stoping criterion now also considers the number of collisions in the appropriate spline in addition to the preset maximum hierarchical level. Collision free parts of the solution do not need to be corrected in the lower levels and thus the total number of iterations is reduced. The maximum level as a part of the stopping criterion is still important, because a collision free path may not exist and also a different fitness function (equation 10) is applied in the lowest level.

Table 3 shows a comparison between the hierarchi-

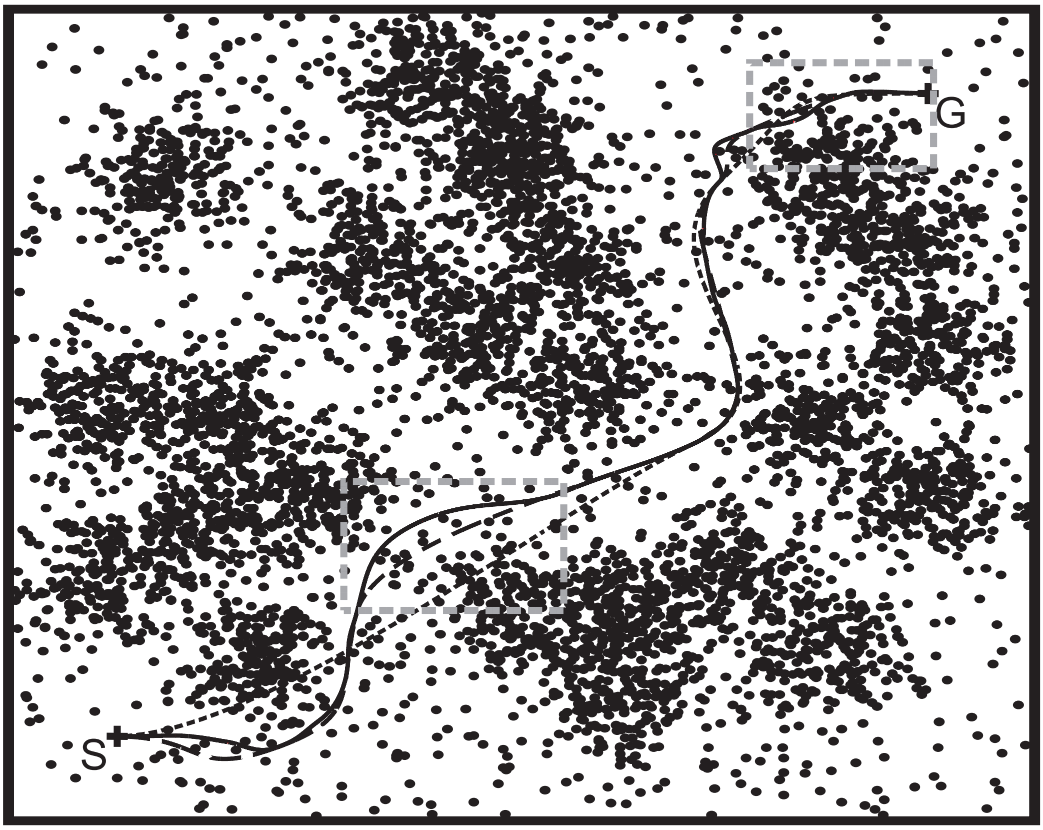


Figure 4: Situation with 3000 randomly generated obstacles. Dotted line - path found in the first level, dashed line - path found in the second level, solid line - final solution found in the third level.

cal and the simple methods. The second row of the table consists of the mean amount of iterations executed by the PSO modules in the hierarchical algorithm with different settings of the maximum level. These values acted as inputs for the relevant runs of the simple PSO algorithm that was run with different numbers of splines (other parameters were set according to (Saska et al., 2006b)).

The number of solutions that intersect with an obstacle are displayed in the last four rows of the table. Already in the third level, the presented hierarchical approach achieved a result which is several times better then the other algorithms. The computation reduction could be obtained, because the number of splines was increased depending on the environment's complexity, but the size of the searched space remained unchanged. The simple PSO method applied with a number of populations lower than 300 obtained the best results by using the shortest particle (only two splines in the string). If the evolution process was stopped after 1000 iterations, the optimal path was composed of three splines. We suppose that the optimal length increases with the duration of the evolution, because a more complicated path allows to find a collision free solution in the complex environment. But due to the extension of the calculation time it is not applicable in real time applications.

4 CONCLUSIONS AND FUTURE WORK

In this paper a novel hierarchical path planning method for complex environments was presented. The developed algorithm is based on the optimization of cubic splines connected to smooth paths. The final solution is optimized by repeatedly used PSO for collision parts of the path. The approach was intensively verified in two experiments with 1000 random generated situations. Each scenario, that was inspired by a real environment, contained more than 2000 obstacles.

The results of the experiments proved the ability of the algorithm to solve the path planning task in very complicated maps. Most of the testing situations were solved during the first five hierarchical levels and therefore the first final path of the robot was designed by only 5 runs of the PSO module (maximum 121 runs are necessary for the whole path). It makes the algorithm usable in real time applications, because the mission can be started already after 4 percent of the total computational time and the rest of the solution will be found during the robot's movement.

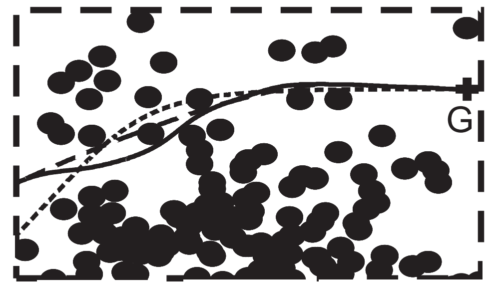


Figure 5: Zoomed part of figure 4 close to the goal position.

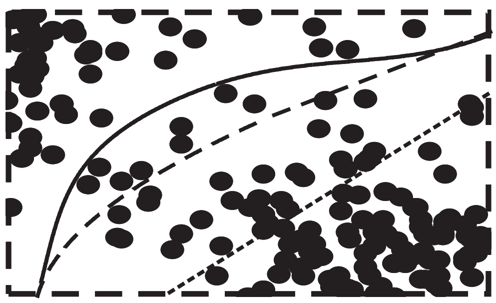


Figure 6: Zoomed part of figure 4.

The optimal adjusted approach failed in the 6.5 percent of the runs. In approximately 80 percent of these unresolved situations a collision free solution did not exist (due to unsuitable position of the random generated clusters) or could not be found during only five hierarchical levels (part of the workspace was too complicated). But error in 20 percent of the unresolved situations (about one percent of the total runs) was caused by a placing of the control points near an obstacle, in spite of the penalization p_{inside} in the fitness function 11. This problem can be solved using constrained optimization methods (Parsopoulos and Vrahatis, 2006),(Vaz and Fernandes, 2006), that guarantee fulfillment of the constrictions.

In future we would like to extend the algorithm to be utilizable in dynamical environments. Only the relevant part of the solution could be corrected due to the decomposition of the total path. Other computational reduction could be achieved by using the old cultivated population for the optimization of a partly changed fitness function.

Also the stoping rules for the PSO module or for the hierarchical sinking need to be improved. The criteria should be combined with physical features of the solution, because the fixed number of iterations and levels must be determined for the specific situation. Therewithal a superior solution can be obtained faster by a more sophisticated setting of the PSO parameters. Experiments showed that using the same values of the constants in all levels is not the best solution which gives scope for an innovation (Beielstein et al., 2002).

Another stream will be focused on testing and comparing different optimization methods (e.g. genetic algorithms (Tu et al., 2003), simulated annealing (Janabi-Sharifi and Vinke, 1993), (Martinez-Alfaro and Gomez-Garcia, 1998) or ant colony (Liu et al., 2006)). We also plan to adapt the PSO algorithm for this special task. The main idea is to use the pleasant feature of the Ferguson splines. The control points lie on the path (opposite for example in the robotic often used Coons curves (Faigl et al., 2006)) and so it is possible to put the initial population for instance on the shortest path in the Voronoi diagram (Aurenhammer and Klein, 2000) or in the Transformed net (Saska et al., 2006a). This innovation reduces the computational time, because the exploratory mode of the particle swarm optimization can be skipped.

ACKNOWLEDGEMENTS

This work was supported by the Elitenetwork of Bavaria through the program ”Identification, Optimization and Control with Applications in Modern Technologies”.

REFERENCES

Aurenhammer, F. and Klein, R. (2000). *Voronoi diagrams. Hand book of Computational Geometry*. Elsevier Science Publishers, Amsterodam.

Azariadis, P. and Aspragathos, N. (2005). Obstacle representation by bump-surface for optimal motion-planning. *Journal of Robotics and Autonomous Systems*, 51/2-3:129–150.

Beielstein, T., Parsopoulos, K. E., and Vrahatis, M. N. (2002). Tuning pso parameters through sensitivity analysis. In *Technical Report, Reihe Computational Intelligence CI 124/02.*, Collaborative Research Center (Sonderforschungsbereich) Department of Computer Science, University of Dortmund.

Borenstein, J. and Koren, Y. (1991). The vector field histogram: Fast obstacle avoidance for mobile robots. *IEEE Journal of Robotics and Automation*, 7(3):278–288.

Eberhart, R. C. and Kennedy, J. (1995). A new optimizer using particle swarm theory. In *Proceedings of the Sixth International Symposium on Micromachine and Human Science.*, pages 39–43, Nagoya, Japan.

Faigl, J., Klacnar, G., Matko, D., and Kulich, M. (2006). Path planning for multi-robot inspection task considering acceleration limits. In *Proceedings of the fourteenth International Electrotechnical and Computer Science Conference ERK 2005*, pages 138–141.

Franklin, W. R., Akman, V., and Verrilli, C. (1985). Voronoi diagrams with barriers and on polyhedra for minimal path planning. In *Journal The Visual Computer*, volume 1(4), pages 133–150. Springer Berlin.

Hess, M., Saska, M., and Schilling, K. (2006). Formation driving using particle swarm optimization and reactive obstacle avoidance. In *Proceedings First IFAC Workshop on Multivehicle Systems (MVS'06)*, Salvador, Brazil.

Janabi-Sharifi, F. and Vinke, D. (1993). Integration of the artificial potential field approach with simulated annealing for robot path planning. In *Proceedings of 8th IEEE International Symposium on Intelligent Control*, pages 536–41, Chicago, IL, USA.

Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. In *Proceedings International Conference on Neural Networks IEEE*, volume 4, pages 1942–1948.

Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. In *The International Journal of Robotics Research*, volume 5, pages 90–98.

Kunigahalli, R. and Russell, J. (1994). Visibility graph approach to detailed path planning in cnc concrete placement. In *Automation and Robotics in Construction XI*. Elsevier.

Latombe, J.-C. (1996). *Robot motion planning*. Kluver Academic Publishers, fourth edition.

Liu, S., Mao, L., and Yu, J. (2006). Path planning based on ant colony algorithm and distributed local navigation for multi-robot systems. In *Proceedings of IEEE International Conference on Mechatronics and Automation*, pages 1733–8, Luoyang, Henan, China.

Macas, M., Novak, D., and Lhotska, L. (2006). Particle swarm optimization for hidden markov models with application to intracranial pressure analysis. In *Biosignal 2006*.

Martinez-Alfaro, H. and Gomez-Garcia, S. (1998). Mobile robot path planning and tracking using simulated annealing and fuzzy logic control. In *Expert Systems with Applications*, volume 15, pages 421–9, Monterrey, Mexico. Elsevier Science Publishers.

Nearchou, A. (1998). Solving the inverse kinematics problem of redundant robots operating in complex environments via a modified genetic algorithm. *Journal of Mechanism and Machine Theory*, 33/3:273–292.

Parsopoulos, K. E. and Vrahatis, M. N. (2006). Particle swarm optimization method for constrained optimization problems. In *Proceedings of the Euro-International Symposium on Computational Intelligence*.

Saska, M., Kulich, M., Klančar, G., and Faigl, J. (2006a). Transformed net - collision avoidance algorithm for robotic soccer. In *Proceedings 5th MATHMOD Vienna- 5th Vienna Symposium on Mathematical Modelling.* Vienna: ARGESIM.

Saska, M., Macas, M., Preucil, L., and Lhotska, L. (2006b). Robot path planning using partical swarm optimization of ferguson splines. In *Proc. of the 11th IEEE International Conference on Emerging Technologies and Factory Automation. ETFA 2006.*

Tu, J., Yang, S., Elbs, M., and Hampel, S. (2003). Genetic algorithm based path planning for a mobile robot. In *Proc. of the 2003 IEEE Intern. Conference on Robotics and Automation*, pages 1221–1226.

Vaz, A. I. F. and Fernandes, E. M. G. P. (2006). Optimization of nonlinear constrained particle swarm. In *Research Journal of Vilnius Gediminas Technical University*, volume 12(1), pages 30–36. Vilnius: Technika.

Ye, J. and Qu, R. (1999). Fairing of parametric cubic splines. In *Mathematical and Computer Modelling*, volume 30, pages 121–31. Elseviers.

A COMPUTATIONALLY EFFICIENT GUIDANCE SYSTEM FOR A SMALL UAV

Guillaume Ducard and Hans P. Geering
ETH Zurich, Measurement and Control Laboratory, 8092 Zurich, Switzerland
ducard@imrt.mavt.ethz.ch, geering@imrt.mavt.ethz.ch

Keywords: Efficient path planning, adaptive guidance algorithm, unmanned aircraft, obstacle avoidance.

Abstract: In this paper, a computationally efficient guidance algorithm has been designed for a small aerial vehicle. Preflight path planning only consists of storing a few waypoints guiding the aircraft to its target. The paper presents an efficient way to model no-fly zones and to generate a path in real-time in order to avoid the obstacles, even in the event of wind disturbance.

1 INTRODUCTION

Motion planning has been extensively studied over the last decade, especially in the context of ground robots. Path-planning methods based on potential field functions present the difficulty of choosing an appropriate potential function, and the algorithm may be stuck at some local minima (Koren and Borenstein, 1991). The Probabilistic Road Maps (PRM) method (Kavraki et al., 1996) explores all the possible paths within the configuration space, and finally selects the lower cost route. However, the computational load makes the PRM method impractical for real time path planning in small UAVs. An extension to the PRM method is presented in (Amin et al., 2006) and is called modified Rapidly-exploring Random Trees (RRT), which is capable of efficiently searching for feasible paths in the space taking into account constraints from the vehicle performance. However, efforts are still going on to implement a path replanning on-the-fly as pop-up obstacles are discovered, or when the performance of the vehicle degrade. Other path-planning techniques are based on optimization methods, such as Mixed Integer Linear Programming (Schouwenaars et al., 2005) or Model Predictive Control techniques (Kuwata et al., 2006), which still involve heavy computations.

This paper presents a guidance algorithm for an unmanned aerial vehicle (UAV), which generates online a flight path based on predefined waypoints, avoids known or appearing obstacles, is simple to implement and requires very low computational power. The complete guidance system is intended to run on small microcontrollers with limited floating point operations capability.

Most of the research dealing with obstacle avoidance seems to be directed towards advanced, relatively complex methods. These methods, mainly based on optimization algorithms, are appropriate for larger UAVs with sufficient processing power onboard or for systems where the data processing can be done at a base station with the flight path being relayed up to the aircraft. The work of this paper focuses on highly simplified methods for real-time path generation in order to avoid no-fly zone (NFZ).

Section II of this paper describes how the aircraft autonomously detects whether any approaching NFZs are a threat. Section III presents a strategy to avoid the NFZ, and Section IV considers cases of wind disturbances, and shows how the guidance algorithm still allows the aircraft to avoid the NFZ.

2 GUIDANCE CONTROL LAW

The control law used in the guidance algorithm is based on work done in (Park, 2004) and (Park et al., 2004). The control law chooses a reference point that is on the desired path and a distance L_1 ahead of the aircraft. It then calculates the angle between the aircraft's velocity vector and the line L_1 to generate a lateral acceleration command a_s using (1), which is converted into a bank angle command ϕ_{com} using (2). This control law is especially suited to follow curved paths, such as circles, and is also efficient to track

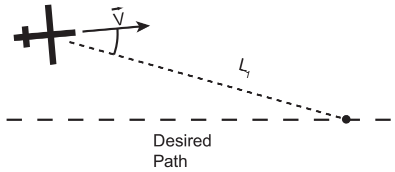


Figure 1: Diagram of Control Law Geometry.

straight line segments.

$$a_s = \frac{2V^2}{L_1} \sin\eta \tag{1}$$

$$\phi_{com} \approx \frac{a_s}{g} \tag{2}$$

3 NO-FLY ZONES

3.1 Definition of a No-Fly Zone

A no-fly zone is any airspace that an aircraft is not permitted to fly in. This airspace can be of any arbitrary shape. In order to simplify the guidance algorithm, two conditions are imposed on how the NFZ is represented.

First, the vertical limits of the NFZ are not considered so that the NFZ is essentially a two-dimensional surface. The aircraft is not allowed to pass over or under the NFZ.

Second, the shape of the NFZ is chosen to be a circle. In this way, the avoidance maneuver can be an arc of a circle in order to benefit from the guidance control law especially suited to track circles, which are described by only two parameters, their center and their radius.

Although this paper only discusses the avoidance of one circular NFZ, the algorithm can be extended to multiple no-fly zones with some simple modifications. Also, a complex no-fly zone shape can be represented by multiple circles.

Before the flight, the location of the known no-fly zones to be encountered during the mission are stored in the memory of the autopilot. If the UAV is equipped with scanning sensors that can detect pop-up obstacles, their position can be taken into account by the path-planning system to recompute on the fly a new trajectory that avoids the threat and continues the mission as soon as possible.

In order to determine whether an NFZ or an obstacle interferes with the planned path, an imaginary "detection line" is used. It has a length R_{LA} and is located in front of the aircraft, as shown in Fig. 2.

3.2 Definition of the Look-ahead Distance R_{LA}

The distance R_{LA} defines the so-called "look-ahead distance". If any part of this detection line penetrates an NFZ or an obstacle, avoidance action is immediately taken as described in the next section.

The guidance algorithm determines whether a NFZ interferes with the planned path using current aircraft position, velocity, and aircraft performance information such as the maximum bank angle that is allowed ϕ_{max}. Although the location of an NFZ is known to the guidance algorithm, the guidance algorithm will only take action if the NFZ is an immediate obstacle.

An NFZ is considered to be an immediate obstacle if any part of it is touched by the imaginary "detection line" of length R_{LA} in front of the aircraft.

Choosing a good value for R_{LA} is important. Too large of a value will cause the guidance algorithm to take unneeded action or to take action too early, while too small of a value will not allow the aircraft enough time to maneuver away from the NFZ without penetrating it.

R_{LA} is chosen such that the aircraft will fly an arc that stays just outside the NFZ at the point of closest approach, which means that the turn was started as late as possible. R_{LA} depends on the radius of the NFZ, R_{NFZ}, the ground speed of the aircraft V, and the maximum bank angle of the aircraft ϕ_{max}.

Given these parameters, and assuming a coordinated turn, the minimum turn radius the aircraft can fly is given by

$$R_{min} = \frac{V^2}{g\tan(\phi_{max})}. \tag{3}$$

In the case of a NFZ with infinite radius, the aircraft would have to make a $90°$ turn, in which case $R_{LA,min} = R_{min}$. For any NFZ with a finite radius, the aircraft has to turn less than $90°$ to avoid it. Assuming that the path of the turning aircraft is tangent to the edge of the NFZ, a triangle can be set up as shown in Fig. 2, with vertices at the center of the NFZ, at the aircraft, and at a point R_{min} off the right wing-tip. The aircraft is at the point where it must begin its turn. $R_{LA,min}$ is then given by

$$R_{LA,min} = \sqrt{2R_{min} + R_{NFZ}}\sqrt{R_{NFZ}} - R_{NFZ}. \tag{4}$$

To obtain the final value for R_{LA}, compensation must be made for the delay needed to initiate the turn, including the time to roll to ϕ_{max}. The delay needed to initiate the turn, τ_{roll}, is compensated for by adding a representative distance to $R_{LA,min}$. The assumption is made that while the aircraft is initiating the turn

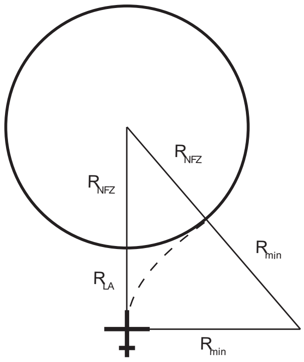


Figure 2: Diagram of R_{LA}.

it continues to fly level, and then as soon as it reaches ϕ_{max} it makes a minimum radius turn. The characteristic time τ_{roll} can be multiplied by the aircraft's speed to get the distance the aircraft will travel during this delay, which is added to $R_{LA,min}$.

The resulting look-ahead distance is

$$R_{LA} = R_{LA,min} + V\tau_{roll}. \tag{5}$$

3.3 Detection of the No-Fly Zone

As mentioned before, the algorithm monitors a line ahead of the aircraft. First, the distance D_{NFZ} from the aircraft to the center of the NFZ is calculated.

$$D_{NFZ} \leq R_{NFZ} + R_{LA} \tag{6}$$

If the condition set in (6) is satisfied, where R_{NFZ} is the radius of the NFZ, then the aircraft is considered to be within range of the NFZ. In this case, a further check is made to see if a part of the NFZ is touching the detection line.

For the second check, there are two possible cases, depending on the position of the aircraft. A pair of triangles is created as shown in Fig. 3 or Fig. 4. The edges h and R_{LA}, and the angle α are known. The length of edges y and a can easily be calculated, using

$$\begin{aligned} y &= h\sin(\alpha) \\ a &= h\cos(\alpha). \end{aligned} \tag{7}$$

Case 1 applies if $a \leq R_{LA}$. The limiting case is when edge a is tangent to the NFZ, in which case y will have a length equal to R_{NFZ}. Thus, the NFZ touches the detection line if

$$y \leq R_{NFZ}.$$

Case 2 applies if $a > R_{LA}$. The limiting case occurs when the end of the detection line is on the edge of the NFZ. This can be checked by comparing

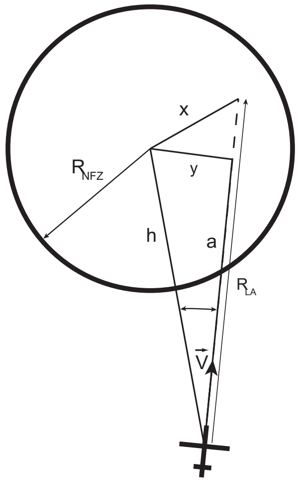


Figure 3: Diagram of NFZ Detection Algorithm, Case 1 (detected).

the length of the edge x to the radius of the NFZ, so that the detection line touches the NFZ if

$$x \leq R_{NFZ}, \tag{8}$$

where

$$x = \sqrt{y^2 + (a - R_{LA})^2}. \tag{9}$$

The check for Case 1 or Case 2 is only done if α is less than or equal to $90°$. If α is greater than $90°$, then the center of the NFZ lies behind the aircraft and no action is taken.

The no-fly zone detection method that was presented provides sufficiently early notice of any impending NFZ penetration for the guidance algorithm to take action to avoid the NFZ. The algorithm for avoiding the NFZ is described in the following section.

4 NO-FLY ZONE AVOIDANCE ALGORITHM

The no-fly zone avoidance algorithm guides the aircraft around any NFZ that the aircraft encounters. The avoidance method is designed to be simple to implement while allowing the aircraft to reach waypoints close to the edge of the no-fly zone.

4.1 Path Template

One key feature of this avoidance method is the selection of a circular arc around the NFZ as a reference path. Such a path minimizes the distance the aircraft flies to avoid the NFZ. Moreover, we saw at the beginning of this chapter that our lateral guidance control

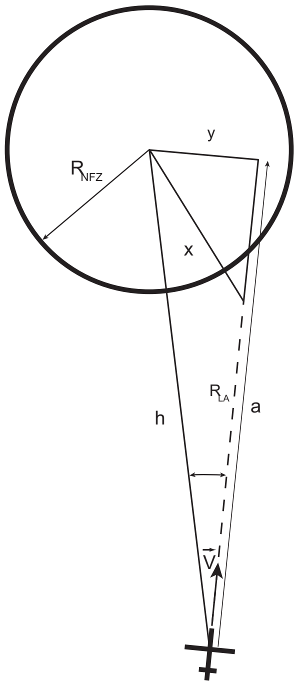


Figure 4: Diagram of NFZ Detection Algorithm, Case 2 (not detected).

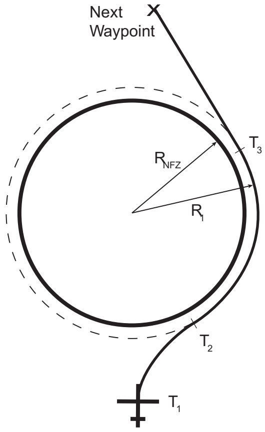


Figure 5: Avoidance Path Template.

law is particularly efficient in tracking such a path. Choosing the reference path to be circular allows the path to be easily defined in relationship to the NFZ dimensions.

The aircraft follows this path until it is able to continue towards the next waypoint in a straight line and without passing through the NFZ. As shown in Fig. 5, the arc has the same center as the NFZ but has a slightly larger radius. The distance between the reference path and the edge of the NFZ serves as a safety margin against deviations the aircraft makes from the reference path.

No complex calculations have to be made to determine where the path lies; it is defined by the center of the NFZ and a path radius, R_1, which is simply the NFZ radius plus a safety margin. R_1 must be chosen to be larger than or equal to the minimum turn radius of the aircraft, such that the reference path represents a feasible path. Also, the point at which the guidance algorithm transitions back to normal guidance towards the next waypoint is easily chosen. This transition occurs when there is a clear line-of-sight from the aircraft's current position to the next waypoint.

4.2 Relevant Control Law Properties

The properties of the chosen control law used in the guidance algorithm, namely its inherent ability to follow a circular path, make the chosen path easy to follow. As shown in (Park, 2004), the aircraft is able to follow a circle without any steady-state error, even with wind. This is because the bank angle command given by the controller causes the aircraft to fly an arc that is tangent to the aircraft's current velocity and that crosses the reference path at a given distance in front of the aircraft. In the case of a circular reference path, the proper bank angle command is given so that the aircraft flies exactly along the this reference path. When the aircraft is on the reference path and flying along it, the bank angle that is commanded provides the right lateral acceleration to fly a circle of the same radius as the reference path. When the aircraft is off the reference path, the bank angle command is such that the aircraft will converge with the reference path.

4.3 Avoidance Guidance Schedule

Upon detecting a no-fly zone, the aircraft initiates a maximum bank turn either to the left or right and then flies around the NFZ along the reference path. This method allows the guidance algorithm to initiate the avoidance maneuver as late as possible. This is desirable since it makes more waypoints reachable than if the avoidance maneuver were started earlier. The only unreachable waypoints are those that lie within a radius of $R_{NFZ} + R_{LA}$ from the center of the NFZ[1].

4.3.1 Choice of Avoidance Side, T_1

Whether the guidance algorithm chooses to go left or right around the NFZ is determined by which side of the NFZ center the aircraft is already flying towards.

[1]In the case of approaching a NFZ head-on, the guidance algorithm begins its avoidance maneuver when it reaches a distance of $R_{NFZ} + R_{LA}$.

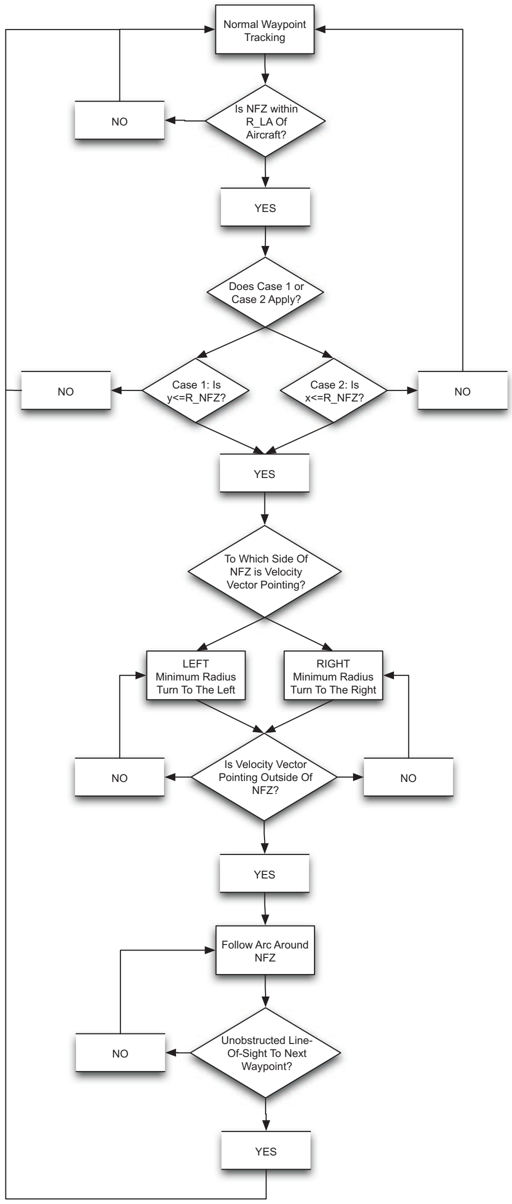


Figure 6: Finite State Diagram of Avoidance Algorithm.

If the aircraft's velocity vector is pointing to the right of the NFZ center, then the aircraft will fly around the right side of the NFZ. If the velocity vector is pointing to the left side, then the aircraft flies around the NFZ on the left side. A circular NFZ makes this decision easy.

4.3.2 Transition to Reference Path, T_2

Once the aircraft begins its turn, it continues to turn until its velocity vector is tangent to, or points outside of the NFZ. At this point the guidance switches to following the circular reference path.

4.3.3 Transition to Normal Waypoint Tracking, T_3

Once the aircraft is following the reference path, it will continue to do so until it has a line-of-sight to the next waypoint that is unobstructed by the NFZ. At this "switchover point", the guidance switches out of avoidance mode and guides the aircraft to the next waypoint. It follows a reference line from the switchover point to the next waypoint. Once the next waypoint is reached, guidance continues as normal. A possible alternative solution is to avoid the obstacle and then follow again the original reference path that passed through the no-fly zone, but this makes the aircraft fly a longer path to finally reach the desired waypoint.

4.4 Properties of the Guidance Schedule

The presented guidance schedule has several desirable properties. It attempts to minimize the number of waypoints that are unreachable, it avoids complex logic to decide how to avoid the no-fly zone, and it minimizes the time and distance to return to the original flight path.

4.4.1 Minimizing Unreachable Waypoints

The guidance schedule minimizes the number of unreachable waypoints by initiating the avoidance maneuver as late as possible. A waypoint is deemed unreachable if it cannot be flown over while following the original path and without causing the aircraft to penetrate the NFZ. Waypoints within R_1 of the center of the NFZ are unreachable.

4.4.2 Avoiding Complex Logic

The guidance schedule avoids complex logic. The main decisions that have to be made are when to begin the avoidance maneuver, which side to fly around the NFZ, and when to begin flying directly to the next waypoint. The first decision is made by the NFZ detection algorithm; the avoidance maneuver begins as soon as the NFZ is detected, which is when the aircraft is within a distance of R_{LA} of the NFZ edge. The side around which the NFZ is circumnavigated is chosen simply by the side to which the velocity vector of the aircraft points at the time the decision is made. In the case of the aircraft approaching the NFZ head-on, the decision can be made arbitrarily. The final decision is also simple, in that the aircraft continues on to the next waypoint when it has a clear line-of-sight to it. A clear line-of-sight can be checked by using an algorithm similar to the NFZ detection algorithm, but with the "detection line" pointed towards the next waypoint, instead of ahead of the aircraft.

4.4.3 Minimizing Time and Distance to Return to Original Flight Path

After the avoidance maneuver is initiated, the goal of the guidance algorithm is to minimize the distance and time to return to the original flight plan. It does this by flying directly to the next waypoint after the NFZ as soon as is safely possible. A possible downside of this is that it may create an excessively sharp turn leaving the waypoint, but the control law is able to handle even sharp turns (with an overshoot that most control laws would have).

5 SIMULATION

5.1 Simulation Setup

Simulations were done on a nonlinear 6-DOF computer model of a radio controlled aerobatic aircraft. The model has a 4-axis low-level autopilot which allows to directly give the autopilot a bank angle command. The airspeed, altitude, and side-slip are kept constant.

5.2 Simulation Scenario

Three similar scenarios were simulated with the results presented below. In all scenarios, the aircraft is following a desired path that passes through a no-fly zone. The simulation was done with maximum banks angles of $\phi_{max} = 30°$.

5.3 Simulation Results

5.3.1 No Wind

This first scenario, shown in Fig. 7, highlights the basic response of the aircraft to a NFZ blocking its path. The aircraft begins south of the NFZ and flies north along the desired path defined by the waypoints 1 to 5 and returns back to the runway. The desired path passes through a no-fly zone, but the aircraft deviates around it before returning to the desired path. The simulation was run at three different flight speeds, 15, 30, and 45 m/s. It can be seen that the aircraft begins its turn much later when flying at 15 m/s than when

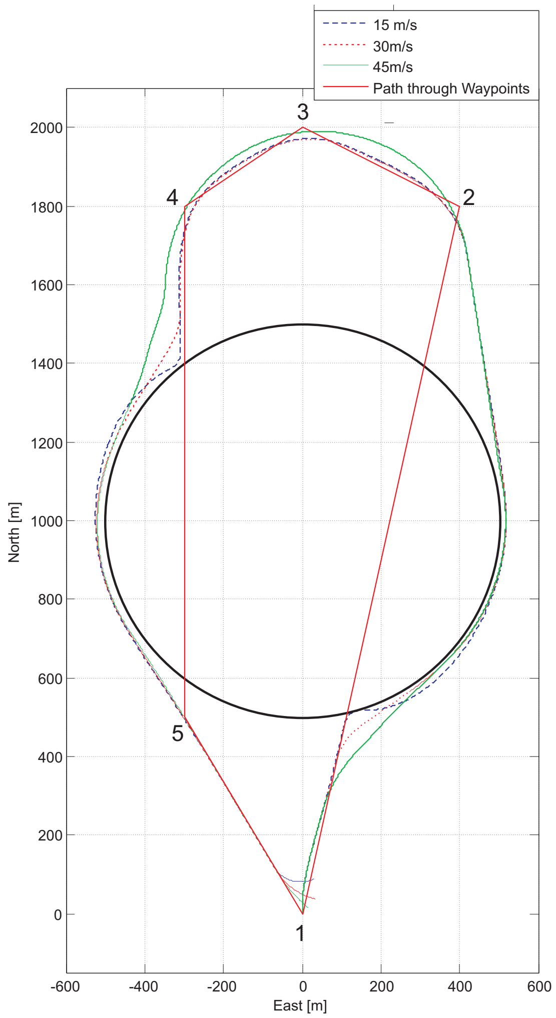


Figure 7: Obstacle avoidance in no wind condition at different speeds.

flying at 45 m/s. The airplane stays outside the no-fly zone at all three speeds.

5.3.2 With Wind

This second scenario, shown in Fig. 8, highlights the response of the aircraft in wind conditions. The desired path remains the same as in the first scenario. The path taken by the aircraft without wind and with wind are shown for comparison. The aircraft is flying at a nominal airspeed of 30 m/s.

A first flight is made with a 6 m/s crosswind blowing from east to west. In this case, the path followed by the aircraft is almost identical to the one without wind.

Another flight simulation is made with wind blowing from south to north with a speed of 6 m/s. The

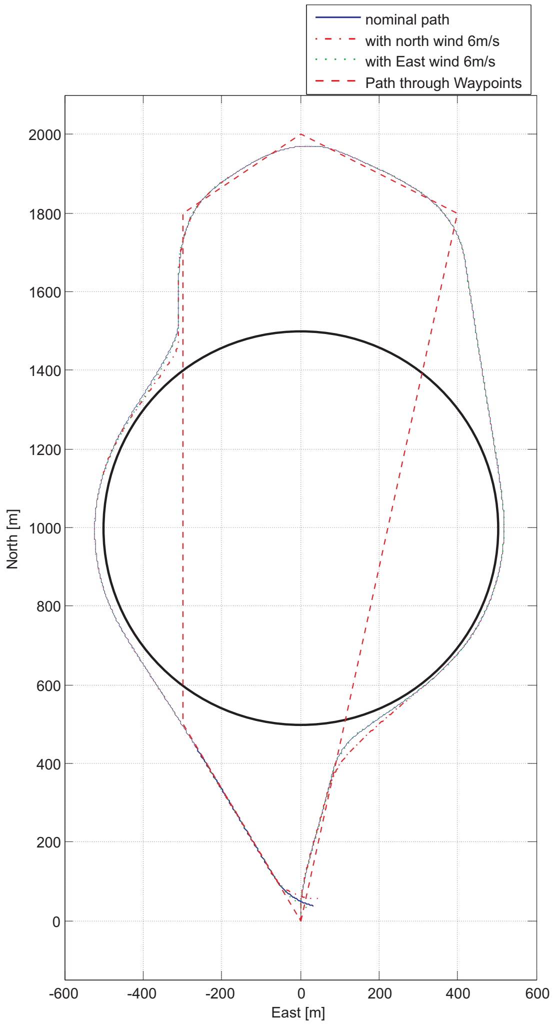


Figure 8: Obstacle avoidance in wind conditions.

trajectory in the latter windy condition differs from the the nominal track (without wind) in the two turns that avoid the obstacle, where there is a maximum difference of $20\ m$.

In both cases the no-fly zone is avoided. After the obstacle has been avoided, the guidance system resumes normal waypoint tracking.

6 CONCLUSIONS

This paper presented a guidance algorithm that combines simplicity and the ability to avoid no-fly zone. The algorithm successfully demonstrated in simulation its ability to guide the aircraft around the no-fly zone and then to resume flying along the desired path. It demonstrated this ability in wind conditions. Finally, the method is computationally efficient.

REFERENCES

Amin, J. N., Boskovic, J. D., and Mehra, R. K. (2006). A fast and efficient approach to path planning for unmanned vehicles. In *Proceedings of AIAA Guidance, Navigation, and Control Conference*, Keystone, Colorado.

Kavraki, L., Svestka, P., Latombe, J., and Overmars, M. (1996). Probabilistic roadmaps for path planning in high-dimensionnal configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4).

Koren, Y. and Borenstein, J. (1991). Potential fields methods and their inherent limitations for mobile robot navigation. In *Proceedings of IEEE Conference on Robotics and Automation*, Sacramento, CA.

Kuwata, Y., Richards, A., Schouwenaars, T., and How, J. (2006). Decentralized robust receding horizon control for multi-vehicle guidance. In *Proceedings of IEEE American Control Conference*, pages 2047–2052, Minneapolis, Minnesota.

Park, S. (2004). *Avionics and Control System Development for Mid-Air Rendez-vous of Two Unmanned Aerial Vehicles*. Ph.D. thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Available at http://hdl.handle.net/1721.1/16662, Cambridge, Massachusetts.

Park, S., Deyst, J., and How, J. (2004). A new nonlinear guidance logic for trajectory tracking. In *AIAA Guidance, Navigation, and Control Exhibit*, Providence, Rhode Island.

Schouwenaars, T., Valenti, M., Feron, E., and How, J. (2005). Implementation and flight test results of milp-based uav guidance. In *Proceedings of IEEE Aerospace Conference*.

AN IMPLEMENTATION OF HIGH AVAILABILITY IN NETWORKED ROBOTIC SYSTEMS

Florin Daniel Anton, Theodor Borangiu and Silvia Anton
University Politehnica of Bucharest, Dept. of Automation and Applied Informatics
313, Spl. Independentei sector 6, RO-060032, Bucharest, E-mail: anton@cimr.pub.ro

Keywords: Networked robotics, high availability, fault tolerant systems, resource monitoring, resource control.

Abstract: In today's complex enterprise environments, providing continuous service for applications is a key component of a successful robotized implementing of manufacturing. High availability (HA) is one of the components contributing to continuous service provision for applications, by masking or eliminating both planned and unplanned systems and application downtime. This is achieved through the elimination of hardware and software single points of failure (SPOF). A high availability solution will ensure that the failure of any component of the solution - either hardware, software or system management, will not cause the application and its data to become permanently unavailable. High availability solutions should eliminate single points of failure through appropriate design, planning, hardware selection, software configuring, application control, carefully environment control and change management discipline. In short, one can define high availability as the process of ensuring an application is available for use by duplicating and/or sharing hardware resources managed by a specialized software component. A high availability solution in robotized manufacturing provides automated failure detection, diagnosis, application recovery, and node (robot controller) re integration. The paper discusses the implementing of a high availability solution in a robotized manufacturing line.

1 HIGH AVAILABILITY VERSUS FAULT TOLERANCE

Based on the response time and response action to system detected failures, clusters and systems can be generally classified as:

- Fault-tolerant
- High availability

1.1 Fault-tolerant Systems

The systems provided with *fault tolerance* are designed to operate virtually without interruption, regardless of the failure that may occur (except perhaps for a complete site going down due to a natural disaster). In such systems all components are at least duplicated for both software and hardware.

This means that all components, CPUs, memory, Ethernet cards, serial lines and disks have a special design and provide continuous service, even if one sub-component fails. Only special software solutions will run on fault tolerant hardware.

Such systems are very expensive and extremely specialized. Implementing a fault tolerant solution requires a lot of effort and a high degree of customization for all system components.

For environments where *no* downtime is acceptable (life critical systems), fault-tolerant equipment and solutions are required.

1.2 High Availability Systems

The systems configured for *high availability* are a combination of hardware and software components configured to work together to ensure automated recovery in case of failure with a minimal acceptable downtime.

In such industrial systems, the software involved detects problems in the robotized environment (production line, flexible manufacturing cell), and manages application survivability by restarting it on the same or on another available robot controller.

Thus, it is very important to eliminate all single points of failure in the manufacturing environment. For example, if a robot controller has only one network interface (connection), a second network interface (connection) should be provided in the same node to take over in case the primary interface providing the service fails.

Another important issue is to protect the data by mirroring and placing it on shared disk areas accessible from any machine in the cluster, directly or using the local area network.

2 HIGH AVAILABILITY TERMS AND CONCEPTS

For the purpose of designing and implementing a high-availability solution for networked robotic stations integrated in a manufacturing environment, the following terminology and concepts are introduced:

RMC: The Resource Monitoring and Control (RMC) is a function giving one the ability to monitor the state of system resources and respond when predefined thresholds are crossed, so that many routine tasks can be automatically performed.

Cluster: Loosely-coupled collection of independent systems (nodes – in this case robot controllers) organized into a network for the purpose of sharing resources and communicating with each other. A cluster defines relationships among cooperating systems, where peer cluster nodes provide the services offered by a cluster node should that node be unable to do so.

There are two types of high availability clusters:

- Peer domain
- Managed domain

The general difference between these types of clusters is the relationship between the nodes.

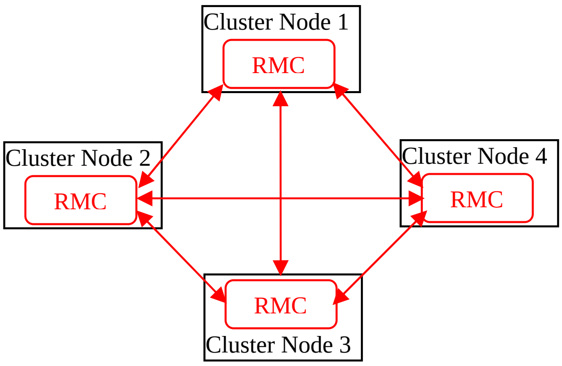


Figure 1: Peer domain cluster topology.

In a *peer domain* (Figure 1), all nodes are considered equal and any node can monitor and control (or be monitored and controlled) by any other node (Harris et. al., 2004).

In a *management domain* (Figure 2), a management node is aware of all nodes it is managing and all managed nodes are aware of their management server, but the nodes themselves know nothing about each other.

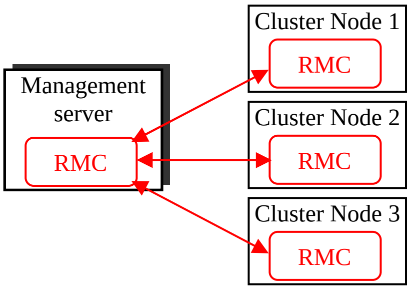


Figure 2: Managed domain cluster topology.

Node: A robot controller that is defined as part of a cluster. Each node has a collection of resources (disks, file systems, IP addresses, and applications) that can be transferred to another node in the cluster in case the node or a component fails.

Clients: A client is a system that can access the application running on the cluster nodes over a local area network. Clients run a client application that connects to the server (node) where the application runs.

Resources: Logical components or entities that are being made highly available (for example, file systems, raw devices, applications, etc.) by being moved from one node to another. All the resources that together form a highly available application or service are grouped in one resource group (RG).

Group Leader: The node with the highest IP as defined in one of the cluster networks (the first communication network available), that acts as the central repository for all topology and group data coming from the applications which monitor the state of the cluster.

SPOF: A single point of failure (SPOF) is any individual component integrated in a cluster which, in case of failure, renders the application unavailable for end users. Good design will remove single points of failure in the cluster - nodes, storage, networks. The implementation described here manages such single points of failure, as well as the resources required by the application.

The most important unit of a high availability cluster is the Resource Monitoring and Control (RMC) function, which monitors resources (selected by the user in concordance with the application) and performs actions in response to a defined condition.

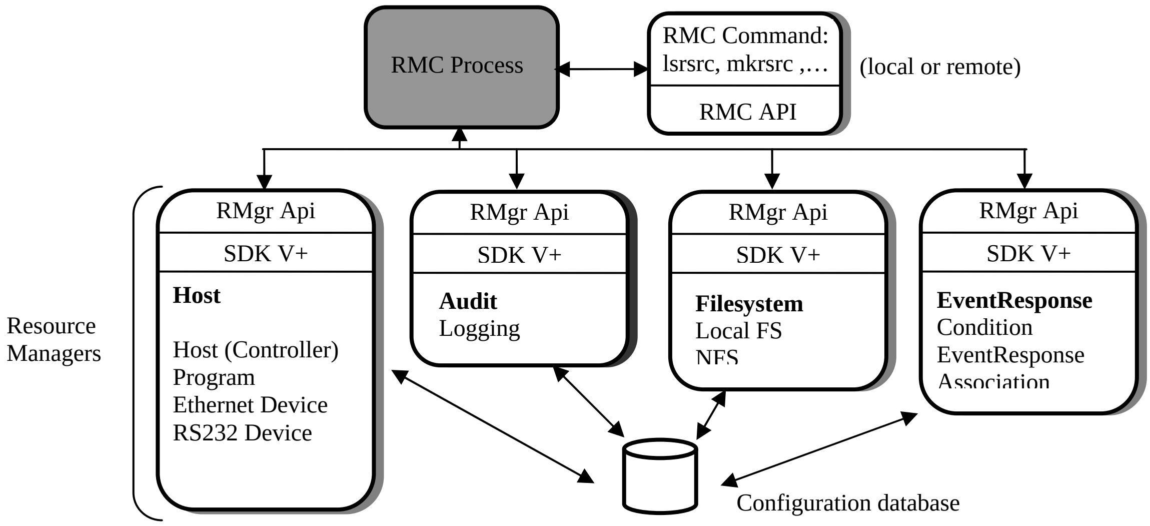


Figure 3: The structure of the RMC subsystem.

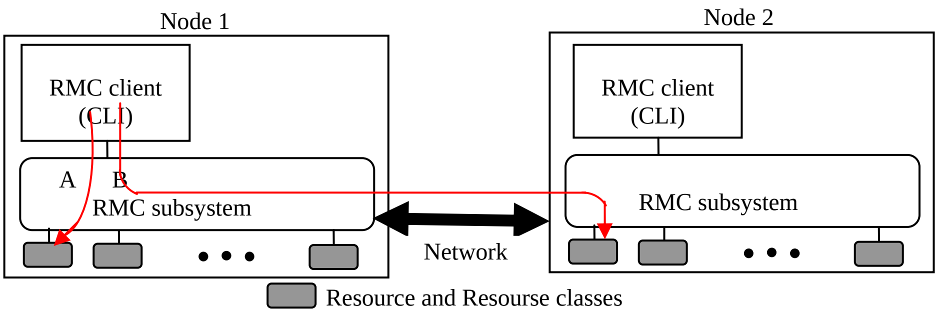


Figure 4: The relationship between RMC Clients (CLI) and RMC subsystems.

3 RMC ARCHITECTURE AND COMPONENTS DESIGN

The design of RMC architecture is presented for a multiple-resource production control system. The set of resources is represented by the command, control, communication, and operational components of networked robot controllers and robot terminals integrated in the manufacturing cell.

The RMC subsystem to be defined is a generic cluster component that provides a scalable and reliable backbone to its clients with an interface to resources.

The RMC has no knowledge of resource implementation, characteristics or features. The RMC subsystem therefore delegates to resource managers the actual execution of the actions the clients ask to perform (see Figure 3).

The RMC subsystem and RMC clients need not be in the same node; RMC provides a distributed service to its clients. The RMC clients can connect to the RMC process either locally or remotely using the RMC API i.e. Resource Monitoring and Control Application user Interface (Matsubara *et al*., 2002).

Similarly, the RMC subsystem interacting with Resource Managers need not be in the same node. If they are on different nodes, the RMC subsystem will interact with local RMC subsystems located on the same node as the resource managers; then the local RMC process will forward the requests. Each resource manager is instantiated as one process. To avoid the multiplication of processes, a resource manager can handle several resource classes.

The commands of the Command Line Interface are V+ programs (V+ is the robot programming environment); the end-user can check and use them as samples for writing his own commands.

A RMC command line client can access all the resources within a cluster locally (A) and remotely (B) located (Figure 4). The RMC command line interface is comprised of more than 50 commands (V+ programs): some components, such as the *Audit* resource manager, have only two commands, while

others, such as *Event Response* resource manager, have 15 commands.
Each resource manager is the interface between the RMC subsystem and a specific aspect of the Adept Windows operating system instance it controls. All resource managers have the same architecture and interact with the other RMC components. However, due to their specific nature, they have different usage for the end user. The resource managers are categorized into four groups:

1. *Logging and debugging* (Audit resource manager) The Audit Log resource manager is used by other RMC components to log information about their actions, errors, and so on.
1. *Configuration* (configuration resource manager). The configuration resource manager is used by the system administrator to configure the system in a Peer Domain cluster. It is not used when RMC is configured in Standalone or Management Domain nodes.
2. *Reacting to events* (Event Response resource manager). The Event Response resource manager is the only resource manager that is directly used in normal operation conditions.
4. *Data monitoring* (Host resource manager, File system resource manager). This group contains the file system resource manager and the Host resource manager. They can be seen by the end user as the containers of the objects and variables to monitor.

The Event Response resource manager (ERRM) plays the most important role to monitor systems using RMC and provides the system administrator with the ability to define a set of conditions to monitor in the various nodes of the cluster, and to define actions to take in response to these events (Lascu, 2005). The conditions are applied to dynamic properties of any resources of any resource manager in the cluster.

The Event Response resource manager provides a simple automation mechanism for implementing event driven actions. Basically, one can do the following actions:

- Define a condition composed of a resource property to be monitored and an expression that is evaluated periodically.
- Define a response that is composed of zero or several actions that consist of a command to be run and controls, such as to when and how the command is to be run.
- Associate one or more responses with a condition and activate the association.

ERRM evaluates the defined conditions which are logical expressions based on the status of resources attributes; if the conditions are true a response is executed.

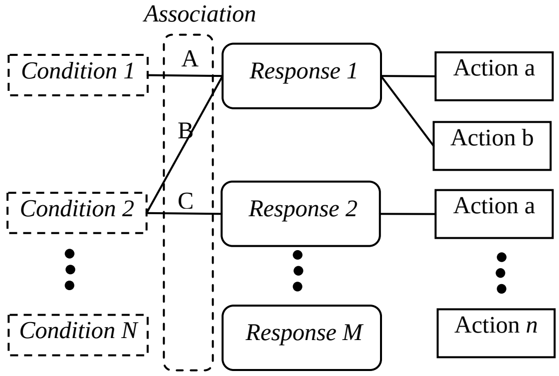


Figure 5: Conditions, responses and actions.

Conditions and responses can exist without being used and with nothing related to each other. Actions are part of responses and only defined relative to them. Although it is possible that multiple responses have an action using the same name, these actions do not refer to the same object.

To start observing the monitored resource, a condition must be associated with at least one response. You can associate a condition with multiple responses.

Figure 5 illustrates the relationship between the conditions, the responses, and the actions. In this scheme, there are three associations (A, B, and C).

The association has no name. The labels A, B, and C are for reference purposes. To refer to the specific association, you have to specify the condition name and the response name that make the association. For example, you have to specify the condition 1 and the response 1 to refer to the association A. Also, it must be clear that the same action name (in this example, action a) can be used in multiple responses, but these actions are different objects.

4 SOLUTION IMPLEMENTING FOR NETWORKED ROBOTS

In order to implement the solution on a network of robot controllers, first a shared storage is needed, which must be reached by any controller from the cluster.

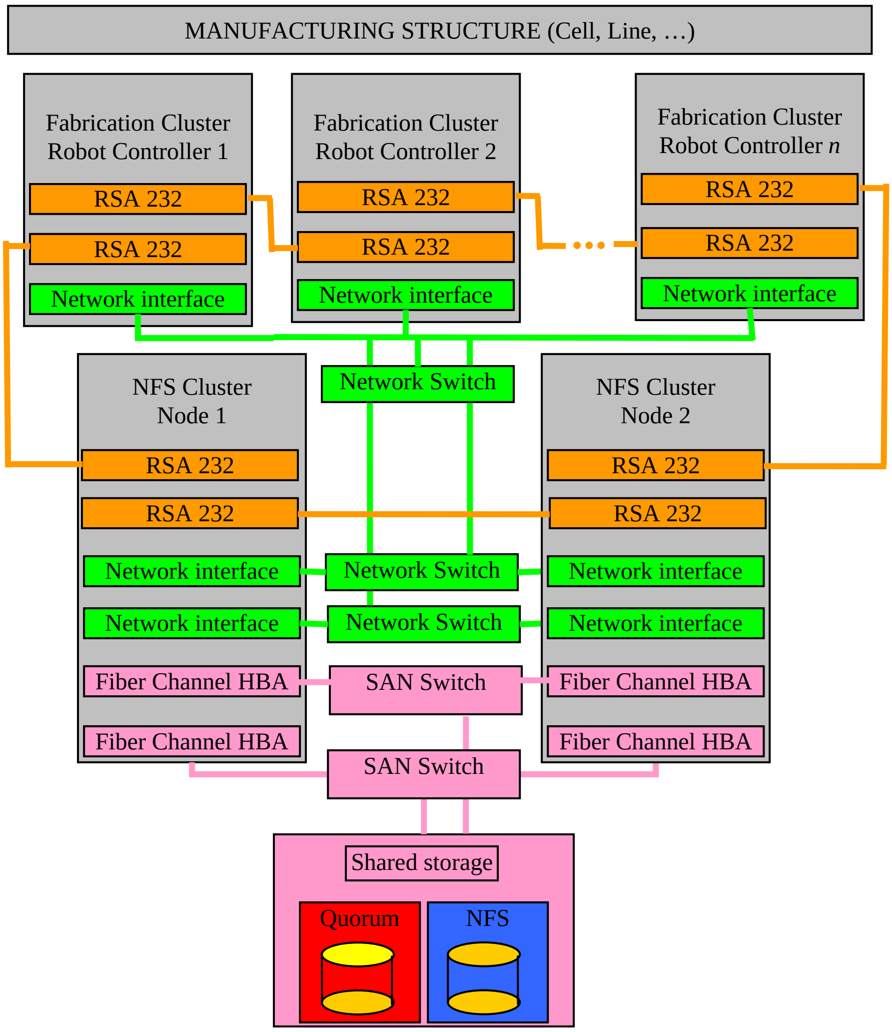


Figure 6: Implementing the high availability solution for the networked robotic system.

The file system from the storage is limited to NFS (network file system) by the operating system of the robot controllers (Adept Windows). Five Adept robot manipulators were considered, each one having its own multitasking controller.

For the proposed architecture, there is no option to use a directly connected shared storage, because Adept robot controllers do not support a Fiber Channel Host Bus Adapter (HBA). Also the storage must be high available, because it is a single point of failure for the Fabrication Cluster (FC).

Due to these constraints, the solution was to use a High Availability cluster to provide the shared storage option (NFS Cluster), and another cluster composed by Adept Controllers which will use the NFS service provided by the NFS Cluster (Figure 6).

The NFS cluster is composed by two identical IBM xSeries 345 servers (2 processors at 2.4 GHz, 1GB RAM, and 75GB Disk space, two RSA 232 lines, two Network adapters, and two Fiber Channel HBA), and a DS4100 storage. The storage contains a volume named Quorum which is used by the NFS cluster for communication between nodes, and a NFS volume which is exported by the NFS service which runs in the NFS cluster. The servers have each interface (network, serial, and HBA) duplicated to assure redundancy (Anton *et al.*, 2006; Borangiu *et al.*, 2006).

In order to detect the malfunctions of the NFS cluster, the servers send and receive status packets to ensure that the communication is established.

There are three communication routes: the first route is the Ethernet network, the second is the Quorum volume and the last communication route is the serial line. If the NFS cluster detects a malfunction of one of the nodes and if this node was the node which served the NFS service the cluster is reconfiguring as follows:

1. The server which is still running writes in the Quorum volume which is taking the functions of the NFS server, then
2. Mounts the NFS volume, then
3. Takes the IP of the other server and
4. Starts the NFS service.

In this mode the Fabrication Cluster is not aware about the problems from the NFS cluster, because the NFS file system is further available.

The Fabrication Cluster can be composed by at least two robot controllers (nodes) – *group leader* and a common node. The nodes have resources like: robot manipulators (with attributes like: collision detection, current robot position, etc...), serial lines, Ethernet adapter, variables, programs, NFS file system. The NFS file system is used to store programs, log files and status files. The programs are stored on NFS to make them available to all controllers, the log files are used to discover the causes of failure and the status files are used to know the last state of a controller.

In the event of a node failure, the production flow is interrupted. In this case, if there is a connection between the affected node and the group leader, the leader will be informed and the GL takes the necessary actions to remove the node from the cluster. The GL also reconfigures the cluster so the fabrication process will continue. For example if one node cluster fails in a three-node cluster, the operations this node was doing will be reassigned to one of the remaining nodes.

The communication paths in the multiple-robot system are: the *Ethernet network* and the *serial network*. The serial network is the last resort for communication due to the low speed and also to the fact that it uses a set of Adept controllers to reach the destination. In this case the *ring network* will be down if more than one node will fail.

5 CONCLUSIONS

The high availability solution presented in this paper is worth to be considered in environments where the production structure has the possibility to reconfigure, and where the manufacturing must assure a continuous production flow at batch level (job shop flow).

There are also some drawbacks like the need of an additional NFS cluster. The spatial layout and configuring of robots must be done such that one robot will be able to take the functions of another robot in case of failure. If this involves common workspaces, programming must be made with much care using robot synchronizations and monitoring continuously the current position of the manipulator.

The advantages of the proposed solution are that the structure provides a high availability robotized work structure with a insignificant downtime.

The solution is tested on a four-robot assembly cell located in the Robotics and IA Laboratory of the University Politehnica of Bucharest. The cell also includes a CNC milling machine and one Automatic Storage and Retrieval System, for raw material feeding and finite products storage.

During the tests the robot network has detected a number of errors (end-effector collision with parts, communication errors, power failure, etc.) The GL has evaluated the particular situation, the network was reconfigured and the abandoned applications were restarted in a time between 0.2 and 3 seconds.

The most unfavourable situation is when a robot manipulator is down; in this case the down time is greater because the application which was executed on that controller must be transferred, reconfigured and restarted on another controller. Also if the controller still runs properly it will become group leader to facilitate the job of the previous GL.

In some situations the solution could be considered as a fault tolerant system due to the fact that even if a robot controller failed, the production continued in normal conditions.

REFERENCES

Anton F., D., Borangiu, Th., Tunaru, S., Dogar, A., and S. Gheorghiu, 2006. Remote Monitoring and Control of a Robotized Fault Tolerant Workcell, *Proc. of the 12th IFAC Sympos. on Information Control Problems in Manufacturing INCOM'06*, Elsevier.

Borangiu, Th., Anton F., D., Tunaru, S., and A. Dogar, 2006. A Holonic Fault Tolerant Manufacturing Platform with Multiple Robots, Proc. of *15th Int. Workshop on Robotics in Alpe-Adria-Danube Region RAAD 2006*.

Lascu, O. et al, 2005. *Implementing High Availability Cluster Multi-Processing (HACMP) Cookbook*, IBM Int. Technical Support Organization, 1st Edition.

Harris, N., Armingaud, F., Belardi, M., Hunt, C., Lima, M., Malchisky Jr., W., Ruibal, J., R. and J. Taylor, 2004. *Linux Handbook: A guide to IBM Linux Solutions and Resources*, IBM Int. Technical Support Organization, 2nd Edition.

Matsubara, K., Blanchard, B., Nutt, P., Tokuyama, M., and T. Niijima, 2002. *A practical guide for Resource Monitoring and Control (RMC)*, IBM Int. Technical Support Organization, 1st Edition.

A MULTIROBOT SYSTEM FOR DISTRIBUTED SENSING

Janne Haverinen, Anssi Kemppainen, Janne Kivijakola and Juha Röning
Department of Electrical and Information Engineering, University of Oulu, Finland
janne.haverinen@oulu.fi, anssi.kemppainen@oulu.fi, janne.kivijakola@oulu.fi, juha.roning@oulu.fi

Keywords: Multi-robot system, distributed sensing, laser, range sensing.

Abstract: This paper presents a modular multirobot system developed for distributed sensing experiments. The multirobot system is composed of modular small size robots, which have various sensor capabilities including a color stereo camera system and an infrared based sensor for infrastructure independent relative pose estimation of robots. The paper describes the current state of the multirobot system introducing the robots, their sensor capabilites, and some initial experiments conducted with the system. The experiments include a distributed structured light based 3-D scanning experiment involving two robots, and an experiment where a group of robots arrange into a spatial formation and measures distributions of light and magnetic field of an environment. The experiments demonstrate how the proposed multirobot system can be used to extract information from the environment, and how they can cooperatively perform non trivial tasks, like 3-D scanning, which is a difficult task for a single small size robot due to limitations of current sensing technologies. The distributed 3-D scanning method introduced in this paper demonstrates how multirobot system's inherent properties, i.e. spatial distribution and mobility, can be utilized in a novel way. The experiment demonstrates how distributed measurement devices can be created, in which each robot has an unique role as a part of the device, and in which the mobility of the robots provides flexibility to the structure of the measurement system.

1 INTRODUCTION

The miniaturization of mobile robots (Floreano and Mondada, 1998; Caprari et al., 2000; Sibley et al., 2002; Colot et al., 2004; Mondada et al., 2004) is rapidly progressing due to developments in electronics and material technology, for example. Multirobot systems composed of a group of miniaturized robots can have numerous applications in the future. They may be a regular sight on space expeditions or they may operate inside the human body for our well being. At present, multirobot systems are a rear sight in real world applications. However, they have already been used in surveillance applications, military demonstrations, and in distributed sensing applications. The distributed sensing is an interesting application domain as the multirobot system is inherently spatially distributed. As this paper shows, the spatial distribution of robots can be naturally utilized to overcome problems that the miniaturization of the robots introduces to the sensing technology in some problem domains. As an example, we will show how a multirobot system can be used as a distributed structured light based scanning device for extracting 3-D information from an environment. The distributed 3-D scanning concept is useful in applications where a group of robots needs to extract objects' geometrical information in remote locations without a requirement that a single robot must carry alone the necessary technology for performing the range measurements (Lee and Song, 2005).

An other experiment, which demonstrates the feasibility of the proposed multirobot system, shows how robots can be driven into a given spatial formation, and used to automatically perform measurements about illumination and magnetic field of the environment.

This paper briefly introduces the developed multirobot system and its sensor capabilities. Then, two experiments are briefly presented in order to demonstrate how the system can be used in interesting real world applications. The multirobot system is composed of miniature mobile robots first introduced in (Haverinen et al., 2005). Each robot can have various sensor capabilities including a color stereo camera system and an infrared based sensor for relative pose estimation (Kemppainen et al., 2006). The experiments are conducted by combining these sensor capabilities (modules) to form both heterogeneous, and homogeneous teams of robots.

2 THE MULTIROBOT SYSTEM

The presented multirobot system is composed of modular miniature mobile robots (Haverinen et al., 2005). One configuration of an individual robot is shown in Fig. 1. In addition to DC-motor (actuator) and power modules, the robot in Fig.1 has four other modules, which are (from top): the IR-location, the stereo camera, the radio, the environment, and the IR-proximity modules, respectively. Each of these modules provide an well defined serial interface for reading or writing data. All modules have an 8-bit low power 8MHz MCU (ATmega32), which implements the serial interface for accessing the module services, and controls the logic of the module.

Each module can have from one to three different serial interfaces (UART,I2C,SPI). While the UART interface is mandatory for each module, I2C and SPI interfaces are optional, and they are used for enhancing the bus performance. Each module with more than one interface can be commanded to switch between interfaces in order to adapt the bus performance.

The infrared based location module (Kemppainen et al., 2006) is used to estimate the relative poses of robots without external infrastructures such as beacons, WLAN base stations, GPS or landmarks. The operation principle of the location module is based on omni directional amplitude modulated infrared transmission and an infrared receiver that utilizes a rotating beam collector for finding the direction of the infrared transmission. The detected modulation frequency at the receiver identifies the transmitting system. Each location module has an unique transmission modulation frequency, which gives identity for each robot having the location unit.

By having the module, the robot can have estimated poses of the surrounding robots within five meters without any external infrastructure. The infrared location module is utilized in the experiments where robots must be driven into specific formations prior distributed sensing procedure. The module can also be utilized to maintain and adapt formations during a task execution. The infrared location module is used in all experiments described in this paper. In the 3-D scanning experiment the location module provides the necessary information to arrange the robots into given line formation prior the scanning procedure. In distributed sensing experiments, the module is used to setup the initial measurement formation (see Fig. 6), and to maintain the formation during the measurement procedure. Fig. 2 demonstrates how the location module can be used to estimate the poses of the surrounding robot. In Fig. 2 the robot at $(0,0)$ estimates the poses of the two other robots for a short period of time. Only the (x,y)-coordinates of the located robots are shown (heading is not plotted). See (Kemppainen et al., 2006) for more detailed description of the IR-location module.

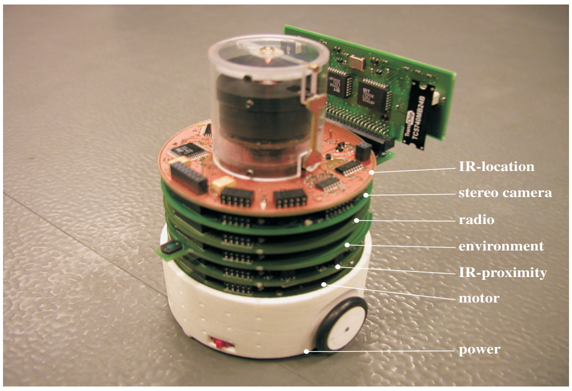


Figure 1: One configuration of the miniature robot. This robot has four modules in addition to motor and power modules. The modules include: the IR-location, the stereo camera, the radio, the environment, and the IR-proximity modules. The environment module is used to measure accelerations, temperature, ambient lighting, and the direction and magnitude of the magnetic field. The radio module implements the 868MHz 100kbits radio link for inter robot and PC communication. The purpose of the motor module is to control the two DC-motors of the robot. The power module is responsible of recharging the battery and providing the power for all modules through the module bus.

3 EXPERIMENTS

The purpose of the presented experiments is to demonstrate different strategies of using the multi-robot system for distributed sensing. The first experiment shows how a pair of heterogeneous robots can be used as unique parts of a distributed measurement device, and how system's inherent mobility can be utilized to provide a measurement device that has the necessary flexibility of adapting measurement geometry to the structure of the environment. The sec-

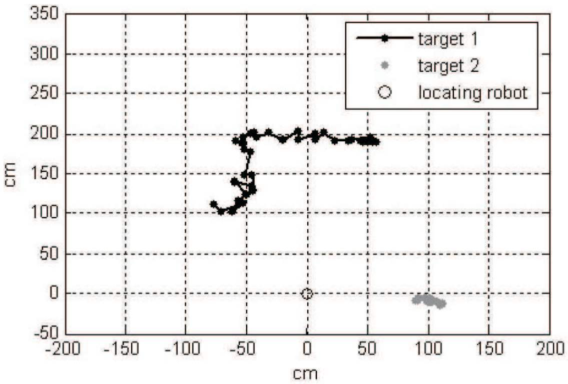


Figure 2: The infrared location system provides the pose estimates for two robots. The robot at $(0,0)$ is locating the two other robots for a short period of time.

ond experiment shows how a team of homogeneous robots can create a map of environment by moving in a given formation and by making point measurements about ambient lighting and magnetic field from locations found by dedicated infrared location modules.

3.1 Distributed 3-d Scanning

The setup of this experiment consist of two heterogeneous robots: one of the robots acts as a laser sheet projector, and the other as a laser stripe detector. The projector robot, shown in Fig. 3, has a special laser module which includes 1mW (class I) laser stripe projector. The projector provides a vertical sheet of light that is projected into the environment. The robot that is acting as a laser stripe detector uses one of the color CMOS cameras of the stereo camera module loaded with a program that extracts the projected laser patterns from the surfaces of objects in the camera's field of view.

The measurement geometry is depicted in Fig. 4. First, the projector and the detector robots are aligned into a given line formation in which the laser and the camera are pointing to the left in regard to the driving direction. Both robots have now the same heading. In order to have absolute range measurements the parameters of the measurement geometry has to be known (Haverinen and Röning, 2000). However, relative range measurements can be made without knowing the exact mutual poses of the two robots. In our experiments, we have used both machine vision technique (Heikkilä and Silven, 1997) and the location module (Kemppainen et al., 2006) to arrange the robots into the known initial measurement formation. The machine vision technique utilizes the know landmark pattern mounted on the laser robot (shown in Fig. 3) in order to estimate the mutual poses of the robots.

The measurement geometry described in Fig. 4 is not the only option. The robots can also be arranged onto a circumference with both optical axis and the laser sheet pointing to the center of the circle. By driving along the the circumference with equal velocities the robots can perform a full scan of an object that resides inside the circle. Yet another option is to keep the robots on fixed locations and rotating the projector robot: in this way the robots can scan a large area from fixed locations. However, the angular motion of the projector robot must be known in order to compute the range image.

In our experiments the scanning procedure starts after the robots are properly aligned. Both robots are moving with equal velocities. The camera robot performs the image processing in order to extract the projected laser patterns reflected from the surfaces of objects. The image coordinates of the laser stripe on the image plane are locally saved into the camera module's SDRAM. At the end of scanning, the image data is transferred through the radio link to the host computer (PC), which computes the final range image.

Fig. 5 shows examples of range images acquired by the cooperating robots. The images demonstrate how cooperating miniature robots can perform a demanding measurement task by acting as distributed parts of the physical measurement device. The cooperative scanning can have important applications: a pair of miniature mobile robots can perform a remote 3-D scanning procedure and acquire range data from selected objects in environments that might be unreachable by traditional 3-D measurement devices. The range data gives important information about an (possible hazardous) environment, objects, and their geometrical properties. The presented distributed sensing technique can provide dense range data for creating 3-D maps of unknown environments without computationally intensive stereo image processing (Rocha et al., 2005), for example. The technique can also be an alternative for creating data for vision based 3-D SLAM (Tomono, 2005).

3.2 Cooperative Measurement of Spatial Data

This experiment demonstrates how heterogeneous robots can be used to perform a simple distributed sensing task, and to create a spatial map of point measurements. The experiment also highlights the feasibility of the infrared location module in applications domains where locations of the robots must be known. The infrared location system is used to create and maintain the measurement formation, while the robots are making point measurements with the

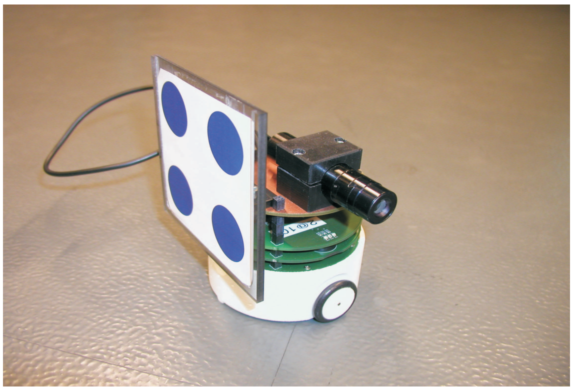


Figure 3: The laser robot. The robot consist of the 1mW laser line projector. This robot also has a landmark pattern which can be used to estimate the pose of the projector robot in the detector robot's coordinate system using the calibrate camera module and defining its extrinsic parameters in regard to the known landmark pattern (Heikkilä and Silven, 1997). Another technique to estimate the pose is to utilize the IR-location module.

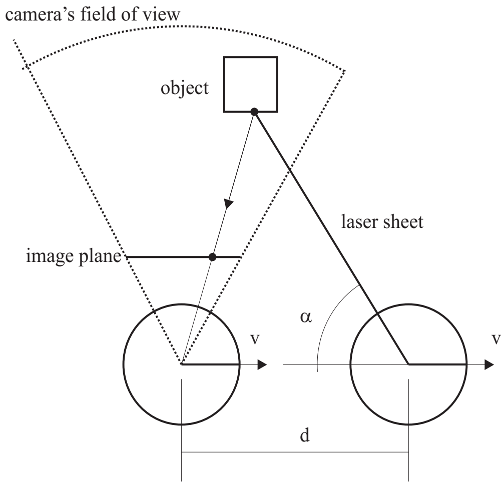


Figure 4: The measurement geometry used in the cooperative scanning experiment. The robots are first aligned into a line formation - both robots having the same heading. The distance d and the angle α control the measurement resolution and the measurement range. d and α can be adapted on the basis of the measurement task. During the scanning procedure, both robots maintain the same velocity v.

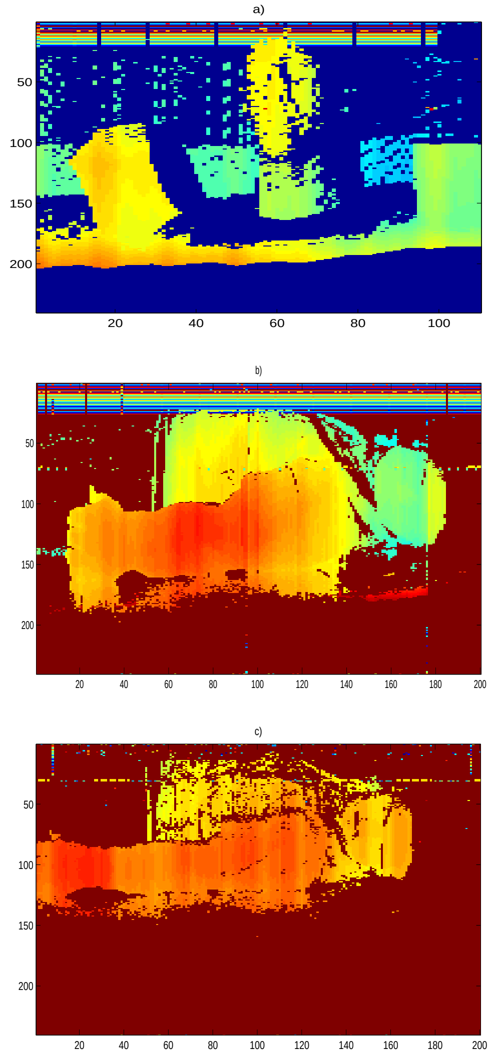


Figure 5: Range images from three cooperative 3-D scanning experiments. a) range image from arbitrary artificial objects (a head statue and some polyhedrons). b) and c) robots have scanned a person laying on the floor. These images demonstrate how, by simple cooperation, a pair of miniature mobile robots can obtain dense range data from objects of an environment.

scalar sensors of the environment module. The measurements are stored into a grid map with predefined cell size. The value of each cell is the mean value of the point measurements made in the corresponding cell.

Although the experiment is simple, it has important applications as it makes possible to automatically gather spatial scalar data from the area of interest. The data can then be used to analyze the lighting properties of the environment or to visualize how the magnetic field of the environment behaves due to metallic structures in the environment, and analyze its properties over time, for example.

Fig. 6 shows the initial arrangement of the robots prior the experiment (left): from these locations the robots are driven into the measurement formation using the infrared location module (right). While maintaining the measurement formation, the robots start moving and doing measurements. The lighting measurements are shown in Fig. 7, and the direction of the magnetic field in each cell in presented in Fig. 8. The Fig. 8 shows how the direction of the magnetic field varies in the environment due to metallic objects and structures. If this magnetic fingerprint of the environment is static it can be used to classify environments based on previous observations of the field on the same area, for example.

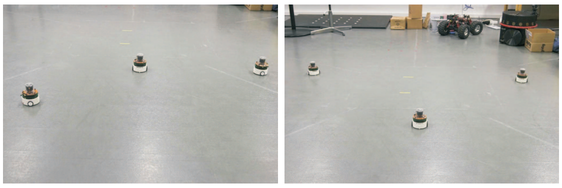


Figure 6: Left: the initial (arbitrary) positions of the robots. From initial poses the robots are driven into the measurement formation by using the IR location module. Right: The IR-location system has been utilized to drive the robots into the given measurement formation. The robot in the middle acts as a leader and the other two robots as followers.

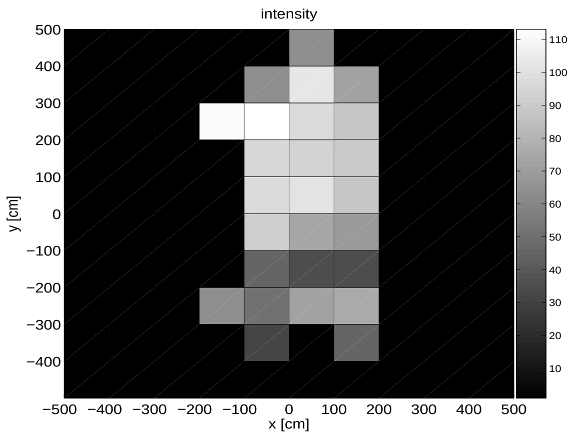


Figure 7: Measured ambient light intensities. The figure shows the value, given by the ambient light sensor, for each cell of the grid.

4 CONCLUSION

This paper presented a multirobot system developed for distributed sensing experiments and applications. Each robot has variety of sensors for measuring properties of an environment. The sensors include a temperature sensor, an ambient light detector, a compass module, accelerometers, and a color stereo camera system.

Distributed measurement tasks are naturally suited for multirobot system as they are inherently

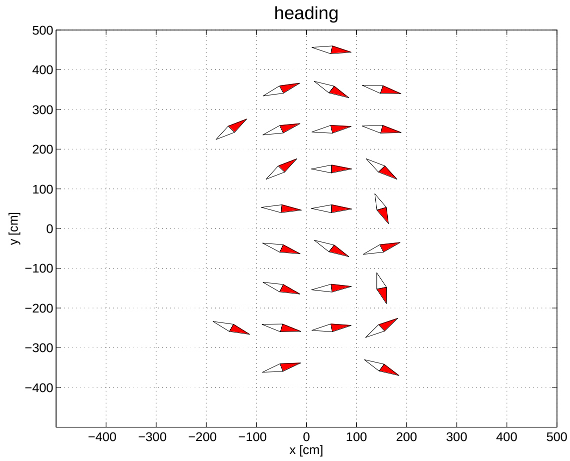


Figure 8: Measured directions of the magnetic field. The figure shows how magnetic field varies due to metallic structures of the environment. The variation (assuming it being static over time) can be used to classify environments based on previous observations from the same environment.

spatially distributed. The 3-D scanning experiment showed how a pair of heterogeneous robots can cooperate by acting as parts of a distributed 3-D scanning device. This observation is particularly important, as this kind of measurement instrument is otherwise very difficult to implement for a single miniature mobile robot. The scanning experiment also demonstrated how the multirobot system's spatial distribution and mobility can be utilized in a novel way to create distributed measurement devices in which each robot has an role as a part of the device, and in which the mobility of the robots provides flexibility to the structure of the measurement system.

The aim of this research is to develop multirobot systems which help humans to have meaningful information from (remote) environments and its objects. This information is then used to analyze the state of the environment or to execute multirobot tasks autonomously.

Based on the presented experiments, our purpose is to go toward more demanding implementations on which a person can instruct a multirobot system to measure selected objects of the environment after which the robots take their positions and cooperatively gather information, like range data, about the objects.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the support of the Academy of Finland.

REFERENCES

Caprari, G., Arras, K., and Siegwart, R. (2000). The autonomous miniature robot alice: from prototypes to applications. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'00)*, pages 793–798.

Colot, A., Caprari, G., and Siegwart, R. (2004). Insbot: Design of an autonomous mini mobile robot able to interact with cockroaches. In *IEEE International Conference on Robotics and Automation (ICRA'2004)*, pages 2418–2423, New Orleans.

Floreano, D. and Mondada, F. (1998). Hardware solutions for evolutionary robotics. In Husbands, P. and Meyer, J.-A., editors, *First European Workshop on Evolutionary Robotics (EVOROBOT'1998*, pages 137–151, Berlin. Springer-Verlag.

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.

Haverinen, J. and Röning, J. (2000). A 3-d scanner capturing range and color for the robotics applications. In *24th Workshop of the AAPR - Applications of 3D-Imaging and Graph-based Modeling*, Villach, Carinthia, Austria.

Heikkilä, J. and Silven, O. (1997). A four-step camera calibration procedure with implicit image correction. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1106–1112, San Juan, Puerto Rico.

Kemppainen, A., Haverinen, J., and Röning, J. (2006). An infrared location system for realtive pose estimation of robots. In *16-th CISM-IFToMM Syposium of Robot Design, Dynamics, and Control (ROMANSY 2006)*, Warsaw, Poland.

Lee, S. and Song, J.-B. (2005). Mobile robot localization using range sensors: consecutive scanning and cooperative scanning. *International Journal of Control, Automation, and Systems*, 3(1):1–14.

Mondada, F., Pattinaro, G., Guignard, A., Kwee, A., Floreano, D., Deneubourg, J.-L., Nolfi, S., Gambardella, L., and Dorigo, M. (2004). Swarm-bot: a new distributed robotic concept. *Autonomous Robots, special Issue on Swarm Robotics*, 17(2-3):193–221.

Rocha, R., Dias, J., and Carvalho, A. (2005). Cooperative multi-robot systems: A study of vision-based 3-d mapping using information theory. In *IEEE International Conference on Robotics and Automation*, pages 386–391, Barcelone, Spain.

Sibley, G., Rahimi, M., and Sukhatme, G. (2002). Robomote: A tiny mobile robot platform for large-scale ad-hoc sensor networks. In *IEEE International Conference on Robotics and Automation (ICRA'2002)*, Washington DC.

Tomono, M. (2005). 3-d localization and mapping using a single camera based on structure-from-motion with automatic baseline selection. In *IEEE International Conference on Robotics and Automation*, pages 3353–3358, Barcelone, Spain.

ENTANGLEMENT DETECTION OF A SWARM OF TETHERED ROBOTS IN SEARCH AND RESCUE APPLICATIONS

T. R. Vishnu Arun Kumar and R. C. Richardson
School of Computer Science, University of Manchester, Manchester, United Kingdom
vishnuarunkumar@gmail.com, R.C.Richardson@cs.man.ac.uk

Keywords: USAR, swarm of robots, tether, entanglement detection.

Abstract: In urban search and rescue (USAR) applications, robots play a pivotal role. As USAR is time sensitive, swarm of robots is preferred over single robot for victim search. Tethered robots are widely used in USAR applications because tether provides robust data communication and power supply. The problem with using tethers in a collapsed, unstructured environment is tether entanglement. Entanglement detection becomes vital in this scenario. This paper presents a novel, low-cost approach to detect entanglement in the tether connecting two mobile robots. The proposed approach requires neither localization nor an environment map. Experimental results show that the proposed approach is effective in identifying tether entanglement.

1 INTRODUCTION

USAR is a time-critical process, which involves complex and hazardous environment. Secondary collapse, confined space, presence of fire and poisonous gas in the environment pose serious threats to the human and canine rescuers. Thus, robots become inevitable in the field of search and rescue. Mobile Robots can be used for a variety of tasks such as localization, communication, victim search, biomedical monitor delivery, environment monitoring and reconnaissance. It is desirable to deploy a swarm of robots as the survival rate of the victims falls drastically after 72 hours.

Untethered autonomous robots depend on wireless communication for information exchange. When such robots are employed simultaneously in the same area, issues such as interferences with other systems, data security and international band differences will arise as stated in (Fukushima et al.). Tethered robots are used in a variety of applications in ground, under-water and space environments (Fukushima et al.). During recent times, tethered robots are being employed in search and rescue because tethers inherently provide robust data communication and uninterrupted power delivery (Fukushima et al.), (Hert et al., 1999). Tethers can be used for navigating the robots through steep slopes (Fukushima et al.) and also for pulling the robot out when it gets stuck into debris (Perrin et al.).

When a swarm of tethered robots is employed in USAR environment, the tether might get entangled with the obstacles or fellow robots. Entanglement detection becomes an important problem that needs to be addressed. Despite this, the merits of using tethers make them a valid option even in search and rescue scenario. Traditionally there are different techniques to tackle entanglement detection. Most of them need the robot to localize using an environment map, which is then used to detect the existence of tether entanglement. Navigation planning could be done in such a way that there is no tether entanglement (Hert et al., 1999). A method to plan the shortest path for a tethered robot to a destination point has been discussed in (Xavier, 1999).

In this paper, a novel, low-cost technique to detect tether entanglement has been proposed. This technique does not need the swarm of robots to be localized. It does not require any environment map. Section 2 discusses related work on localization and map building in USAR. In Section 3, the tether entanglement detection hardware is explained. Section 4 throws light on experimental results and Section 5 deals with the characterization of tether entanglement using the experimental results. In Section 6, a static model of tether is derived and the experimental results are analyzed using the model. Section 7 and 8 discuss current work and conclusion respectively.

2 RELATED WORK

In USAR, communication and power supply play crucial role. In case of autonomous robots using wireless communication, the information exchange is always noisy because of the thick concrete and steel structures present in the search and rescue environment (Perrin et al.). Also untethered robots carry on board power supply for their operation. This limits the life time of the entire system. So tether based multi-robot systems are preferred over untethered robots in search and rescue scenario. In a tether based multi-robot system, a robot can detect tether entanglement based on its pose in the environment map and the pose of the fellow robots in the same map. This technique is often referred to as Simultaneous Localization and Map building (SLAM) (Wijesoma et al., 2004).

One of the approaches for SLAM is to use Inertial Measurement Unit (IMU) for localization and Global Positioning System (GPS) for map building. An Inertial Navigation system has errors rising from factors like bias, scale factor uncertainties, misalignment errors and noise (Sukkarieh et al., Volume 19). Also in an uneven terrain, gyrometer readings tend to be very noisy. Fault detection and fault isolation form an integral part of an IMU as stated in (Sukkarieh et al., Volume 19). In search and rescue scenario, GPS becomes futile as discussed in (Gustafon et al., 2005), (Cheng et al., 2004), and (Ramirez-Serrano et al., SSRR).

Vision based approaches, which rely on landmarks are used to localize the robot (Saeedi et al., 2003), (Ramirez-Serrano et al., SSRR). As search and rescue environment is complex and unstructured, landmark based approaches are not efficient (Cheng et al., 2004). The same reason can be applied to (Saeedi et al., 2003) in which a vision based approach for 3D localization and tracking has been proposed. In this approach, distinctive scene features extracted from the environment are used for localization, but uncertainty in perception rising due to different regions appearing similar is an issue to be addressed.

The concept of Intelligent Dynamic Landmarks is discussed in (Ramirez-Serrano et al., SSRR), wherein some members of the robot group act as portable landmarks for other robots to localize. In (Gustafon et al., 2005), a swarm of robots have been employed to achieve localization and target identification. Line of sight approach is adopted to localize heterogeneous teams of robots in (Grabowski et al., 2004).

In order to detect tether entanglement, most of the approaches need the robots to be localized and/or an environment map. In this paper, a novel tether entanglement detection technique has been proposed that eliminates the need for localization and environment map.

3 ENTANGLEMENT DETECTION HARDWARE

The essence of this approach is that by recoiling tethers and monitoring the force across the tether during this process, entanglements, snags and chafing effects on the tether can be detected. The proposed system consists of two components (i) the tether winding unit that pulls up tether slack (ii) a sensor to detect horizontal forces across the tether. The principle described here is to be applied to swarms of interlinked tethered robots.

3.1 Tether Winding Unit (TWU)

Tether Winding Unit comprises of a pair of wheels tightly coupled with a spring. One of the wheels is driven by a 6 volt-5 Watt DC Motor. The tether passes between a pair of wheels as shown in the Figure-1. This unit is mounted on one of the robots (Robot-A). There is an automatic wire coiling system on the robot, which would hold one side of the cable.

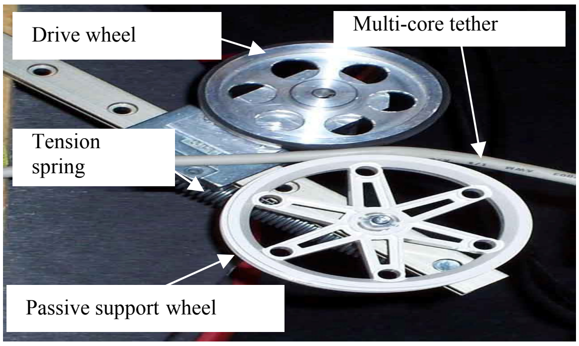


Figure 1: Tether Winding Unit (TWU).

3.2 Force Measurement Unit (FMU)

This unit comprises of a force sensor to measure the force exerted on the tether. This unit is mounted on the other robot (Robot-B). Tether entanglement detection is carried out in the following steps:

Step-1: The tether connecting two robots is pulled taut using the TWU.

Step-2: During that process, the horizontal force exerted on the tether is measured using the FMU.

Step-3: Based on the pattern in which the force exerted on the tether increases, it can be identified whether the tether is snagged by an obstacle or not.

4 EXPERIMENTAL RESULTS

Tether Entanglement can be detected experimentally through *Snag Test,* which makes use of TWU attached to one robot and FMU attached to the other robot. An open loop test is performed under four different scenarios with or without obstacles. The input voltage and the output force are recorded during each test. Each of the four scenarios depicts different levels of friction and different tether dynamics involved when the tether is being pulled. The scenarios are as follows:

Case-A is the scenario in which the tether is freely hanging and there is no entanglement. Case-B models the scenario in which the tether is stuck in rubble and is subjected to friction at discrete points along its length, as a result of which there is an uneven movement when it is being pulled. In Case-C, as the tether is wound around a pillar-like object, the friction would be so high that the TWU would not be able to hold the tether taut. Case-D depicts a scenario in which the tether is bent by pillar-like object and there is slow and steady movement of the tether when it is being pulled. This is because the friction is uniform through out the length of contact with the obstacle. Figure 2 shows a snapshot of rubble and pillar-like objects respectively.

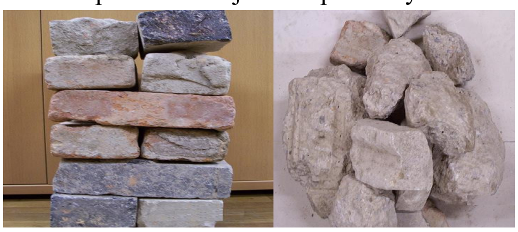


Figure 2: Snapshot of rubble and pillar-like objects.

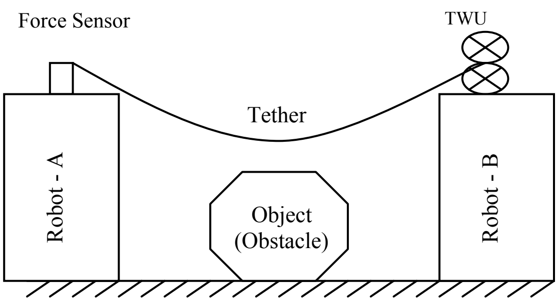


Figure 3: Schematic of Experimental Setup.

The snag test is conducted using unequal square wave pulse, which excites the system so that the dynamic effects of the tether being pulled are tested on different lengths of the tether at different stages of the experiment. Also the pulsing signal would excite the stick-slip friction. A schematic of the experimental setup is shown in Figure-3.

The force sensor readings are plotted against time for all the four scenarios along with the unequal square wave input signal as in Figure-4. Initially as the input voltage is zero, the force exerted on the tether remains a constant. After 6000ms, the voltage raises to +2 V. This is reflected in the graph as steep raise in the force value. It is also observed that there is a lag between the time of application of the voltage pulse and the time at which the force value starts to raise. This is because the force exerted on one end of the tether by TWU has to reach the other end of the tether containing the FSU.

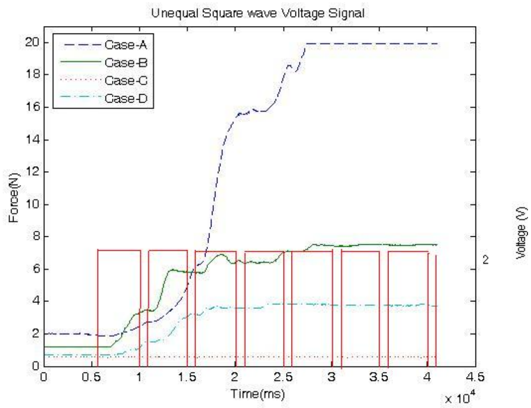


Figure 4: Input-Output Graph.

5 CHARACTERISATION OF TETHER ENTANGLEMENT

In Figure-4, the force value for Case-A raises steeply whenever the voltage pulse is applied because the tether is hanging freely. For Case-B, there is uneven raise in force value, as the friction from the rubble acts at discrete point throughout the length of contact of the tether. Case-C has no effect in force value, as the tether is completely snagged and the friction is so high that it is not possible for the TWU to pull the tether. For Case-D, as there is uniform friction throughout the length of the tether, there is a smooth transition in the force value.

The force curves are analyzed using three different methods namely Range of force analysis, Area under the curve analysis and Static Model

Analysis. From these analyses, an attempt has been made to identify the type of snag from the force sensor readings.

5.1 Range of Force Analysis

In this method, the force curve is preprocessed so that any offset in force is eliminated. From the force curves, it is very evident that when the tether is not snagged by an obstacle (Case-A), the TWU holds the tether taut and the maximum force is around 18 N. For all the scenarios where the tether is entangled, the maximum force is around 6 N as shown in Table-1. Thus Force Range can be used as a parameter to model the type of snag.

5.2 Area Under Curve Analysis

If there are false spikes in the force curve due to factors like slippage, drift, small object falling on the tether, analysis using range of force would be misleading. Area under curve analysis would reduce such effects. After eliminating the offset from the curves, the area is calculated as summation of the product of the time interval (20ms) and corresponding force value. The mean area of the force curve for the three samples is calculated for all the four different scenarios and listed in Table-1.

It could be seen that Case-A has maximum area of around 400 square units. For Case-B and Case-D the area is around 200 and 100 square units respectively. Case-C has least area of less than unity. Thus for a given input signal, based on the area under the force curve, the type of snag can be determined. For this analysis, time duration of the test plays a significant role, as the area of the curve is directly proportional to time.

Table 1: Average values of Force Range and Area under Curve.

Case	Force Range (N)	Area under Curve (square units)
A	18.195	400.0576
B	6.627	173.9507
C	0.124	0.1794
D	3.289	89.2996

6 TETHER MODELLING

From the experimental results it was observed that the system is non-linear. This is evident from the force curves in Figure-4. A non-linear model of the system would give better insight into the behavior of the system. A static model of the Tether Entanglement Detection System (TEDS) is shown in Figure-5. It comprises of two robots (Robot-A and Robot-B) connected using a tether. Robot-A has the tether linked with FMU. Robot-B has the tether passing through TWU. The following are the parameters, which influence the model.

F_{pull} - Horizontal pulling force (N)
θ_{sag} - Sag angle of the tether (radian)
α_{wheel} - Angular Velocity of drive wheel (radian /s)
L_c - Half of the catenary length of the tether (m)
L_h - Half of the horizontal length of the tether (m)
a - Distance between the vertex and the axis of the catenary curve (m)
Z_w - Distance of the top of the catenary curve from its axis (m)

6.1 Static Model of Freely Hanging Tether - Derivation

A static model of the tether based robot system has been derived. It is assumed that the dynamic effects of the tether are negligible because the angular velocity of the wheel α_{wheel} is low. It is also assumed that the mass is evenly distributed throughout the length of the tether and the curve created by the freely hanging tether is a catenary curve. A catenary curve is the shape created by a chain-like object fixed on both ends and hanging freely under the force of gravity. The model can be used to determine the horizontal pulling force (F_{pull}) acting on the tether and the sag angle (θ_{sag}) of the tether (Flugge, 1962).

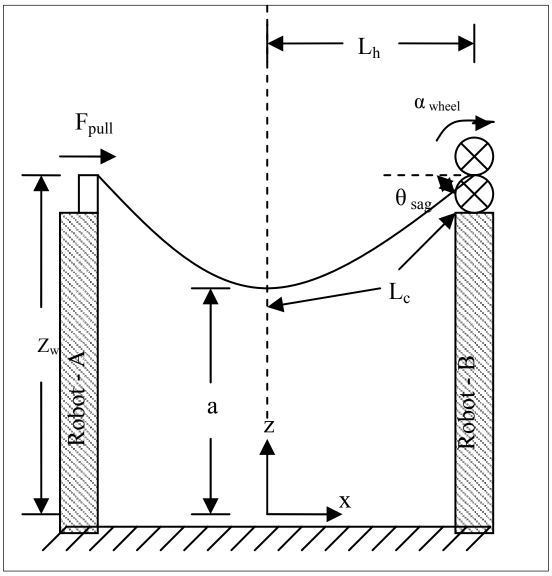


Figure 5: Tether Entanglement Detections System (TEDS) Model.

6.1.1 Horizontal Pulling Force Derivation

The horizontal pulling force (F_{pull}) is given by the following formula:

$$F_{pull} = q \times a \tag{1}$$

where
q – weight per unit length of the tether (N/m)

In order to determine '*a*', the following formulae are used.

$$L_c = L_{ini} - \left(\left(\frac{R}{2} \right) \times \alpha_{wheel} \times t \right) \tag{2}$$

$$L_c = a \times \sinh\left(\frac{L_h}{a} \right) \tag{3}$$

where
L_{ini} – Half the initial length of the tether (m)
R – Radius of the wheel (m)

Formula (2) is used to deduce the value of L_c. This value is used in formula (3) to find out the value of '*a*' by creating the following non-linear equation:

$$f(a) = L_c - \left(a \times \sinh\left(\frac{L_h}{a} \right) \right) = 0 \tag{4}$$

Newton-Raphson iterative method is used to solve the above equation. For that the derivative of *f(a)* is needed.

$$f'(a) = -\sinh\left(\frac{L_h}{a} \right) + \left(\left(\frac{L_h}{a} \right) \times \cosh\left(\frac{L_h}{a} \right) \right) \tag{5}$$

Assume the first guess $a_0 = L_h$, then

$$a_1 = a_0 - \left(\frac{f(a_0)}{f'(a_0)} \right), \; a_2 = a_1 - \left(\frac{f(a_1)}{f'(a_1)} \right) \cdots$$

This is repeated until

$$\left(\frac{|a_{k+1} - a_k|}{|a_{k+1}|} \right) < threshold \tag{6}$$

Threshold can be lower than 0.00001. Lower the threshold higher is the accuracy of the value of '*a*'. The value of a_{k+1} is used as the value of '*a*' in formula (1).

6.1.2 Sag Angle Derivation

The formula for Sag angle is as follows:

$$\theta_{sag} = a\cos\left(\frac{a}{Z_w} \right) \tag{7}$$

where

$$Z_w = a \times \cosh\left(\frac{L_h}{a} \right) \tag{8}$$

6.2 Analysis

The static model is verified experimentally by measuring the force sensor readings for different catenary length of the tether (L_c) keeping the distance between the robots (L_h) as constant. Then a graph is plotted with x-axis containing the ratio between L_h and L_c and y-axis containing the corresponding force readings. In the same graph the force curve predicted using the static model for the same value of L_h is drawn as shown in the Figure-6. It is observed that the actual force readings are very close to the predicted values. This validates the static model.

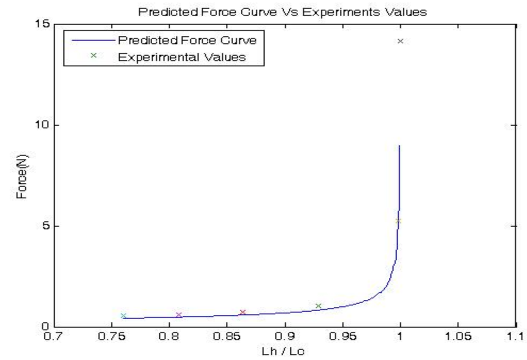


Figure 6: Predicted Force Curve Vs Experimental values.

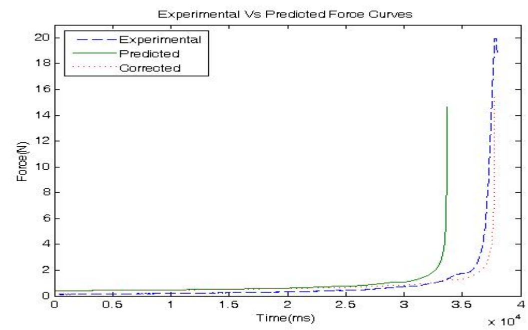


Figure 7: Actual Vs Predicted Vs Corrected Force Curve.

Figure-7 shows the actual force readings and those predicted using the static model for Case-A. The predicted force curve closely follows the pattern of the actual force curve except that it lags in time. This is because the angular velocity of the drive wheel will reduce when it is running under load (tether passing through the wheels) compared to no load condition. This is verified by simulating the predicted force curve with 90% of the measured angular velocity. The corrected curve matched very closely to the experimental force curve as shown in Figure-7. Another reason for the time lag could be attributed to slippage of the tether. An optical encoder could be attached to the wheel to measure the tether length, as it eliminates the time lag error.

From the above three analyses, it is evident that the static model analysis is more promising than the other two methods in terms of providing an accurate model of freely hanging tether. Such a model can be used to predict the force curve for freely hanging scenario and the predicted curve can be compared with the experimental curve. Based on the error between the two curves, it could be identified whether the tether is freely hanging or snagged with obstacles. The static model can also be used to identify different types of snags if dynamic effects are introduced into it. One such approach could be friction modeling.

7 CURRENT WORK

Currently friction modeling is being investigated to understand the dynamic effects of the system. Also a robust and low-cost 3D localization strategy for a swarm of tethered robots is being developed. This technique does not require an environment map for localization. It includes a tether length measurement unit (TLMU) and a tether orientation measurement unit (TOMU) to localize the robot in 3D space. TLMU comprises of an optical encoder attached to the passive wheel of the TWU to measure the length of the tether. TOMU consists of a joystick attached to the end of the TWU to measure pitch and roll of the tether.

8 CONCLUSION

In this paper a novel, low-cost and robust system, which does not require localization or environment map to detect tether entanglement has been proposed. A static model has been derived for the proposed system. Experiments have been conducted to verify the validity of the approach. The results are analyzed using three different methods. From the analyses it is clear that the static model analysis is a promising way of detecting entanglement because it clearly identifies the scenario in which the tether is freely hanging.

REFERENCES

Erik H. Gustafon, Christopher T. Lollini, Bradley, E. Bishop, Carl E. Wick, “Swarm Technology for Search and Rescue through Multi-Sensor Multi-Viewpoint Target Identification”, Proceedings of the Thirty-Seventh South-eastern Symposium on System Theory (SSST 2005), Page(s):352 – 356

Edwardo F. Fukushima, Noriyuki Kitamara, Shigeo Hirose, “A New Flexible Component for Field Robotic System”, Proceedings of the IEEE International Conference on Robotics and Automation, San Francisco, CA, Page(s): 2583 – 2588

Doughlas P. Perrin, Albert Kwon, Robert D. Howe, “A Novel Actuated Tether Design for Rescue Robots using Hydraulic Transients”, Proceedings of the IEEE International Conference on Robotics & Automation, Page(s): 3482-3487

Patrick G.Xavier,“Shortest path planning for a tethered robot or an anchored Cable”, Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 1999), Volume 2, Page(s): 1011- 1017

Susan Hert, Vladimir Lumelsky, “Motion Planning in R^3 for Multiple Tethered Robots”, IEEE Transactions on Robotics and Automation, Volume 15, Issue 4, August 1999, Page(s): 623 – 639

Chee Kong Cheng; Leng, G., “Cooperative search algorithm for distributed autonomous robots”, Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, (IROS 2004), Volume 1, Page(s):394 - 399

P. Saeedi, D. G. Lowe, P.D.Lawrence, “3D Localization and tracking in unknown environments”, Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2003), Volume 1, Page(s):1297 –1303

Salah Sukkarieh, Peter Gibbens, Ben Grocholsky, Keith Willis, Hugh F. Durrant-Whyte, “A Low-Cost Redundant Inertial Measurement Unit for Unmanned Air Vehicles”, International Journal of Robotics Research, Volume 19, No. 11, Page(s): 1089-1103

Alejandro Ramirez-Serrano, Giovanni C. Pettinaro, “Navigation of Unmanned Vehicles Using a Swarm of Intelligent Dynamic Landmarks”, Proceedings of IEEE International Workshop on Safety, Security and Rescue Robotics (SSRR), Page(s): 60 – 65

Robert Grabowski, Pradeep Khosla, Howie Choset, “Development and Deployment of a Line of Sight Virtual Sensor for Heterogeneous Teams”, Proceedings of IEEE International Conference on Robotics and Automation (ICRA 2004), Volume 1, Page(s):3024 – 3029

W.S. Wijesoma, L. D. L. Perera, M. D. Adams, “Navigation in Complex Unstructured Environments”, Proceedings of 8th International Conference on Control, Automation, Robotics and Vision (ICRA 2004), Volume 1, Page(s): 167 – 172

W. Flugge, “Handbook of Engineering Mechanics”, First Edition 1962, McGraw-Hill Book Company, Page(s): 4.7, 21.15

INTERCEPTION AND COOPERATIVE RENDEZVOUS BETWEEN AUTONOMOUS VEHICLES

Yechiel J. Crispin and Marie-Emmanuelle Ricour
Department of Aerospace Engineering, Embry-Riddle Aeronautical University, Daytona Beach, Florida 32114, USA
crispinj@erau.edu

Keywords: Multiple Vehicles, Cooperative Rendezvous, Interception, Autonomous Vehicles, Optimal Control, Genetic Algorithms.

Abstract: The rendezvous problem between autonomous vehicles is formulated as an optimal cooperative control problem with terminal constraints. A major approach to the solution of optimal control problems is to seek solutions which satisfy the first order necessary conditions for an optimum. Such an approach is based on a Hamiltonian formulation, which leads to a difficult two-point boundary-value problem. In this paper, a different approach is used in which the control history is found directly by a genetic algorithm search method. The main advantage of the method is that it does not require the development of a Hamiltonian formulation and consequently, it eliminates the need to deal with an adjoint problem. This method has been applied to the solution of interception and rendezvous problems in an underwater environment, where the direction of the thrust vector is used as the control. The method is first tested on an interception chaser-target problem where the passive target vehicle moves along a straight line at constant speed. We then treat a cooperative rendezvous problem between two active autonomous vehicles. The effects of gravity, thrust and viscous drag are considered and the rendezvous location is treated as a terminal constraint.

1 INTRODUCTION

In an active-passive rendezvous problem between two vehicles, the passive or target vehicle does not apply any control maneuvers along its trajectory. The active or chaser vehicle is controlled or guided such as to meet the passive vehicle at a later time, matching both the location and the velocity of the target vehicle. In a cooperative rendezvous problem, the two vehicles are active and maneuver such as to meet at a later time, at the same location with the same velocity. The rendezvous problem consists of finding the control sequences or the guidance laws that are required in order to bring the two vehicles to a final state of rendezvous.

An optimal control problem consists of finding the control histories (control as a function of time) and the state variables of the dynamical system such as to minimize a performance index. The differential equations of motion of the vehicles are then treated as dynamical constraints. A possible approach to the solution of the rendezvous problem is to formulate it as an optimal control problem in which it is required to find the controls such as to minimize the differences between the final locations and final velocities of the vehicles. The methods of approach for solving optimal control problems include the classical indirect methods and the more recent direct methods. The indirect methods are based on the calculus of variations and its extension to the maximum principle of Pontryagin, which is based on a Hamiltonian formulation. These methods use necessary first order conditions for an optimum, they introduce adjoint variables and require the solution of a two-point boundary value problem (TPBVP) for the state and adjoint variables. Usually, the state variables are subjected to initial conditions and the adjoint variables to terminal or final conditions. Two-point boundary value problems (TPBVP) are much more difficult to solve than initial value problems (IVP). For this reason, direct methods of solution have been developed which avoid completely the Hamiltonian formulation. For example, a possible approach is to reformulate the optimal control problem as a nonlinear programming (NLP) problem by direct transcription of the dynamical equations at prescribed discrete points or collocation points. This method was originally developed by Dickmanns and Well (Dickmanns, 1975.) and used by Hargraves and Paris (Hargraves, 1987) to solve several atmospheric trajectory optimization problems.

Another class of direct methods is based on biologically inspired methods of optimization. These include evolutionary methods such as genetic algorithms, particle swarm optimization methods and ant colony optimization algorithms. PSO) mimics the social behavior of a swarm of insects, see for example (Venter, 2002), (Crispin,2005). Genetic Algorithms (GAs) (Goldberg, 1989) are a powerful alternative method for solving optimal control problems, see also (Crispin, 2006 and 2007). GAs use a stochastic search method and are robust when compared to gradient methods. They are based on a directed random search which can explore a large region of the design space without conducting an exhaustive search. This increases the probability of finding a global optimum solution to the problem. They can handle continuous or discontinuous variables since they use binary coding. They require only values of the objective function but no values of the derivatives. However, GAs do not guarantee convergence to the global optimum. If the algorithm converges too fast, the probability of exploring some regions of the design space will decrease. Methods have been developed for preventing the algorithm from converging to a local optimum. These include fitness scaling, increased probability of mutation, redefinition of the fitness function and other methods that can help maintain the diversity of the population during the genetic search.

2 COOPERATIVE RENDEZVOUS AS AN OPTIMAL CONTROL PROBLEM

We study trajectories of vehicles moving in an incompressible viscous fluid in a 2-dimensional domain. The motion is described in a cartesian system of coordinates (x,y), where x is positive to the right and y is positive downwards in the direction of gravity. The vehicle weight acts downward, in the positive y direction. The vehicle has a propulsion system that delivers a thrust of constant magnitude. The thrust is always tangent to the trajectory. The vehicle is controlled by varying the thrust direction. Since the fluid is viscous, a drag force acts on the vehicle, in the opposite direction of the velocity. The control variable of the problem is the thrust direction $\gamma(t)$. The angle $\gamma(t)$ is measured positive clockwise from the horizontal direction (positive x direction).

The rendezvous problem is formulated as an optimal control problem, in which it is required to determine the control functions, or control histories $\gamma_1(t)$ and $\gamma_2(t)$ of the two vehicles, such that they will meet at a prescribed location at the final time t_f. Since GAs deal with discrete variables, we discretize the values of $\gamma(t)$. We assume that the mass of the vehicles is constant. The motion of the vehicle is governed by Newton's second law and the kinematic relations between velocity and distance:

$$d(mV)/dt = mg + T + D \tag{2.1}$$

$$dx/dt = V\cos\gamma \tag{2.2}$$

$$dy/dt = V\sin\gamma \tag{2.3}$$

where D is the drag force acting on the body, V is the velocity vector, T is the thrust vector and g is the acceleration of gravity. Since we assumed m is constant,

$$dV/dt = g + T/m + D/m \tag{2.4}$$

Writing this equation for the components of the forces along the tangent to the vehicle's path, we get:

$$dV/dt = g\sin\gamma + T/m - D/m \tag{2.5}$$

The drag D is given by:

$$D = \frac{1}{2}\rho V^2 S C_D \tag{2.6}$$

where ρ is the fluid density, S a typical cross-section area of the vehicle and C_D the drag coefficient, which depends on the Reynolds number $Re = \rho V d/\mu$, where d is a typical dimension of the vehicle and μ the fluid viscosity.

Substituting the drag from Eq.(2.6) and writing $T = amg$, where a is the thrust to weight ratio T/mg, Eq.(2.5) becomes:

$$dV/dt = g\sin\gamma + ag - \rho V^2 S C_D/2m \tag{2.7}$$

Introducing a characteristic length L_c, time t_c and speed v_c as

$$L_c = 2m/\rho S C_D,\ \ t_c = \sqrt{L_c/g},\ \ v_c = \sqrt{g L_c} \tag{2.8}$$

the following nondimensional variables can be defined:

$$x = L_c\bar{x}, \quad y = L_c\bar{y}$$

$$t = (L_c/g)^{1/2}\bar{t}, \quad V = (gL_c)^{1/2}\overline{V} \tag{2.9}$$

Substituting in Eq.(2.7), we have:

$$d\overline{V}/d\bar{t} = a + \sin\gamma(t) - \overline{V}^2 \tag{2.10}$$

Similarly, the other equations of motion can be written in nondimensional form as

$$d\bar{x}/d\bar{t} = \overline{V}\cos\gamma(t) \tag{2.11}$$

$$d\bar{y}/d\bar{t} = \overline{V}\sin\gamma(t) \tag{2.12}$$

For each vehicle the initial conditions are:

$$V(0) = V_0, \ x(0) = x_0, \ y(0) = y_0 \tag{2.13}$$

In rendezvous problems, terminal constraints on the final location can also be required

$$\bar{x}(\bar{t}_f) = \bar{x}_f = x_f/L_c$$

$$\bar{y}(\bar{t}_f) = \bar{y}_f = y_f/L_c \tag{2.14}$$

where the nondimensional final time is given by

$$\overline{t_f} = t_f/\sqrt{L_c/g}$$

We now define a rendezvous problem between two vehicles. We denote the variables of the first vehicle by a subscript 1 and those of the second vehicle by a subscript 2. We will now drop the bar notation indicating nondimensional variables. The two vehicles might have different thrust to weight ratios, which are denoted by a_1 and a_2, respectively. The equations of motion for the system of two vehicles are:

$$dV_1/dt = a_1 + \sin\gamma_1(t) - V_1^2 \tag{2.15}$$

$$dx_1/dt = V_1\cos\gamma_1(t) \tag{2.16}$$

$$dy_1/dt = V_1\sin\gamma_1(t) \tag{2.17}$$

$$dV_2/dt = a_2 + \sin\gamma_2(t) - V_2^2 \tag{2.18}$$

$$dx_2/dt = V_2\cos\gamma_2(t) \tag{2.19}$$

$$dy_2/dt = V_2\sin\gamma_2(t) \tag{2.20}$$

The vehicles can start the motion from different locations and at different speeds. The initial conditions are given by:

$$V_1(0) = V_{10}, \ x_1(0) = x_{10}, \ y_1(0) = y_{10} \tag{2.21}$$

$$V_2(0) = V_{20}, \ x_2(0) = x_{20}, \ y_2(0) = y_{20} \tag{2.22}$$

The cooperative rendezvous problem consists of finding the control functions $\gamma_1(t)$ and $\gamma_2(t)$ such as the two vehicles arrive at a given terminal location (x_f, y_f) and at the same speed in the given time t_f. The terminal constraints are then given by:

$$x_1(t_f) = x_f, \ x_2(t_f) = x_f$$

$$y_1(t_f) = y_f, \ y_2(t_f) = y_f \tag{2.23}$$

$$V_1(t_f) = V_2(t_f)$$

We can also define an interception problem, of the target-chaser type, in which one vehicle is passive and the chaser vehicle maneuvers such as to match the location of the target vehicle, but not its velocity. Consistent with the above terminal constraints, we define the following objective function for the optimal control problem:

$$f(x_j(t_f), V_j(t_f)) = \sum_{j=1}^{N_v} \left\| x_j(t_f) - x_f \right\|^2 + \Delta V_j^2(t_f) = \min \tag{2.24}$$

where N_v is the number of vehicles, $x_f = (x_f, y_f)$ is the prescribed interception or rendezvous point and $\Delta V_j^2(t_f)$ is the square of the difference between the magnitudes of the velocities of the vehicles. If we define the norm as a Euclidean distance, we can write the following objective function for the case of two vehicles:

$$f[x_1(t_f), x_2(t_f), y_1(t_f), y_2(t_f), V_1(t_f), V_2(t_f)] =$$

$$= (x_1(t_f) - x_f)^2 + (x_2(t_f) - x_f)^2 + (y_1(t_f) - y_f)^2$$

$$+ (y_2(t_f) - y_f)^2 + (V_1(t_f) - V_2(t_f))^2 = \min \tag{2.25}$$

We use standard numerical methods for integrating the differential equations. The time interval t_f is divided into N time steps of duration $\Delta t = t_f/N$. The discrete time is $t_i = i\Delta t$. We used a second-order Runge-Kutta method with fixed time step. We also tried a fourth-order Runge-Kutta method and a variable time step and found that the results were not sensitive to the method of integration. The control function $\gamma(t)$ is discretized to $\gamma(i) = \gamma(t_i)$ according to the number of time steps N used for the numerical integration. Depending on the accuracy of the desired solution, we can choose the number of bits n^i for encoding the value of the control $\gamma(i)$ at each time step i. The size n^i used for encoding $\gamma(i)$ and the number of time steps N will have an influence on the computational time. Therefore n^i and N must be chosen carefully, in order to obtain an accurate enough solution in a reasonable time. The total length of the chromosome is given by:

$$L_{ch} = n^i N N_v \tag{2.26}$$

For this problem, we were able to increase the rate of convergence of the algorithm by introducing heuristic arguments. For instance, having noticed that $\gamma(t)$ is a monotonically decreasing function of time, we were able to speed up the algorithm by choosing a function with such a property, a priori. Therefore, instead of waiting for the algorithm to converge towards a monotonous $\gamma(t)$, we can sort the values of γ of each individual solution in decreasing order, before calculating its fitness. We also use smoothing of the control function by fitting a third or fourth-order polynomial to the discrete values of γ. The values of the polynomial at the N discrete time points are then used as the current values of γ and are used in the integration of the differential equations.

An appropriate range for γ is $\gamma \in [0, \pi/2]$. We choose $N = 30$ as a reasonable number of time steps. We now need to choose the parameters associated with the Genetic Algorithm. First, we select the lengths of the "genes" for encoding the discrete values of γ. A choice of $n^i = 8$ bits for $\forall i \in [0, N-1]$ was made. The interval between two consecutive possible values of γ is given by:

$$\Delta\gamma = (\gamma_{max} - \gamma_{min})/(2^n - 1) \approx 0.0062\,\text{rad} = 0.35\,\text{deg}$$

For two vehicles and 30 time steps, the length of a chromosome is then given by:

$$L_{ch} = n^i N N_v = 480\,\text{bits}$$

A reasonable size for the population of solutions is typically in the range $n_{\text{pop}} \in [50, 200]$. For this problem, there is no need for a particularly large population, so we select $n_{\text{pop}} = 50$. The probability of mutation is set to a value of 5 percent $p_{\text{mut}} = 0.05$.

3 CHASER-TARGET INTERCEPTION

We study a chaser-target interception problem between two vehicles. In this case the first vehicle is active and the second is passive and moves along a straight line at a constant depth $y_2 = y_f$, constant speed V_2 and constant angle $\gamma_2 = 0$. The two vehicles start from different points and the interception occurs at the known depth of the target vehicle $y_2 = y_f$. The horizontal distance x_f to the interception point is free. We present the case where the target moves at moderate speed and can be captured by the active chaser vehicle. Since this is an interception problem, we do not require matching between the final velocities.

$$a_1 = a = 0.05, \quad a_2 = 5a = 0.25$$

$$t_0 = 0, t_f = 5$$

$$\gamma_1 \in [0, \pi/2], \quad \gamma_2 = 0, \quad y_f = 2 \tag{3.1}$$

The initial conditions are:

$$x_1(0) = x_2(0) = 0, \; y_1(0) = 0$$

$$y_2(0) = y_f = 2, \; V_1(0) = 0$$

$$V_2(0) = V_2 = \sqrt{a_2 + \sin\gamma_2} = \sqrt{a_2} = 0.5 \tag{3.2}$$

In order to match the final locations, the following objective or fitness function is defined:

$$f[x_1(t_f), x_2(t_f), y_1(t_f)] =$$

$$= (x_1(t_f) - x_2(t_f))^2 + (y_1(t_f) - y_f)^2 = \min \tag{3.3}$$

The parameters for this test case are summarized in the following table and the results are given in Figs. 1-3.

N_v	n_{pop}	n^i	N
2	50	8	30
p_{mut}	N_{gen}	$\gamma_{1\min}$	$\gamma_{1\max}$
0.05	50	0	$\pi/2$
a	t_0	t_f	(x_{01}, y_{01})
0.05	0	5	$(0,0)$
(x_{02}, y_{02})	(V_{01}, V_{02})	x_f	y_f
$(0,2)$	$(0, \sqrt{5a})$	free	2

4 RENDEZVOUS BETWEEN TWO ACTIVE VEHICLES

We next treat a rendezvous between two vehicles. The two vehicles start from different points and rendezvous at point (x_f, y_f) in a given time t_f. The vehicles have the same thrust to weight ratio $a_1 = a_2 = a$.

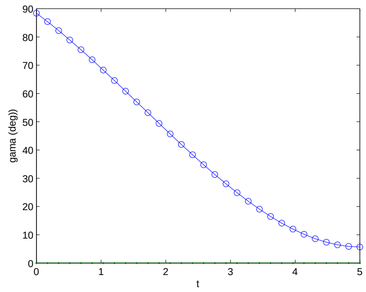


Figure 1: Control function $\gamma_1(t)$ and $\gamma_2 = 0$ for a chaser-target interception with prescribed target depth.

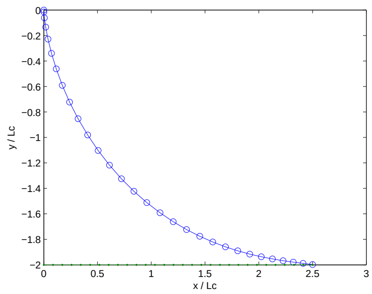


Figure 2: Trajectories for a chaser-target interception with prescribed target depth. The sign of y was reversed for plotting.

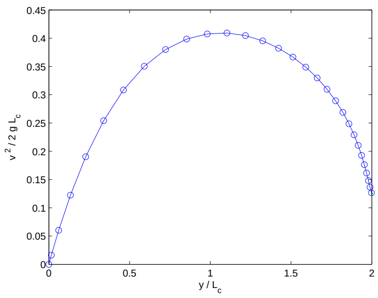


Figure 3: Kinetic energy as a function of depth for a chaser-target interception with prescribed target depth.

$$a_1 = a = 0.05, \quad a_2 = a = 0.05$$

$$t_0 = 0,\ t_f = 5,\ x_f = 2.8,\ y_f = 2$$

$$\gamma_1 \in [0, \pi/2] \quad \gamma_2 \in [0, \pi/2] \tag{4.1}$$

with initial conditions:

$$x_1(0) = 0,\ y_1(0) = 0,\ x_2(0) = 0.5,\ y_2(0) = 0$$

$$V_1(0) = 0,\ V_2(0) = 0 \tag{4.2}$$

In a rendezvous problem, the objective or fitness function is given by:

$$f[x_1(t_f), x_2(t_f), y_1(t_f), y_2(t_f), V_1(t_f), V_2(t_f)] =$$

$$= (x_1(t_f) - x_f)^2 + (y_1(t_f) - y_f)^2 + (x_2(t_f) - x_f)^2$$

$$+(y_2(t_f) - y_f)^2 + (V_1(t_f) - V_2(t_f))^2 = \min \tag{4.3}$$

The parameters for this test case are summarized in the following table and the results are presented in Figs.4-5.

N_v	n_{pop}	n^i	N
2	50	8	30
p_{mut}	N_{gen}	$\gamma_{1\min}$	$\gamma_{1\max}$
0.05	200	0	$\pi/2$
a	t_0	t_f	(x_{01}, y_{01})
0.05	0	5	$(0,0)$
(x_{02}, y_{02})	(V_{01}, V_{02})	x_f	y_f
$(0.5,0)$	$(0,0)$	2.8	2

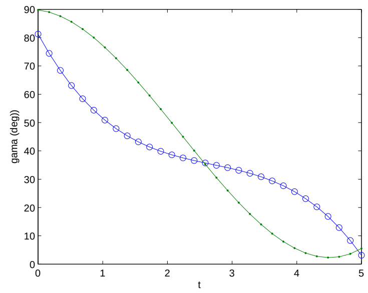


Figure 4: Control functions $\gamma_1(t)$ and $\gamma_2(t)$ for rendezvous between two vehicles with prescribed terminal point.

5 CONCLUSION

The rendezvous problem between two active autonomous vehicles moving in an underwater environment has been treated using an optimal control formulation with terminal constraints. The two vehicles have fixed thrust propulsion system and use the direction of the velocity vector for steering and control. We use a genetic algorithm to determine directly the control histories of both vehicles by evolving populations of possible solutions. An interception problem, where one vehicle moves along a straight line with constant

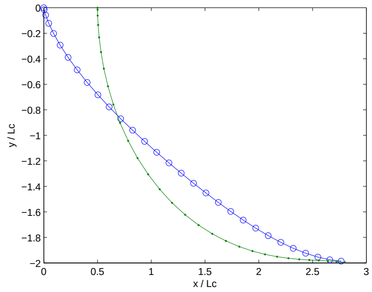


Figure 5: Trajectories for rendezvous between two vehicles with prescribed terminal point.

velocity and the second vehicle acts as a chaser, maneuvering such as to capture the target in a given time, has also been treated as a test problem. It was found that the chaser can capture the target within the prescribed time as long as the target speed is below a critical speed. We then treated the rendezvous problem between two active vehicles where both the final positions and velocities are matched. As the initial horizontal distance between the two vehicles is increased, it becomes more difficult to solve the problem and the genetic algorithm requires more generations to converge to a near optimal solution.

REFERENCES

Bryson, A. E., “Dynamic Optimization”, 1999, Addison Wesley Longman.

Crispin, Y., 2005, “Cooperative control of a robot swarm with network communication delay”, First International Workshop on Multi-Agent Robotic Systems (MARS 2005), INSTICC Press, Portugal.

Crispin, Y., 2006, “An evolutionary approach to nonlinear discrete-time optimal control with terminal constraints”, in Informatics in Control, Automation and Robotics I, pp.89-97, Springer, Dordrecht, Netherlands.

Crispin, Y., 2007, “Evolutionary computation for discrete and continuous time optimal control problems”, in Informatics in Control, Automation and Robotics II, Springer, Dordrecht, Netherlands.

Dickmanns, E. D. and Well, H., 1975, “Approximate solution of optimal control problems using third-order hermite polynomial functions,” Proceedings of the 6th Technical Conference on Optimization Techniques, Springer-Verlag, New York

Goldberg, D.E., 1989, “Genetic Algorithms in Search, Optimization and Machine Learning”, Addison-Wesley.

Hargraves, C. R. and Paris, S. W., 1987, “Direct Trajectory Optimization Using Nonlinear Programming and Collocation”, AIAA Journal of Guidance, Control and Dynamics, Vol. 10, pp. 338-342

Venter, G. and Sobieszczanski-Sobieski, J., 2002, “Particle Swarm Optimization”, 43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference.

COORDINATED MOTION CONTROL OF MULTIPLE ROBOTS

Wojciech Kowalczyk and Krzysztof Kozlowski
Chair of Control and Systems Engineering, Poznan University of Technology, Piotrowo 3A, 60-965 Poznan, Poland
wojciech.kowalczyk@put.poznan.pl, krzysztof.kozlowski@put.poznan.pl

Keywords: Cooperative robotics, robot formation, team work.

Abstract: In this paper a set of robot coordination approaches is presented. Described method 0s are based on formation function concept. Accuracy of different approaches is compared using formation function time graphs. Virtual structure method is analyzed, then virtual structure expanded with behavioral formation feedback is presented. Finally leader-follower scheme is described. Presented methods are illustrated by simulation results. Differentially driven nonholonomic mobile robots were used in simulations.

1 INTRODUCTION

Multiple robot coordination is currently one of the most investigated area of robotics. Great development in computer sciences, multi-agent systems and availability of low-cost, effective and compact digital equipment caused that many researchers focused their attention on this subject. Multi-agent robotic systems have wide range of applications: service robots, transportation systems, mapping, surveillance, security and many others.

Multi-robot coordination methods can be conventionally partitioned into three classes of approaches: virtual structure approach (Egerstedt and Hu, 2001), (P. Ogren, 2002), (W. Kang, 2000), (Kar-Han Tan, 1997), behavioral approach (Esposito and Kumar, 2000), (J. R. Lawton and Beard, 2000), (Yamaguchi, 1998), (Yamaguchi, 1999), (Kowalczyk and Kozlowski, 2005) and leader follower scheme (R. Fierro and Ostrowski, 2001), (J. Spletzer and Kumar, 2001) (sometimes treated as a combination of first two approaches). Each of them is more or less suitable for particular application. There exist some solutions with characteristic features of more than one approach (B. J. Young and Kelsey, 2001).

In virtual structure methods control is centralized. It is suitable for the tasks that require high precision coordinated motion of few robots, i.e. when it is necessary to transport one huge object by the formation of robots. Centralized architecture of the control cause that system is not scalable. Adding new agents causes more intensive utilization of the main controller. This method requires also high-speed communication between main controller and agents. For virtual structure method it is usually relatively easy to analyze and proof stability of the system mathematically.

In behavioral method control is entirely distributed. It is not necessary to use communication; however, using it may increase efficiency. Behavioral methods were inspired by observations in biology and physics. Control is decentralized and in result system is easy scalable. Stability analysis is difficult or even impossible. These methods are not suitable for high-precision motion tasks, but they are very effective for applications that can be decomposed into many independent subtasks. In opposition to virtual structure methods behavioral methods are fault-tolerant.

Leader-follower methods own some features from virtual structure and behavioral methods. Communication can be used to make control more effective, but it is not necessary. Control is distributed and in result easy scalable, however, there is hierarchical dependency between robots and as a result system is not as fault-tolerant as in behavioral approach. Stability

[1]This work was supported by the Polish Ministry of Education and Science grant 1532/T11/2005/29.

roof is usually possible for leader-follower methods. Typical applications for leader-follower methods are spacecraft and aircraft formations. It can be also used in mapping and exploration of the terrain.

Paper is organized as follows. In section 2 we describe feedback linearization of differentially-driven mobile robot. In section 3 virtual structure approach is presented. In section 4 virtual structure approach is expanded with formation keeping behavior. In section 5 leader-follower scheme is presented. In section 6 we conclude the paper. Simulation results are included in sections 3-5.

2 FEEDBACK LINEARIZATION

Most of formation control methods require robots to be fully actuated or transformed into fully actuated. Model of such robot is given:

$$\ddot{P}_i = u_i, \tag{1}$$

where P_i is the position vector of the i-th robot, $P_i \in R^2$, u_i - control force vector exerted on the i-th robot, $u_i \in R^2$, $i = 1, 2, \ldots, N$, N - number of robots.

Two-wheel differentially-driven mobile robot can be transformed into fully actuated using feedback linearization. The same technique can be applied also to other kind of mobile platforms. This causes that formation control can be implemented independently from motion controller and architecture of the robots.

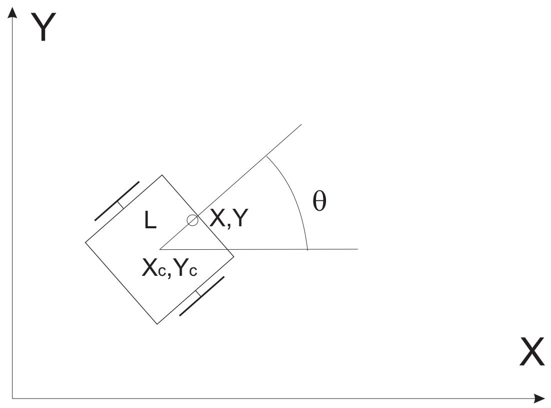


Figure 1: Nonholonomic differentially driven wheeled mobile robot (index designating number of the robot was omitted for clarity).

The motion of the i-th robot is given by:

$$\begin{bmatrix} \dot{x}_{ci} \\ \dot{y}_{ci} \\ \dot{\theta}_i \\ \dot{v}_i \\ \dot{\omega}_i \end{bmatrix} = \begin{bmatrix} v_i \cdot \cos(\theta_i) \\ v_i \cdot \sin(\theta_i) \\ \omega_i \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{m_i} & 0 \\ 0 & \frac{1}{J_i} \end{bmatrix} \begin{bmatrix} F_i \\ \tau_i \end{bmatrix}, \tag{2}$$

where $[x_{ci}, y_{ci}]^T$ - position of the midpoint of the wheel axis, θ_i - orientation of the robot, v_i - linear velocity, ω_i - angular velocity, m_i - mass of the robot, J_i - moment of inertia of the robot, F_i - control force and τ_i - control moment of force.

Dynamics of this kind of robot can be linearized if robot's position output is chosen suitably. As shown in (J. R. Lawton and Beard, 2002) a good choice is position of the point located in a distance L_i along the line that is perpendicular to the wheel axis and intersects with the point $[x_{ci} \;\; y_{ci}]^T$ (Fig. 1). Selected output can be described as follows:

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} x_{ci} \\ y_{ci} \end{bmatrix} + L_i \cdot \begin{bmatrix} \cos(\theta_i) \\ \sin(\theta_i) \end{bmatrix} \tag{3}$$

Differentiating above equation twice we obtain:

$$\begin{aligned} \begin{bmatrix} \ddot{x}_i \\ \ddot{y}_i \end{bmatrix} &= \begin{bmatrix} -v_i\omega_i \sin(\theta_i) - L_i\omega_i^2 \cos(\theta_i) \\ v_i\omega_i \cos(\theta_i) - L_i\omega_i^2 \sin(\theta_i) \end{bmatrix} \\ &+ \begin{bmatrix} \frac{1}{m_i}\cos(\theta_i) & -\frac{L_i}{J_i}\sin(\theta_i) \\ \frac{1}{m_i}\sin(\theta_i) & \frac{L_i}{J_i}\cos(\theta_i) \end{bmatrix} \begin{bmatrix} F_i \\ \tau_i \end{bmatrix} \end{aligned} \tag{4}$$

Since

$$\det \begin{bmatrix} \frac{1}{m_i}\cos(\theta_i) & -\frac{L_i}{J_i}\sin(\theta_i) \\ \frac{1}{m_i}\sin(\theta_i) & \frac{L_i}{J_i}\cos(\theta_i) \end{bmatrix} = \frac{L_i}{m_i J_i} \neq 0 \tag{5}$$

the system with output $[x_i \;\; y_i]^T$ can be output feedback linearized.

The output feedback linearizing control law is

$$\begin{aligned} \begin{bmatrix} F_i \\ \tau_i \end{bmatrix} &= \begin{bmatrix} \frac{1}{m_i}\cos(\theta_i) & -\frac{L_i}{J_i}\sin(\theta_i) \\ \frac{1}{m_i}\sin(\theta_i) & \frac{L_i}{J_i}\cos(\theta_i) \end{bmatrix}^{-1} \\ &\cdot \left(u_i - \begin{bmatrix} -v_i\omega_i \sin(\theta_i) - L_i\omega_i^2 \cos(\theta_i) \\ v_i\omega_i \cos(\theta_i) - L_i\omega_i^2 \sin(\theta_i) \end{bmatrix} \right) \end{aligned} \tag{6}$$

Substituting above result into Eq. (5) and simplifying we obtain feedback linearized robot model given by Eq. (1).

3 VIRTUAL STRUCTURE

In this section virtual structure method is presented. Concept of formation function that was introduced in (P. Ogren, 2002) is used. Virtual structure is suitable for applications that require very precise, coordinated motion of formation of robots. As mentioned in the introduction this approach has some disadvantages, however, in some applications it is the only suitable method.

In Fig. 2 formation of four robots tracks desired trajectory. All robots keep their relative positions P_i, $i = 1, \ldots, 4$ to the current point of desired trajectory P_{trj}.

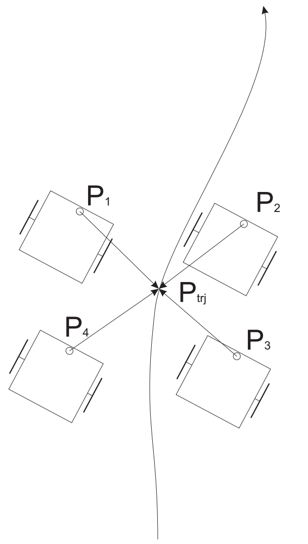


Figure 2: Virtual structure approach - robots of the formation tracks desired trajectory with offsets given by constant vectors (offset vectors).

The formation function is as follows:

$$F = \sum_{i=1}^{N} \|(P_i - P_{i\,of}) - P_{trj}\|^2, \qquad (7)$$

where $P_{trj} = [x_{trj} \;\; y_{trj}]^T$ is the current point on the trajectory to be tracked by the formation, $P_{i\,of} = [x_{i\,of} \;\; y_{i\,of}]^T$ is the offset vector for *i*-th robot. Function F is positive definite, equal to zero only when all robots of the formation match their desired positions.

Control of the *i*-th robot is proposed as follows:

$$u_i = -K_P \begin{bmatrix} \frac{\partial F}{\partial x_i} \\ \frac{\partial F}{\partial y_i} \end{bmatrix} - K_V \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \end{bmatrix}, \qquad (8)$$

where K_P and K_V are positive gains that determine characteristics of the control. Second term in Eq. (8) represents dumping.

In Fig. 3 trajectories of centers of masses of four robots are shown. They follow formation trajectory that starts in $(0,0)$ position and ends in $(-1.1, 4.5)$. Offset vectors for robots $1, \ldots, 4$ are as follows: $(0.25, 0.25)$, $(-0.25, 0.25)$, $(-0.25, -0.25)$ and $(0.25, -0.25)$. Initial orientations of robots are: $\theta_1 = -\frac{\pi}{2}$, $\theta_2 = \pi$, $\theta_3 = \frac{\pi}{2}$ and $\theta_4 = 0$. The values

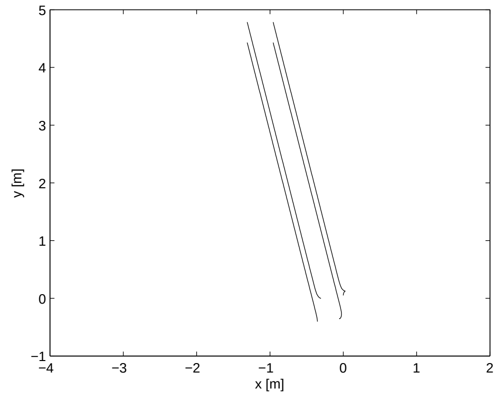


Figure 3: Formation of four robots tracks desired trajectory using virtual structure control method.

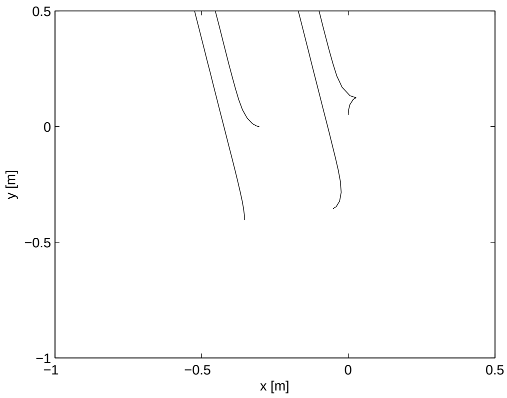


Figure 4: Starting segment of trajectories shown in Fig. 3.

of control gains are: $K_P = 30$ and $K_V = 10$. In Fig. 4 starting segments of robots trajectories are shown. Initially all robots of the formation change their orientations to $\theta_i \approx 1.88 rad$ $(i = 1, \ldots, 4)$ to track the desired trajectory.

In Fig. 5 the graph of formation function as a function of time is shown. It can be used to evaluate the control because value of formation function represents formation error. As one can see in Fig. 5, after transient state (about 1.5s), formation error stabilizes below $0.9m^2$.

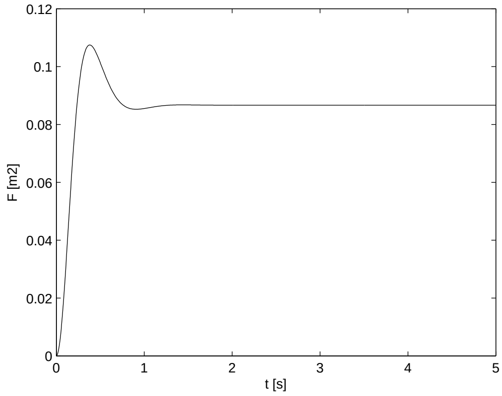


Figure 5: Formation function (time graph) for virtual structure control.

4 VIRTUAL STRUCTURE WITH FORMATION KEEPING BEHAVIOR

In this section we present virtual structure method expanded with behavioral component. This component provide formation feedback that cause formation to slow down when one of robots slows down or when it stops. In such case two concurrent goals occur: trajectory tracking and formation keeping. Presented method does not avoid collisions between robots. The control algorithm try to fulfill both of them. Tuning control gains one can set more to track the trajectory or to keep the formation.

In Fig. 6 formation of four robots tracks desired trajectory. All robots keep their relative positions P_i, to the current position of desired trajectory P_{trj}. Additionally robots keep positions relatively to their neighbors.

The formation function is given as follows:

$$F = F_1 + F_2, \tag{9}$$

where F_1 is given by Eq. (7) and F_2 is as follows:

$$\begin{aligned} F_2 = \sum_{i=1}^{N} \quad & [\quad \|(P_i - P_{i\,of}) - (P_k - P_{k\,of})\|^2 \\ & + \quad \|(P_i - P_{i\,of}) - (P_j - P_{j\,of})\|^2\,], \end{aligned} \tag{10}$$

where $P_{k\,of} = [x_{k\,of} \;\; y_{k\,of}]^T$ and $P_{j\,of} = [x_{j\,of} \;\; y_{j\,of}]^T$ are offset vectors to k-th and j-th neighbor robot; k and j are $\{4,2\}$ for robot 1, $\{1,3\}$ for robot 2, $\{2,4\}$ for robot 3 and $\{3,1\}$ for robot 4. Component F_2 of the formation function represents coupling between robots and formation feedback.

Control of the i-th robot is given as follows:

$$\begin{aligned} u_i = -K_P \begin{bmatrix} \frac{\partial F_1}{\partial x_i} \\ \frac{\partial F_1}{\partial y_i} \end{bmatrix} \quad & - \quad K_F \begin{bmatrix} \frac{\partial F_2}{\partial x_i} \\ \frac{\partial F_2}{\partial y_i} \end{bmatrix} \\ & - \quad K_V \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \end{bmatrix}, \end{aligned} \tag{11}$$

where K_F is a positive factor representing the strength of the formation feedback.

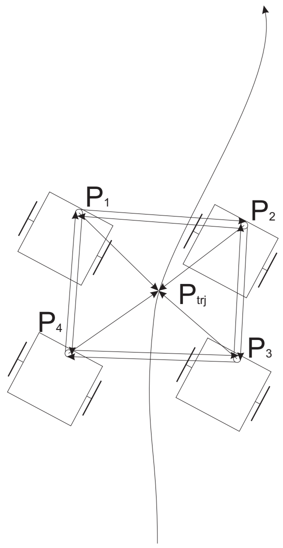


Figure 6: Virtual structure expanded with formation feedback behavior; positions of robots depend not only on the desired formation trajectory but also on positions of other robots of the formation.

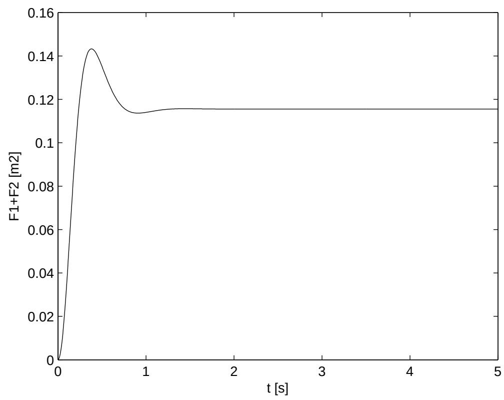


Figure 7: Formation function (time graph) for virtual structure with formation keeping behavior.

In Fig. 7 graph of formation function for formation of robots that executes the same task as in section 3 is shown. The values of control gains are: $K_P = 30$,

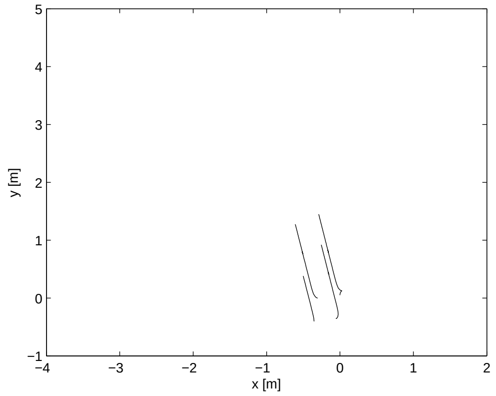


Figure 8: Formation of four robots tracks desired trajectory using virtual structure control method. Left-down robot fails after 1 second. Other robots slow down (in case without robot failure the formation goes to position around (-1.1, 4.5), like in case shown in Fig.3).

$K_F = 30$ and $K_V = 10$. It is very likely that in real application F_2 component of the formation function will be much greater due to time delay of sensor measurements and communication. Especially for large formation of robots disturbances of the motion of one robot will be transferred through the formation structure and affect motion of other robots. In this method worse accuracy is the cost paid for failure immunity.

In Fig. 8 simulation results for the case when one of robots fails is presented. Trajectory tracking and formation keeping are performed simultaneously. The priority of the goal depends on K_P/K_F ratio.

5 LEADER-FOLLOWER

In this section method based on leader-follower concept is presented. In most known leader-follower methods nonholonomic mobile robots are used. Robot called leader tracks a desired trajectory. Other robots keep desired separation and bearing to the leader. Dependencies between robots in large formations may be complex: some of them are followers and are followed by other robots at the same time.

Solution shown in this section is not typical leader-follower scheme. As methods described in previous sections this control is based on formation function.

Leader follower approach, in its simplest form, may be treated as a kind of virtual structure method. In Fig. 9 formation of four robots is shown. This control differs from virtual structure method only with reference point for formation in fact. In the pure virtual structure it is the point of the virtual trajectory. In leader-follower scheme it is position of leader robot.

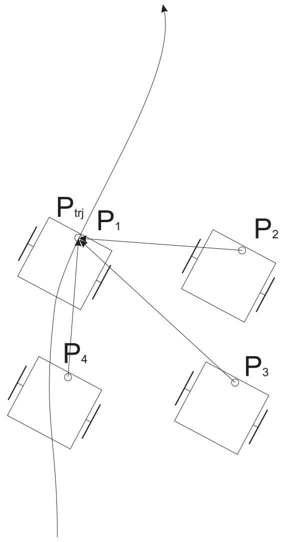


Figure 9: Leader-follower approach; one robot tracks desired formation, other robots keep relative position to the leader.

The formation function is given by the following equation:

$$F = \sum_{i=2}^{N} \|P_i - P_{1iof}\|^2, \qquad (12)$$

where $P_{1iof} = [x_{1iof} \;\; y_{1iof}]$ is offset vector between i-th robot leader (robot number 1).

Control of the i-th robot is given by Eq. (8).

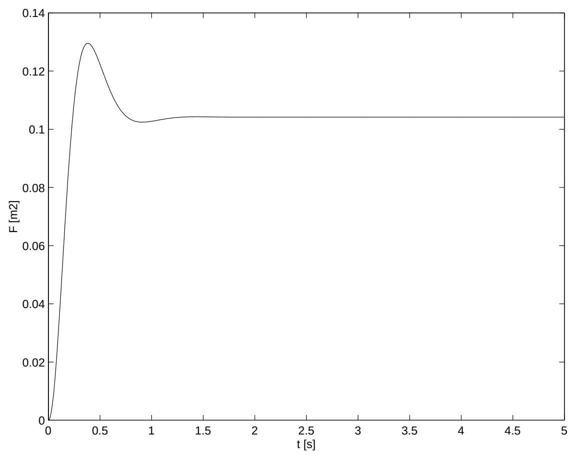


Figure 10: Formation function (time graph) for leader follower scheme.

In Fig. 10 the graph of formation function for four robots that execute the same task as in section 3 is shown.

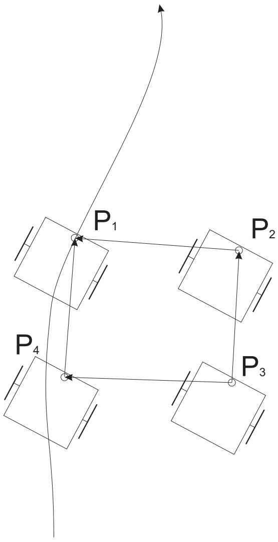


Figure 11: Leader-follower approach; one robot tracks desired formation, two other robots keep relative position to the leader, fourth robot keep relative position to followers.

In case shown in Fig. 11 the dependency between robots is constructed in a different way. The leader is followed by two robots, fourth robot keep relative position to followers and in fact they are leaders for this robot. Based on this concept very complex formations of robots with hierarchical structure may be built.

6 CONCLUSIONS

Three control methods for robot formation coordination were presented: virtual structure, virtual structure expanded with behavioral formation feedback and leader-follower scheme. Their accuracies were compared on basis of formation function graphs. Presented methods will be verified experimentally in our future work.

REFERENCES

B. J. Young, R. W. B. and Kelsey, J. M. (2001). Coordinated control of multiple robots using formation feedback. Technical report, MAGICC Lab.

Egerstedt, M. and Hu, X. (2001). Formation constrained multi-agent control. In *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*, pages 3961–3966, Seoul, Korea.

Esposito, J. M. and Kumar, V. (2000). A formalism for parallel composition of reactive and deliberative control objectives for mobile robots. Technical report, Mechanical Engineering and Applied Mechanics, University of Pennsylvania, Philadelphia.

J. R. Lawton, B. J. Y. and Beard, R. W. (2000). A decentralized approach to elementary formation maneuvers. In *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, pages 2728–2733, San Francisco, CA.

J. R. Lawton, B. Y. and Beard, R. (2002). A decentralized approach to elementary formation manouvers.

J. Spletzer, A. K. Das, R. F. C. J. T. and Kumar, V. (2001). Cooperative localization and control for multi-robot manipulation. Technical report, GRASP Laboratory - University of Pennsylvania, Philadelphia.

Kar-Han Tan, M. A. L. (1997). Virtual structures for high-precision cooperative mobile robotic control. Technical report, Computer Science Department, University of California, Los Angeles.

Kowalczyk, W. and Kozlowski, K. (2005). Artificial potential based motion control for a large formation of mobile robots. In *ICRA '2005*, Barcelona, Spain.

P. Ogren, M. E. n. X. H. (2002). A control lyapunov function approach to multi-agent coordination. In *IEEE Transactions on Robotics and Automation, vol. 18, no. 5*, pages 847–851.

R. Fierro, A. K. Das, V. K. and Ostrowski, J. P. (2001). Hybrid control of formation of robots. In *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*, pages 3672–3677, Seoul, Korea.

W. Kang, N. Xi, A. S. (2000). Formation control of autonomous agents in 3d workspace. In *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, pages 1755–1760, San Francisco, CA.

Yamaguchi, H. (1998). A cooperative hunting behavior by mobile robot troops. In *ICRA '98*, pages 3204–3209, Leuven, Belgium.

Yamaguchi, H. (1999). A cooperative hunting behavior by mobile-robot troops. In *The International Journal of Robotics Research, Vol, 18, No. 8*, pages 931–940.

APPLICATION OF A HUMAN FACTOR GUIDELINE TO SUPERVISORY CONTROL INTERFACE IMPROVEMENT

Pere Ponsa, Ramon Vilanova
Automatic Control Department, Technical University of Catalonia, Av. Victor Balaguer s/n
08800 Vilanova i la Geltrú, Spain
Telecomunication and Systems Engineering Department, Universitat Autónoma de Barcelona
ETSE, 08913 Bellaterra, Spain
pedro.ponsa@upc.edu, ramon.vilanova@uab.cat

Marta Díaz, Anton Gomà
4all-L@b Usability Laboratory, Technical University of Catalonia, Av. Victor Balaguer s/n
08800 Vilanova i la Geltrú, Spain
S.A.F. Sports Service Area, Universitat Autónoma de Barcelona
ETSE, 08913 Bellaterra, Spain
marta.diaz@upc.edu, anton.goma@upc.edu

Keywords: Supervisory control, user interface design, human computer interaction.

Abstract: In tasks of human supervision in industrial control room they are applied generic disciplines as the software engineering for the design of the computing interface and the human factors for the design of the control room layout. From the point of view of the human computer interaction, to these disciplines it is necessary to add the usability engineering and the cognitive ergonomics since they contribute rules for the user centered design. The main goal of this work is the application of a human factors guideline for supervisory control interface design in order to improve the efficiency of the human machine systems in automation. This communication presents the work developed to improve the Sports Service Area interface of the Universitat Autónoma de Barcelona.

1 INTRODUCTION

In recent years, control systems and the role of control room human operators have changed dramatically. Human operator activity has evolved from manually performing the process, to control system supervision. Today, the human operator requires an in-depth knowledge of the process that he/she is overseeing and the ability to make effective decisions within demanding constraints.

The increased complexity of industrial process control calls for a new methodological approach (for research and design purposes), which reproduces the essential components of current control systems: the environment, the task at hand and human operator activity (Samad and Weyrauch, 2000).

The complexity of industrial process supervision makes it necessary to supplement the Human Factors approach and the Human-Computer Interaction approach with a cross-disciplinary cooperation in order to integrate knowledge and methods from other fields, especially Cognitive Ergonomics, Automation and Artificial Intelligence (Granollers et. al., 2005), (Holstom, 2000) (Petersen, 2000). Our view is that complete control systems engineering must encompass all these approaches.

Ergonomics is concerned with the adaptation of technology to suit human operator need and ability so as to achieve effectiveness, efficiency and user/worker satisfaction and comfort (Cañas, 2004).

Supervisory control is the set of activities and techniques developed over a set of controllers (programmable logic controllers and industrial regulators) which ensures the fulfilling of control goals. One of the main goals is to prevent possible plant malfunctions that can lead to economical lose and/or result in damage (Petersen and May, 2006). For this reason, other fields of knowledge concerned with manufacturing systems performance – such as maintenance and industrial security – are complementary in the study of supervision systems.

In this paper a methodology for the creation of a human factor guideline for supervisory control interface design is proposed. In section 2 we present a checklist of indicators of the guideline called 'ergonomic guideline for supervisory control interface design' (GEDIS *Guia ergonómica para el diseño de interfaz de supervision* in Spanish version). The Sports Service Area project is described in section 3. The purpose is not to cover with detail the entire project but to give an idea of the different kind of topics that have been covered. In section 4, transition from the GEDIS model to Sports Service Area interface in control room is evaluated. In this section, a set of recommendations about graphical interface improvement are studied. Finally, conclusions and future research lines.

2 GEDIS GUIDELINE

The previous research on human interface design guidelines includes for example the standard ISO 11064 that establishes ergonomic principles for the evaluation of control centers (ISO, 2004), the Human Factors Design Standards HFDS of the Federal Aviation Administration of the United States (Federal Aviation Administration, 1996), the Human Interface Design Review Guidelines NUREG 0700 in nuclear power plants (Nuclear Regulatory Commission, 2002), the I-002 Safety and Automation Systems NORSOK about Norwegian petroleum industry (Norsok, 2006) and the Man Systems Integration Standard NASA-STD-3000 about manned space programs (Nasa, 1995).

An example of cognitive modelling in human computer interaction is the GOMS guideline, about goals, operators, methods and selection rules in usability analysis (Card et. al., 1983). In combination with Keystroke-Level Model KLM an interface can be studied, also task execution time and human efficiency can be studied too.

The GEDIS guide is a method that seeks to cover all the aspects of the interface design (Ponsa and Díaz, 2007). From the initial point of view of strategies for effective human-computer interaction

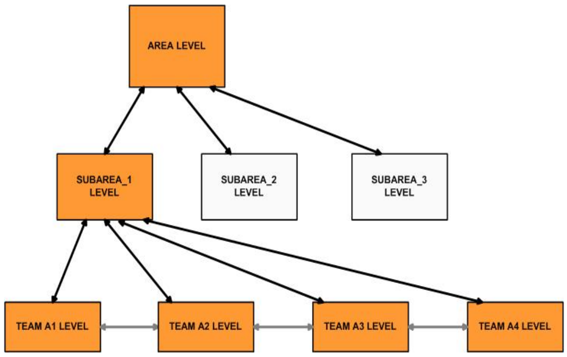


Figure 1: A typical cyclic network menu in supervisory control interface associated to navigation indicator.

applied to supervision tasks in industrial control room (Nimmo, 2004), (Schneiderman, 1997).

The GEDIS guide offers design recommendations in the moment to create the interface. Also, already offers recommendations of improvement of interfaces created. The GEDIS guide is composed of two parts: description of ten indicators and measure of ten indicators. The indicators have been defined from extracted concepts of other generic human factors guidelines, and for aspects of human interface design in human computer interaction.

The method to continue for the use of the GEDIS guide is: analyze the indicator, measure the indicator, obtain the global evaluation index and finally offer recommendations of improvement.

For the correct use of the GEDIS guide it is necessary the collaboration between the control room technical team and the human factor technician, since in some cases to analyze the indicator is necessary the expert's opinion.

2.1 Indicators List

The GEDIS guide consists of ten indicators that seek to cover all the aspects of the interface design in the supervisory control domain. The indicators are: architecture, distribution, navigation, color, text font, status of the devices, process values, graphs and tables, data-entry commands, and finally alarms. For example, the relationship between architecture and navigation indicators is illustrated in Fig. 1. The physical plant can separate in area, subarea, and team. In the same way, the interface presents four navigation levels. Fig. 1 shows a possible layout to locate all the connections between screens. The connection among screens is complex in a supervisory control interface. From the point of view

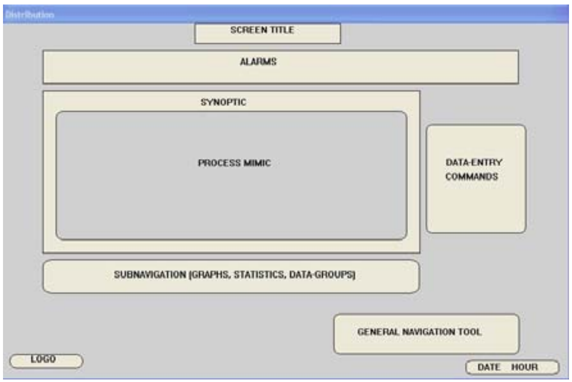


Figure 2: An example of object's layout inside the screen for the distribution indicator.

of human computer interaction, is a typical example of cyclic network menu.

Distribution indicator of Fig. 2 shows a possible layout to locate all the objects inside the screen. The objects homogeneous distribution allows us to maintain the interface coherence when user changes the screen. The secondary objects are located in screen areas that don't require the user's attention (enterprise logo, and date/hour information). The user should recognize the screen title and the general navigation tool to move among screens. The main objects are located in visible screen areas (alarms, data-entry commands, subnavigation tool, and synoptic objects). The user can surveillance the process evolution without acting (human out of the loop), or he can decide to introduce changes in the set point or in the controller's parameters (human in the loop) inside a faceplate window in the data-entry command object. The user should have special attention to the alarm indicators, which should be located in a clear way in the screen so that the user can recognize the situation (situation awareness).

3 SAF PROJECT

This section presents the development of the supervisory control system, with special emphasis on the interface features, for the Sports Service Area (SAF in Catalan version) of the Universitat Autonoma of Barcelona (UAB). This supervisory control system has been developed by a team of Computer Sciences Engineers with common design guidelines. Even some basic principles on ergonomics and interface design were taken into account; the GEDIS analysis will show existing weakness. An alternative presentation of the SAF project can be found in (Vilanova and Gomà, 2006).

Figure 3: Main window of the developed monitoring system with a global view of the Sports Service Area.

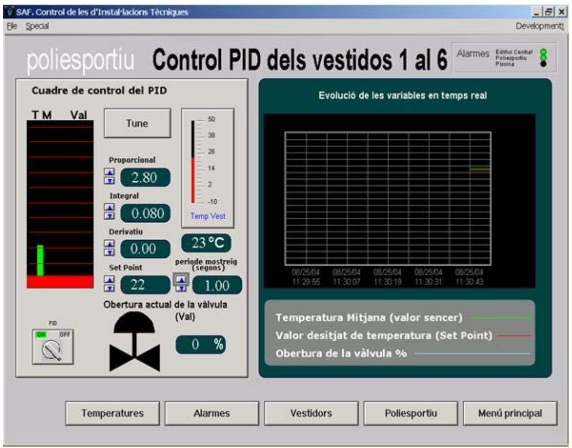


Figure 4: ISA-PID used to close the loops.

First of all, it is worth to know that the UAB is a campus based university with more than 40.000 inhabitants (students, academics, staff, etc.). In fact, this makes the University campus to behave like a city with some sort of facilities offered for their inhabitants. Among them, the Sports Service Area (SAF) is one of the largest and with more complex installations. It encompasses indoor as well as outdoor activities that run for more than 12h each day. Just to give an idea of the different installations that give support to the offered activities. We can find there: covered swimming pool, boulders, outdoor facilities for tennis, football, athletics, etc., indoor installations for fitness, basketball, aerobic, gym, etc. (Antsaklis et. al., 1999), (Astrom, 1999), (Kheir et. al., 1996).

Therefore, large complexes build up from different subsystems. Each one of these subsystems has to assure a quality of service each day. This fact introduces the need for good monitoring tools to help on this task. In addition, there is a hug number of automation and control problems (automated watering, temperature controls for water and indoor areas, lightning systems, ozone controlled system for water cleaning in covered swimming pools, etc.).

The SAF project has different automation levels: from field instrumentation and data collection, PLC programming and feedback loop configuration, to information integration on a SCADA system. The SAF project use PLC from different manufacturers (SIEMENS, GE-FANUC, Landys, and Mitsubishi). All the data has been integrated through implementing the corresponding supervisory control interface with Wonderware suite called In TOUCH. The basic communications use specific drivers to connect PLC with PC based control; the advanced communications use the standard OPC protocol.

An example: for indoor activities temperature control of both the SAF building and the water for the gym showers has been implemented. This means the students, in control room operator role, had to close some loops by using the ISA-PID present either in the PLC or in the software (see Fig. 4).

One important aspect of a monitoring system is how it deals with alarms. As this feature is a common feature, it should be incorporated in every part of the system according to the same rules. This way, in every SCADA window and alarm indicator has to be included that shows the human operator if an alarm is currently fired and can let you go directly to the main alarm window to process it.

Finally, design implementation and configuration of the In TOUCH based SCADA system has been done starting from zero. This allowed to think of a distributed application where from the different computers located either at the main SAF office or at the technical staff room the overall system can be accessed. In addition a special access, using terminal services, for the technical staff has been enabled so remote operation can also be done from outside the SAF installations.

4 SAF EVALUATION

The connection between SAF designer and GEDIS guideline designer is necessary to define a global evaluation of the SAF interface and can give a set of recommendations about graphical screen improvement (see Fig. 5).

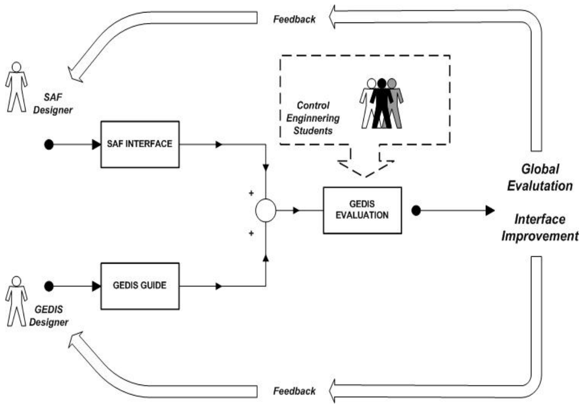


Figure 5: SAF interface evalutation with GEDIS guide method.

4.1 Evaluation

The evaluation expressed in quantitative numeric form or in qualitative format it seeks to promote the user's reflection that stuffs the GEDIS guide by way of questionnaire, so that it picks up the use experience that doesn't end up being verbalized in many occasions.

Each one of the indicators of the Table 1 and Table 2 can substructure in diverse subindicators. For example, the indicator Color can be detailed in: absence of non appropriate combinations (5), number of colors (5), blink absence (no alarm situation) (5), contrast screen versus graphical objects (3), relationship with text (3). For each subindicator it is recommended it is punctuated numerically in a scale from 1 to 5. In this example the number of subindicators of the indicator Color is J = 5 (see formula 1). The formula necessary to calculate the numeric value of each indicator is the formula 1.

$$Indicator = \frac{\sum_{j=1}^{J} w_j Subind_j}{\sum_{j=1}^{J} w_j} \qquad (1)$$

where, Subind= subindicator and w = weight.

The mean value that one obtains by the formula 1 with these numeric values is 4,2 . If it is rounded, the value is 4, so that to the indicator Color it is assigned the value 4 in this example, considering that each one of the subindicators has the same weight (w1 = w2… =wJ = 1).

Table 1: GEDIS guide indicators (part one).

Indicator name and Subindicator name	Numeric/qualitative range and SAF numeric value
Architecture	**1,7**
Map existence	[YES, NO] [5,0] 0
Number of levels le	[le<4, le>4] [5,0] 0
Division: plant, area, subarea, team	[a, m. na] [5,3,0] 5
Distribution	**3**
Model comparison	[a, m. na] [5,3,0] 3
Flow process	[clear, medium, no clear] [5.3,0] 3
Density	[a, m. na] [5,3,0] 3
Navigation	**3**
Relationship with architecture	[a, m. na] [5,3,0] 3
Navig. between screens	[a, m. na] [5,3,0] 3
Color	**5**
Absence of non appropriate combinations	[YES, NO] [5,0] 5
Color number c	[4<c<7, c>7] [5,0] 5
Blink absence (no alarm situation)	[YES, NO] [5,0] 5
Contrast screen versus graphical objects	[a, m. na] [5,3,0] 5
Relationship with text	[a, m. na] [5,3,0] 5
Text font	**3,2**
Font number f	[f<4, f>4] 5
Absence of small font (smaller 8)	[YES, NO] [5,0] 0
Absence of non appropriate combinations	[YES, NO] [5,0] 5
Abbreviation use	[a, m. na] [5,3,0] 3

where, a= appropriate, m=medium and na = no appropiate.

Each one of the indicators of the Table 1 is measured in a scale from 1 to 5. The human expert operator prepares in this point of concrete information on the indicator, so that it can already value the necessities of improvement. The values of the indicators can group so that the GEDIS guide offers the global evaluation of the interface and it can be compared with others interfaces.

The formula necessary to calculate the GEDIS guide global evalutation index is the formula 2.

$$Global\_evaluation = \frac{\sum_{i=1}^{10} p_i ind_i}{\sum_{i=1}^{10} p_i} \qquad (2)$$

where, ind = indicator and p = weight.

Table 2: GEDIS guide indicators (part two).

Indicator name and Subindicator name	Numeric/qualitative range and SAF numeric value
Status of the devices	**4**
Uniform icons and symbols	[a, m. na] [5,3,0] 3
Status team representativeness	[YES, NO] [5,0] 5
Process values	**3**
Visibility	[a, m. na] [5,3,0] 3
Location	[a, m. na] [5,3,0] 3
Graphs and tables	**4**
Format	[a, m. na] [5,3,0] 3
Visibility	[a, m. na] [5,3,0] 3
Location	[a, m. na] [5,3,0] 5
Grouping	[a, m. na] [5,3,0] 5
Data-entry commands	**3**
Visibility	[a, m. na] [5,3,0] 3
Usability	[a, m. na] [5,3,0] 3
Feedback	[a, m. na] [5,3,0] 3
Alarms	**3,8**
Visibility of alarm window	[a, m. na] [5,3,0] 3
Location	[a, m. na] [5,3,0] 3
Situation awareness	[YES, NO] [5,0] 5
Alarms grouping	[a, m. na] [5,3,0] 5
Information to the operator	[a, m. na] [5,3,0] 3

where, a= appropriate, m=medium and na = no appropiate.

In a first approach it has been considered the mean value among indicators expressed in the formula 2. That is to say, to each indicator it is assigned an identical weight (p1 = p2… =p10 = 1) although it will allow it in future studies to value the importance of some indicators above others. The global evaluation is expressed in a scale from 1 to 5. Assisting to the complexity of the systems of industrial supervision and the fact that an ineffective interface design can cause human error, the global evaluation of a supervision interface it should be located in an initial value of 3-4 and with the aid of GEDIS guide it is possible to propose measures of improvement to come closer at the 5.

4.2 Experimental Study

The experimental study is the evaluation of SAF interface with the collaboration of control engineering students from Technical University of Catalonia. From Vilanova i la Geltrú city, twenty five students monitoring SAF interface around three weeks. The students define a numeric value for each indicator and propose interface improvement.

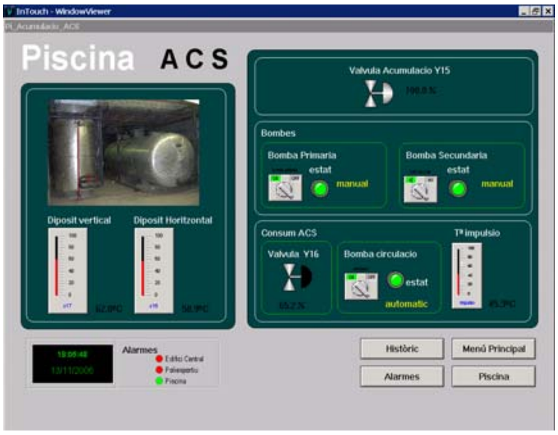


Figure 6: Original Piscina ACS screen.

Figure 8: Original Fronto and Rocodrom screen.

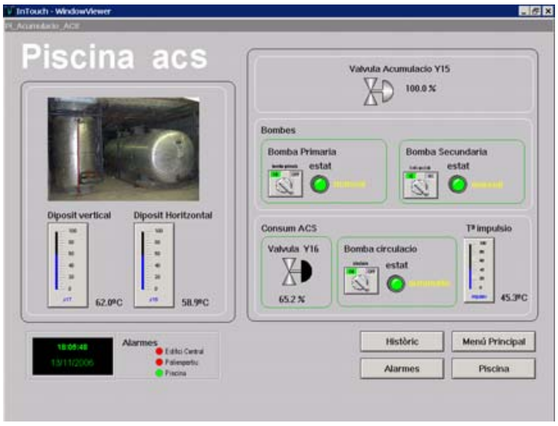


Figure 7: Piscina ACS revisited with changes in color indicator.

Figure 9: Fronto and Rocodrom revisited with changes in distribution indicator.

The SAF interface global evaluation is 3,4. The global evaluation is expressed in a scale from 1 to 5, so it is necessary to indicate SAF designer a set of important recommendations:

- revise the relationship between architecture, distribution and navigation indicators
- improve the feedback between interface and human operator in data-entry commands indicator
- improve the location of alarm indicator

With GEDIS guide is possible too to indicate SAF designer a set of important recommendations about graphical screen improvement. For example, the Piscina ACS screen can improve with a set of changes in color and text font indicators. Fig. 6 shows the original Piscina ACS screen and Fig. 7 shows revisited Piscina ACS screen.

A second example, the Fronton and Rocodrom screen can improve with a set of changes in distribution indicator.

Fig. 8 shows the original Fronton and Rocodrom screen and Fig. 9 shows revisited Fronton and Rocodrom screen.

5 CONCLUSIONS

In tasks of human supervision in industrial control room is habitual that an external engineer, - by means of the commercial programs Supervisory Control and Data Acquisition SCADA -, take charge of designing the supervision interfaces in function to the knowledge on the physical plant and the group of physical-chemical processes contributed by the process engineers.

Although standards exist about security in the human machine systems that impact in aspects of physical ergonomics, interface design by means of rules of style, it is remarkable the absence of the design of interactive systems centered in the user where the engineering usability and the cognitive ergonomics can contribute significant improvements (Nielsen, 1993).

The GEDIS guide is an approach that tries to fill a methodological hole that joins the efforts of the systems engineering and the human factors for the improvement of the effectiveness of the human-machine system in industrial control room.

The application of the GEDIS guide to the study of cases contributes among other details the measure in form of indicators of aspects of interface design, the recommendation of changes for the improvement of the interface, and a global evaluation index that allows to quantify the current state of the interface regarding the future state after applying the correct measures.

The studied case presented shows a Spanish academic application, but with the same characteristics of an industrial project. With the GEDIS guide approach it's possible to perceive diverse anomalies and to propose improvements in the interface design.

Another current study with the GEDIS guide is the analysis of a sugar mill interface. The Sugar Technology Center (CTA) in Spain has been developed a training simulator to modeling and simulating the production process and the human operators' supervisory tasks. The simulator developed in this center is an example of full scale simulator, a type of simulator that reproduces the whole operating environment (Merino et. al., 2005). This simulator emulates the control room of a sugar mill. A series of object oriented modelling library tools are used to create each part of the sugar mill: diffusion, evaporation, purification, sugar room, boilers, dryer, and liquor storage.

In these moments the 4all-L@b Usability Laboratory of the Technical University of Catalonia is analyzing the GEDIS guide to simplify the number of indicators of the guide, to improve the evaluation method, and to promote the use of the guide inside the cycle of life of the software engineering, in this case in the early phases of the supervisory control interface design.

ACKNOWLEDGEMENTS

Participation of the second author is supported in part by the Spanish CICYT program under contract DPI2004-06393.

REFERENCES

Antsaklis, P., Basar, T., DeCarlo, R., McClamroch,N.H., Spong, M., Yurkovich, S., 1999. *Report on the nsf/css workshop on new directions in control engineering education.* IEEE Control Systems Magazine, (10), pp 53-58.

Astrom, K.J., 1999. *Automatic control: the hidden technology*. P.M. Frank ed.. *Advances in Control: Highlights of ECC-99*, pp. 1-28.

Cañas, J.J., 2004. *Personas y máquinas*. Editorial Pirámide, Colección Psicología

Card, S., Moran, T.P., Newell, A., 1983. *The psychology of human computer interaction.* Lawrence Erlbaum Associates, Inc.

Federal Aviation Administration, 1996. *Human factors design guide for acquisition of commercial-off-the-shelf subsystems, non-developmental items, and developmental systems (DOT/FAA/CT-96/01)*. Atlantic City International Airport, DOT/FAA Technical Center

Granollers, T., Lorés, J., Cañas, J.J., 2005. *Diseño de sistemas interactivos centrados en el usuario*. Editorial UOC, Colección Informática, No 43

Holstom, C., 2000. *Human factors and control centre issues. What lessons we have learned*. Institute for Energy Tecnology, OECD Halden Reactor Project

ISO International Organization for Standarization, 2004. *Ergonomic design of control centres, parts I, II, III, IV*. In URL: http://www.iso.org

Kheir, N.A., Astrom, K.J., Auslander, D., Cheok, K.C., Franklin. G.F., Masten, M., Rabins, M., 1996. *Control systems engineering education*. Automatica 32 (2), pp. 147-166.

Merino, A., Alves, R., Acebes. L.F., 2005. *A training Simulator fot the evaporation section of a beet sugar production.* European Simulation Multiconference, ESM-05, Oporto, Portugal.

NASA, 1995. *Man system integration standards*. NASA-STD-3000, In URL: http://msis.jsc.nasa.gov/

Nielsen, J., 1993. *Usability engineering.* Academic Press, Boston.

Nimmo, I., 2004. *Designing control rooms for humans*. Control Magazine

Norsok Standard, 2006. *I-002 Safety automation system*. Norwegian Technology Centre Oscarsgt. 20, Postbox 7072 Majorstua N-0306 Oslo. In URL: http://www.olf.no y http://trends.risoe.dk/detail-organisation.php?id=52#corpus

Petersern, J., 2000. *Knowledge based support for situation assessment in human supervisory control.* Ph.D. Thesis. Department of Automation. Technical University of Denmark.

Petersen, J., May, M., 2006. *Scale transformations and information presentation in supervisory control.* International Journal of Human-Computer Studies, Vol 64, 5, May, pp. 405-419.

Ponsa,P., Díaz, M., 2007. *Creation of an ergonomic guideline for supervisory control interface design*. 12 th International Conference on Human-Computer

Interaction. July, Beijing, P.R. China. In URL: http://www.hcii2007.org

Samad, T., Weyrauch, W., 2000. *Automation, control and complexity*. John Wiley and Sons

Schneiderman, B., 1997. *Designing the user interface. Strategies for effective human-computer interaction*. Addison-Wesley, third edition

U.S. Nuclear Regulatory Commission, 2002. *NUREG-0700, Human-system interface design review guidelines*. Office of Nuclear Regulatory Research, Washington DC 20555-0001. In URL: http://www.nrc.gov/reading-rm/doc-collections/nuregs/staff/sr0700/nureg700.pdf

Vilanova, R., Gomà, A., 2006. *A collaborative experience to show how the University can play the industrial role*. 7th IFAC Symposium on Advances in Control Education, June, Madrid, Spain. URL: http:www.dia.uned/ace2006

MULTI-RESOLUTION BLOCK MATCHING MOTION ESTIMATION WITH DEFORMATION HANDLING USING GENETIC ALGORITHMS FOR OBJECT TRACKING APPLICATIONS

Harish Bhaskar and Helmut Bez
Loughborough University, U.K.
H.Bhaskar@lboro.ac.uk, H.E.Bez@lboro.ac.uk

Keywords: Block motion estimation, affine, deformation handling, genetic algorithms, multi-resolution, object tracking.

Abstract: Motion Estimation is a popular technique for computing the displacement vectors between objects or attributes between images captured at subsequent time stamps. Block matching is a well known technique of motion estimation that has been successfully applied to several applications such as video coding, compression and object tracking. One of the major limitations of the algorithm is its ability to cope with deformation of objects or image attributes within the image. In this paper we present a novel scheme for block matching that combines genetic algorithms with affine transformations to accurate match blocks. The model is adapted into a multi-resolution framework and is applied to object tracking. A detailed analysis of the model alongside critical results illustrating its performance on several synthetic and real-time datasets is presented.

1 INTRODUCTION

Motion estimation techniques aim at deducing displacement vectors for objects or image attributes between two consecutive frames (A. Gyaourova and Cheung, 2003). The main idea behind block matching motion estimation strategies is to divide the image frame into blocks and match blocks between successive frames within a search window using specific search techniques (A.Barjatya, 2005). It is clear that the two distinct phases that make up any block matching method is block partitioning and block searching. The block partitioning scheme is concerned with dividing the original image frame into non-overlapping regions. Block partitioning can be performed using the fixed size or variable size methods (C-C.Chang, 2006) (F.J.Ferri and J.Soret, 1998). The block search mechanism is the process of locating the block in the destination frame that best matches the block in the anchor frame using a specific matching criterion (Turaga and M.Alkanhal, 1998).

Different models have been developed in literature to accomplish robust motion estimation using block based techniques. (C-W.Ting and L-M.Po, 2004) propose the use of different search schemes with fixed and variable block partitioning methods to accomplish robust estimation. In a similar study by (M.Wagner and D.Saupe, 2000), a quad-tree block motion estimation scheme is proposed. Other methods of variable block matching have also been proposed, particularly in the form of polygon approximation, mesh based (Y.Wang and O.Lee, 1996) and binary trees. Another class of block matching methods that have recently used for deformation handling particularly in applications of object tracking is the deformable block matching (O.Lee and Y.Wang, 1995). In a study by (J.H.Velduis and G.W.Brodland, 1999), deformable block matching has been adapted for use in tracking cell particles. A bilinear transformation is used with block matching to handle deformation. In the context of deformable models,triangular or mesh based block decomposition is much popular (Y.Wang and A.Vetro, 1996). The idea behind these schemes is to partition image frames using techniques of finite element analysis, triangulation (M.Yazdi and A.Zaccarin, 1997), mesh grid etc. and employ deformable block matching of vertex points to handle complex motion changes during motion estimation. In the context of mesh based methods, a nodal based scheme for block matching is also popular. According to the nodal scheme, mesh is generated such that nodes lies across object boundaries and a simple search of linear motion of these nodal position from the anchor frame to the destination frame will be able to suffice deformation (O.Lee and Y.Wang, 1995). In this paper, we shall highlight a framework that integrates a vector quantization based block partitioning method to an genetic algorithm based search scheme with affine parametrization to accomplish robust, accurate motion estimation with deformation handling. The model is built on a multi-resolution platform with performance feedback.

2 PROPOSED MODEL

The proposed model constitutes of different phases. The first phase is the multi-resolution platform that the framework is based on. The platform combines a scale space representation of data with a multi-resolution level analysis. A multi-resolution model aims at capturing a wide range of levels of detail of an image and can in-turn be used to reconstruct any one of those levels on demand. The distinction between different layers of an image is determined by the resolution. A simple mechanism of tuning the resolution can add finer details to coarser descriptions providing a better approximation of the original image. Mathematically, we can represent the above analysis in the following way. If the resolution is represented using λ, then the initial level is associated with $\lambda = 0$ is 1 and that with any arbitrary resolution λ is $\frac{1}{2^\lambda}$. If f_λ is the image at resolution λ, then at resolution λ+1,

$$f_{\lambda+1} = f_\lambda + \Gamma_\lambda \tag{1}$$

where Γ_λ is the details at resolution λ. In contrast, the scale space representation of data deals with representing images in such a way that the spatial-frequency localizations are simultaneously preserved. This is achieved by decomposing images into a set of spatial-frequency component images. Scale space theory, therefore, deals with handling image structures at different scale such that the original image can be embedded into a one-parameter family of derived component images thereby allowing fine-scale structures to be successively suppressed.Mathematically, to accomplish the above, a simple operation of convolution can be used. However, it is important to note that the overhead of using the convolution operator is kept low. For any given image $I(x, y)$, its linear scale space representation is composed of components $L_\vartheta(x, y)$ defined as a convolution operator of the image $I(x, y)$ and a Gaussian kernel of the form:

$$G_\vartheta(x, y) = \frac{1}{2\pi\vartheta} e^{-\frac{x^2+y^2}{2\vartheta}} \tag{2}$$

, such that

$$L_\vartheta(x, y) = G_\vartheta(x, y) * I(x, y) \tag{3}$$

where $\vartheta = \sigma^2$ is the variance of the Gaussian. Performance based feedback automates the selection of relevant resolution and scale for any particular frame pair. A brief algorithm describing the process is as follows.

- Initialize the resolutions $\lambda_{[1:q]}$ to $[0, 1, 2, ..., q]$ and scales $\vartheta_{[1:q]}$ to $[1, 2, 3, ..., q + 1]$ for any value of q (4 chosen for this experiment).
- Select the median of resolutions as the initial starting resolution and scale. The median is 2 in our experiments and the chosen values of (λ, ϑ) are $(2, 3)$
- Input at any time instant t, two successive frame pairs of a video sequence, (f_t, f_{t+1}).
- Re-sample the images f_t and f_{t+1} into the selected resolution using bi-cubic interpolation
- Convolve the image at selected scale (in matching positions with the resolution) with a Gaussian kernel to obtain a filtered output $(G_\vartheta * f_t, G_\vartheta * f_{t+1})$
- Perform Motion Estimation of these input images at this scale-resolution using the motion estimation algorithm specified in the subsection below and reconstruct the target frame using the estimated motion parameters.
- Evaluate the performance of the model using the metrics: PSNR, Entropy and Time as in (H.Bhaskar and S.Singh, 2006)
- If the frame pair processed is (f_t, f_{t+1}) at $t = 1$ then automatically slide up to a higher resolution and repeat process by incrementing t. Otherwise, if $t > 1$ then if $PSNR_t > PSNR_{t-1}$ then slide down to lower resolution - scale otherwise slide up to higher resolution - scale combination.
- Repeat the process for all frame pairs

The second phase of the algorithm deals with motion estimation. For the purpose of motion estimation we extend the technique of deformable block matching that combines the process of block partitioning, block search and motion modeling. A vector quantization based block partitioning scheme is combined with a genetic algorithm based search method for robust motion estimation (H.Bhaskar and S.Singh, 2006). We extend the basic model in such a way that block deformation is handled using a combined genetic algorithm affine motion model.

The block partitioning phase remains unchanged while the genetic algorithm based block search scheme is altered to include the affine transformations. In the subsection below, a detailed algorithm of the modified block search scheme based on genetic algorithm and affine transforms is presented.

2.1 Vector Quantization Based Block Partitioning

The vector quantization scheme for block partitioning illustrated in (H.Bhaskar and S.Singh, 2006) has been used in the proposed deformable block matching. It is important to realize that the image frames f_t and f_{t+1} that is input to this stage of the algorithm refers to the filtered output of the previous stage. According to the vector quantization scheme, image frames are partitioned based on the information content present within them. The model separates regions of interest based on clustering and places a boundary separating these regions. For this, the vector quantization mechanism uses the gray level feature attributes for separating different image regions and the center of mass of different intersection configurations is employed to deduce the best partition suitable for the image frames.

2.2 Affine-Genetic Algorithm Motion Model

The idea behind the genetic algorithm affine motion model combination is to use the affine transformation equation on every block during fitness function evaluation. The algorithm for the block search scheme is as follows.

The genetic algorithm based block matching algorithm described below is used to match the centroid of any block from the partitioned structure of frame f_t to its successive frame f_{t+1} at different angles theta and parameters shear and scale. The inputs to the genetic algorithm are the block b_t and the centroid (x_c, y_c) of the block.

- Parameter Initialization: The variable parameters of the genetic algorithm will be the genes in the chromosomes. In our experiments they will be the the pixel displacement value in x and y directions, the angle theta of the input block, the shear factor s and scale (r_x, r_y) are encoded as the chromosome $(T_x, T_y, \theta, s, r_x, r_y)$. The translation, rotation and scale parameters of the model are initialized using the phase correlation and log-polar

Figure 1: Phase Correlation.

transforms. This speeds up the genetic algorithm search scheme and also increases the accuracy of estimation.

– Translation parameters using phase correlation: The phase correlation technique is a frequency domain approach to determine the translative movement between two consecutive images. A simple algorithm illustrating the process of determining an approximate translative motion characteristics between two images is as follows.

∗ Consider the input block b_t and its corresponding block at the successive frame b_{t+1}

∗ Apply a window function to remove edge effects from the block images

∗ Apply a 2D Fourier transform to the images and produce $F_t = \Psi(b_t)$ and $F_{b+1} = \Psi(b_{t+1})$; where ψ is the Fourier operator.

∗ Compute the complex conjugate of F_{t+1}, multiply the Fourier transforms element-wise and normalize to produce a normalized cross power spectrum NPS using

$$NPS = \frac{F_t F_{t+1}^*}{|F_t F_{t+1}^*|} \quad (4)$$

∗ Apply inverse Fourier transform on the normalized power spectrum to obtain $PS = \psi^{-1}(NPS)$; where ψ^{-1} is the inverse Fourier operator.

∗ Determine the peak as the the translative coordinates using

$$(\Delta x, \Delta y) = argmax(PS) \quad (5)$$

∗ An illustration describing the process of phase correlation using a sample image is as shown in Figure 1.

– Rotation and Scale using Log-Polar Transforms: The log-polar transform is a conformal mapping of points on cartesian plane to points on the log-polar plane. The transformation can accommodate an arbitrary rotations and a range of scale changes. If an block image in the cartesian plan is represented using $b(x, y)$, then the log polar transform of the block image with origin O at location (x_o, y_o) is

Figure 2: Log Polar Transform.

$$b^*(\psi, \phi) = b\xi(x, y) \qquad (6)$$

where,
$\psi = Mlog(r + \alpha)$, α is any constant
$r = \sqrt{(x - x_o)^2 + (y - y_o)^2}$ and
$\phi = \tan^{-1} \frac{y - y_o}{x - x_o}$

In order to determine the approximate values of rotation and scale using the log-polar transforms, we convert the image frames into the log polar domain and then use phase correlation between the log-polar images to identify the rotation and scale parameters as in Figure 2.

- Population Initialization: A population P of these n chromosomes representing $(T_x, T_y, \theta, s, r_x, r_y)$ 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.
  - The values of pre-initialized parameters such as translational, rotational and scale are generated within a small range of their initial value.
- 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), rotation angle θ, shear factor s and scale r_x, r_y using,
  $$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & s & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_x & 0 & 0 \\ 0 & r_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$
  $$\begin{bmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & T_x \\ 0 & 1 & T_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$
  - 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).

$$MAD = \frac{1}{hw} \sum_{i=1}^{h} \sum_{j=1}^{w} \left| b_t(i,j) - b^*_{t+1}(i,j) \right| \qquad (7)$$

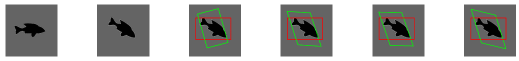


Figure 3: 2D Deformable Block Matching.

- 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.
- Cross-Over: All chromosomes n_{cr} that are chosen for cross-over are taken into the next generation after swapping one or more random genes between every successive chromosome.
- Mutation: All Chromosomes n_{mu} chosen for mutation are replaced with uniformly distributed random values for centroid, angle, shear, scale and squeeze.
- Termination: Three termination criterion are specified in the proposed model. Check if any condition is satisfied, otherwise iterate until termination.
  - Zero Error: If a chromosome returned an error value zero through fitness evaluation, Or
  - Maximum Generations: If the number of generations (i.e. process loops) exceeds a predefined threshold, Or
  - Stall Generations: If the number of stall generations (i.e. process loops where there is no change in the fitness values) exceeds a predefined threshold.

3 RESULTS AND ANALYSIS

Detailed results and analysis of the proposed model is presented in this section of the paper. On the second part of this section we demonstrate how the motion estimation scheme is adapted to object tracking applications.

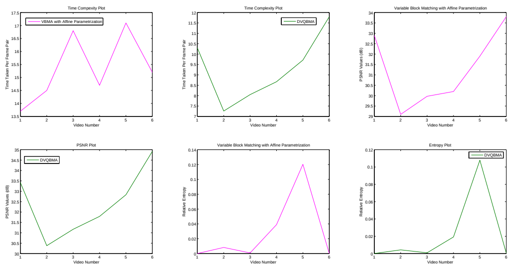


Figure 4: Performance Comparison of Proposed Model to Baselines.

3.1 Performance Evaluation of Motion Estimation

In Figure 3 we illustrate the stages of the proposed block matching scheme. The first and the second images illustrate the original frame and the transformed frame of a sample synthetic image. Through the other images we illustrate how the genetic algorithm is used to identify the optimal motion parameters. Different configurations are evolved through increasing generations getting the solution closer to optimal. The red block represents the objects original position in the anchor frame and the green boundary specifies the location of the block during block searching using genetic algorithm at different generations. To further affirm the performance of the model on different real time datasets, we perform experiments of the model on 6 different video data each containing around 40 frames. The averaged performances on each videos are measured using time, relative entropy and PSNR metrics and compared to the baseline model in Figure 4. The baseline model uses affine parametrization with other search schemes on a variable block partitioned data. We also compare the proposed model against the original block matching model that does not handle deformation and a rotation invariant model.

It is very evident that the averaged time complexity of the proposed motion estimation mechanism that handles deformation still does not match the requirements of real-time. However, with a multi-resolution optimization approach it might well be possible to improve the time efficiency. The results compare well with the quad-tree block matching mechanism with affine parametrization. There is a clear advantage in using the proposed strategy for deformation handling than an extension to any other variable block partitioning scheme with sub-optimal search. The quality of motion estimation is recorded and compared in the graphs. It can be observed that there is clear improvement in the quality of motion estimation when deformation of objects is handled during motion estimation. In comparison to the baseline model, there is clear increase of about 2dB in the PSNR values. A clear increase in the PSNR values can be noted during the progressive improvements in the model from the basic framework to the rotation invariant model and finally to the deformation handling model. This clearly indicates how useful deformation handling is during motion estimation. A very similar trend can also be visualized between different models when compared against the performance metric of relative entropy. The reconstructions made from the deformation handling model match closer to the expected outcome of the image frame. This highlights the accuracy and robustness of the strategy in accomplishing motion estimation. In comparison to the baseline model there is a clear improvement in the values of relative entropy.

3.2 Object Tracking Applications

In this section we describe how the motion estimation mechanism above can be adapted to object tracking applications and also analyze how the efficiency of motion estimation influences the quality of object tracking. We have extended the model for application in object tracking through simple clustering of features characteristics including motion information. To use the proposed model into object tracking motion vectors are clustered such that the moving group of blocks possessing similar motion and feature characteristics will form the object of interest. Trajectories are plotted using the center of mass location of the blocks that constitute the objects. We have tested the approach on a number of different datasets. We have displayed the results of the model on some of them. Figure 5 illustrates the motion trajectory (represented using red dots) of a single/multiple object tracked over different time stamps. As the model does not perform object segmentation, produces a number of small unwanted trajectories that have been removed through manually entered semantic information. The semantic information can be of the form of velocity information of the object in motion, color of the moving objects etc. We have in our experiments displayed the motion trajectory of the group of blocks that have been tracked longest on the image sequences. Generally in any scene this information corresponds to the object of interest. The first two images are the trajectories of the proposed model and a polygon shape feature based nearest neighbor tracking scheme proposed in (H.Bhaskar and S.Singh, 2005). As it can be clearly observed the trajectory of the baseline model

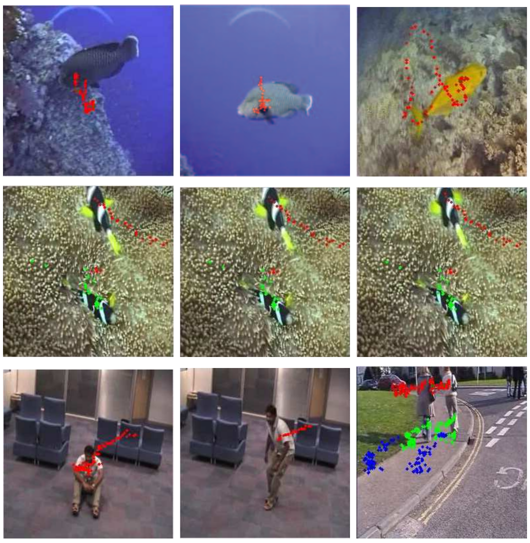


Figure 5: Object Trajectory of Sample Video Sequences.

disintegrates once the object has complex deformation whereas the proposed scheme continues to handle deformation reliably for the entire video. The main reason to this is that the model relies on image segmentation and polygon based shape approximation. The second group of 3 images illustrate the tracking of multiple objects in an image sequence. This sequence is also an example of a very noisy sequence with most of its background moving as well. Finally, examples of human tracking are also illustrated. In the first, we have extracted the trajectory of the body of the person moving and displayed it. The model actually produces different trajectories for moving parts of hands, legs, face etc. as in the next image. The second image is the output of the shape feature based baseline model. Again the technique fails to track objects immediately after a complex deformation is noticed.

4 CONCLUSION

In this paper we presented a novel deformation handling mechanism for block matching based on genetic algorithm that can be extended for use in object tracking applications. The model combines the vector quantization based variable block partitioning and applies an affine based genetic algorithm matching scheme for block matching. We have also presented results on several real time datasets to illustrate the proof of concept. Analysis of the results on the model has proved that the model is robust and reliable for tracking deformational changes in objects in video sequences.

REFERENCES

A. Gyaourova, C. K. and Cheung, S.-C. (2003). Block matching for object tracking. Technical report, Lawrence Livermore Nation Laboratory.

A.Barjatya (2005). Block matching algorithms for motion estimation. Technical report.

C-C.Chang, L-L.Chen, T.-S. (2006). Multi-resolution based motion estimation for object tracking using genetic algorithm. In *VIE Conference*.

C-W.Ting, W.-H. and L-M.Po (2004). Fast block matching motion estimation by recent-biased search for multiple reference frames. In *International Conference on Image Processing (ICIP)*.

F.J.Ferri, J. J. and J.Soret (1998). Variable-size block matching algorithm for motion estimation using a perceptual-based splitting criterion. In *International Conference on Pattern Recognition (ICPR)*, page 286.

H.Bhaskar and S.Singh (2005). Multiple particle tracking in live cell imaging using green fluorescent protein (gfp) tagged videos. In *International Conference on Advances in Patter Recognition*, pages 792–803.

H.Bhaskar, R. and S.Singh (2006). A novel vector quantization based block matching strategy using genetic algorithm search. Submitted to Pattern Recognition Letters.

J.H.Velduis and G.W.Brodland (1999). A deformable block-matching algorithm for tracking epithelial cells. In *Image and Vision Computing*, pages 905–911.

M.Wagner and D.Saupe (2000). Video coding with quadtrees and adaptive vector quantization. In *Proceedings EUSIPCO*.

M.Yazdi and A.Zaccarin (1997). Interframe coding using deformable triangles of variable size. In *International Conference on Image Processing*, page 456.

O.Lee and Y.Wang (1995). Motion compensated prediction using nodal based deformable block matching. In *Visual Communications and Image Representation*, pages 26–34.

Turaga, D. and M.Alkanhal (1998). Search algorithms for block-matching in motion estimation.

Y.Wang and O.Lee (1996). Use of 2d deformable mesh structures for video compression, part i - the synthesis problem: Mesh based function approximation and mapping. In *IEEE Trans. Circuits and Systems on Video Technology*, pages 636–646.

Y.Wang, O. and A.Vetro (1996). Use of 2d deformable mesh structures for video compression, part ii - the analysis problem and a region-based coder employing an active mesh representation. In *IBID*, pages 647–659.

GEOMETRIC ADVANCED TECHNIQUES FOR ROBOT GRASPING USING STEREOSCOPIC VISION

Julio Zamora-Esquivel and Eduardo Bayro-Corrochano
Department of Electrical Engineering and Computer Science
CINVESTAV, unidad Guadalajara. Jalisco, Mexico
jzamora@gdl.cinvestav.mx, edb@gdl.cinvestav.mx

Keywords: Conformal Geometry, Kinematics, Grasping, Tracking.

Abstract: In this paper the authors propose geometric techniques to deal with the problem of grasping objects relying on their mathematical models. For that we use the geometric algebra framework to formulate the kinematics of a three finger robotic hand. Our main objective is by formulating a kinematic control law to close the loop between perception and actions. This allows us to perform a smooth visually guided object grasping action.

1 INTRODUCTION

In this work the authors show how to obtain a feasible grasping strategy based on the mathematical model of the object and the manipulator. In order to close the loop between perception and action we estimate the pose of the object and the robot hand. A control law is also proposed using the mechanical Jacobian matrix computed using the lines of the axis of the Barrett hand. Conformal geometric algebra has been used within this work instead of the projective approach (Ruf, 2000) due to the advantages which are provided by this mathematical framework in the process of modeling of the mechanical structures like the one of the Barrett Hand.

In our approach first we formulate the inverse kinematics of the robot hand and analyze the object models in order to identify the grasping constraints. This takes into account suitable contact points between object and robot hand. Finally a control law to close the perception and action loop is proposed. In the experimental analyzes we present a variety of real grasping situations.

2 GEOMETRIC ALGEBRA

Let $\mathcal{G}_n$ denote the geometric algebra of n-dimensions, this is a graded linear space. As well as vector addition and scalar multiplication we have a non-commutative product which is associative and distributive over addition – this is the *geometric* or *Clifford product*.

The inner product of two vectors is the standard *scalar* product and produces a scalar. The outer or wedge product of two vectors is a new quantity which we call a *bivector*. Thus, $b \wedge a$ will have the opposite orientation making the wedge product anti-commutative. The outer product is immediately generalizable to higher dimensions – for example, $(a \wedge b) \wedge c$, a *trivector*, is interpreted as the oriented volume formed by sweeping the area $a \wedge b$ along vector c. The outer product of k vectors is a k-vector or k-blade, and such a quantity is said to have *grade* k. A *multivector* (linear combination of objects of different type) is *homogeneous* if it contains terms of only a single grade.

We will specify a geometric algebra $\mathcal{G}_n$ of the n dimensional space by $\mathcal{G}_{p,q,r}$, where p, q and r stand for the number of basis vector which squares to 1, -1 and 0 respectively and fulfill $n = p+q+r$.

We will use e_i to denote the vector basis i. In a Geometric algebra $G_{p,q,r}$, the geometric product of two basis vector is defined as

$$e_i e_j = \begin{cases} 1 & for \quad i = j \in 1, \cdots, p \\ -1 & for \quad i = j \in p+1, \cdots, p+q \\ 0 & for \quad i = j \in p+q+1, \cdots, p+q+r. \\ e_i \wedge e_j & for \quad i \neq j \end{cases}$$

This leads to a basis for the entire algebra:

$$\{1\}, \{e_i\}, \{e_i \wedge e_j\}, \{e_i \wedge e_j \wedge e_k\}, \ldots, \{e_1 \wedge e_2 \wedge \ldots \wedge e_n\} \qquad (1)$$

Any multivector can be expressed in terms of this basis.

3 CONFORMAL GEOMETRY

Geometric algebra $G_{4,1}$ can be used to treat conformal geometry in a very elegant way. To see how this is possible, we follow the same formulation presented in (H. Li, 2001) and show how the Euclidean vector space $\mathbb{R}^3$ is represented in $\mathbb{R}^{4,1}$. This space has an orthonormal vector basis given by $\{e_i\}$ and $e_{ij} = e_i \wedge e_j$ are bivectorial basis and e_{23}, e_{31} and e_{12} correspond to the Hamilton basis. The unit Euclidean pseudo-scalar $I_e := e_1 \wedge e_2 \wedge e_3$, a pseudo-scalar $I_c := I_e E$ and the bivector $E := e_4 \wedge e_5 = e_4 e_5$ are used for computing the inverse and duals of multivectors.

3.1 The Stereographic Projection

The conformal geometry is related to a stereographic projection in Euclidean space. A stereographic projection is a mapping taking points lying on a hypersphere to points lying on a hyperplane. In this case, the projection plane passes through the equator and the sphere is centered at the origin. To make a projection, a line is drawn from the north pole to each point on the sphere and the intersection of this line with the projection plane constitutes the stereographic projection.

For simplicity, we will illustrate the equivalence between stereographic projections and conformal geometric algebra in $\mathbb{R}^1$. We will be working in $\mathbb{R}^{2,1}$ with the basis vectors $\{e_1, e_4, e_5\}$ having the usual properties. The projection plane will be the x-axis and the sphere will be a circle centered at the origin with unitary radius.

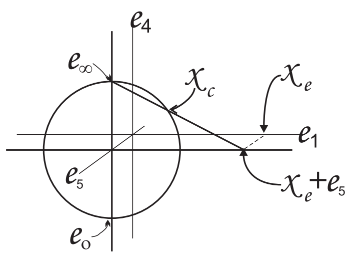


Figure 1: Stereographic projection for 1-D.

Given a scalar x_e representing a point on the x-axis, we wish to find the point x_c lying on the circle that projects to it (see Figure 1). The equation of the line passing through the north pole and x_e is given by $f(x) = -\frac{1}{x_e}x + 1$ and the equation of the circle $x^2 + f(x)^2 = 1$. Substituting the equation of the line on the circle, we get the point of intersection x_c, which can be represented in homogeneous coordinates as the vector

$$x_c = 2\frac{x_e}{x_e^2+1}e_1 + \frac{x_e^2-1}{x_e^2+1}e_4 + e_5. \qquad (2)$$

From (2) we can infer the coordinates on the circle for the point at infinity as

$$e_\infty = \lim_{x_e \to \infty} \{x_c\} = e_4 + e_5, \qquad (3)$$

$$e_o = \frac{1}{2} \lim_{x_e \to 0} \{x_c\} = \frac{1}{2}(-e_4 + e_5), \qquad (4)$$

Note that (2) can be rewritten to

$$x_c = x_e + \frac{1}{2}x_e^2 e_\infty + e_o, \qquad (5)$$

3.2 Spheres and Planes

The equation of a sphere of radius ρ centered at point $p_e \in \mathbb{R}^n$ can be written as $(x_e - p_e)^2 = \rho^2$. Since $x_c \cdot y_c = -\frac{1}{2}(\mathbf{x}_e - \mathbf{y}_e)^2$ and $x_c \cdot e_\infty = -1$ we can factor the expression above to

$$x_c \cdot (p_c - \frac{1}{2}\rho^2 e_\infty) = 0. \qquad (6)$$

Which finally yields the simplified equation for the sphere as $s = p_c - \frac{1}{2}\rho^2 e_\infty$. Alternatively, the dual of the sphere is represented as 4-vector $s^* = sI_c$. The sphere can be directly computed from four points as

$$s^* = x_{c_1} \wedge x_{c_2} \wedge x_{c_3} \wedge x_{c_4}. \qquad (7)$$

If we replace one of these points for the point at infinity we get the equation of a plane

$$\pi^* = x_{c_1} \wedge x_{c_2} \wedge x_{c_3} \wedge e_\infty. \qquad (8)$$

So that π becomes in the standard form

$$\pi = I_c \pi^* = n + d e_\infty \qquad (9)$$

Where n is the normal vector and d represents the Hesse distance.

3.3 Circles and Lines

A circle z can be regarded as the intersection of two spheres s_1 and s_2 as $z = (s_1 \wedge s_2)$. The dual form of the circle can be expressed by three points lying on it

$$z^* = x_{c_1} \wedge x_{c_2} \wedge x_{c_3}. \qquad (10)$$

Similar to the case of planes, lines can be defined by circles passing through the point at infinity as:

$$L^* = x_{c_1} \wedge x_{c_2} \wedge e_\infty. \qquad (11)$$

The standard form of the line can be expressed by

$$L = l + e_\infty (t \cdot l), \qquad (12)$$

the line in the standard form is a bivector, and it has six parameters (Plucker coordinates), but just four degrees of freedom.

4 DIRECT KINEMATICS

The direct kinematics involves the computation of the position and orientation of the end-effector given the parameters of the joints. The direct kinematics can be easily computed given the lines of the axes of screws.

4.1 Rigid Transformations

We can express rigid transformations in conformal geometry carrying out reflections between planes.

4.1.1 Reflection

The reflection of conformal geometric entities help us to do any other transformation. The reflection of a point x with respect to the plane π is equal x minus twice the direct distance between the point and plane as shown in figure 2.

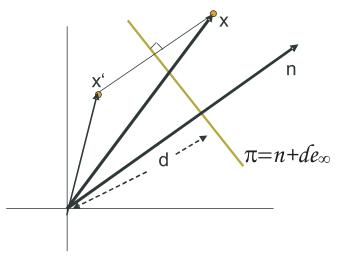


Figure 2: Reflection of a point x respect to the plane π.

For any geometric entity Q, the reflection respect to the plane π is given by

$$Q' = \pi Q \pi^{-1} \tag{13}$$

4.1.2 Translation

The translation of conformal entities can be done by carrying out two reflections in parallel planes π_1 and π_2 see the figure (3), that is

$$Q' = \underbrace{(\pi_2\pi_1)}_{T_a} Q \underbrace{(\pi_1^{-1}\pi_2^{-1})}_{\widetilde{T_a}} \tag{14}$$

$$T_a = (n + de_\infty)n = 1 + \frac{1}{2}ae_\infty = e^{-\frac{a}{2}e_\infty} \tag{15}$$

With $a = 2dn$.

4.1.3 Rotation

The rotation is the product of two reflections between nonparallel planes, (see figure (4))

$$Q' = \underbrace{(\pi_2\pi_1)}_{R_\theta} Q \underbrace{(\pi_1^{-1}\pi_2^{-1})}_{\widetilde{R_\theta}} \tag{16}$$

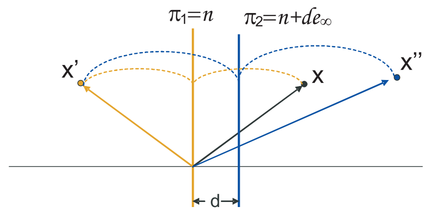


Figure 3: Reflection about parallel planes.

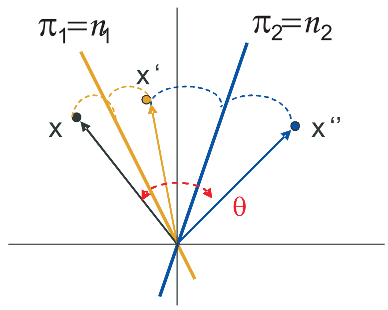


Figure 4: Reflection about nonparallel planes.

Or computing the conformal product of the normals of the planes.

$$R_\theta = n_2 n_1 = Cos(\frac{\theta}{2}) - Sin(\frac{\theta}{2})l = e^{-\frac{\theta}{2}l} \tag{17}$$

With $l = n_2 \wedge n_1$, and θ twice the angle between the planes π_2 and π_1. The screw motion *called motor* in (Bayro-Corrochano and Kahler, 2000) related to an arbitrary axis L is $M = TR\widetilde{T}$

$$Q' = \underbrace{(TR\widetilde{T})}_{M_\theta} Q \underbrace{((T\widetilde{R}\widetilde{T}))}_{\widetilde{M_\theta}} \tag{18}$$

$$M_\theta = TR\widetilde{T} = Cos(\frac{\theta}{2}) - Sin(\frac{\theta}{2})L = e^{-\frac{\theta}{2}L} \tag{19}$$

4.2 Kinematic Chains

The direct kinematics for serial robot arms is a succession of motors and it is valid for points, lines, planes, circles and spheres.

$$Q' = \prod_{i=1}^{n} M_i Q \prod_{i=1}^{n} \widetilde{M}_{n-i+1} \tag{20}$$

5 BARRETT HAND DIRECT KINEMATICS

The direct kinematics involves the computation of the position and orientation of the end-effector given the parameters of the joints. The direct kinematics can be easily computed given the lines of the axes of screws.

In order to explain the kinematics of the Barrett hand, we show the kinematics of one finger. In this example we will assume that the finger is totally extended. Note that such a hypothetical position is not reachable in normal operation, but this simplifies the explanation.

We initiated denoting some points on the finger which help to describe their position.

$$x_{1o} = A_w e_1 + A_1 e_2 + D_w e_3, \quad (21)$$
$$x_{2o} = A_w e_1 + (A_1 + A2) e_2 + D_w e_3, \quad (22)$$
$$x_{3o} = A_w e_1 + (A_1 + A_2 + A_3) e_2 + D_w e_3. \quad (23)$$

The points x_{1o}, x_{2o} and x_{3o} describe the position of each union and the end of the finger in the Euclidean space, see the figure 5.

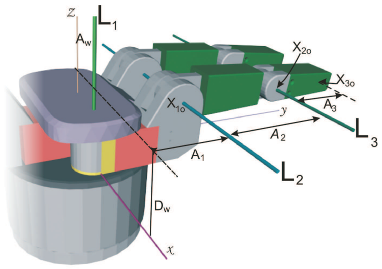


Figure 5: Barrett hand hypotetical position.

Having defined these points it is quite simple to calculate the axes,which will be used as motor's axis.

$$L_{1o} = -A_w(e_2 \wedge e_\infty) + e_{12}, \quad (24)$$
$$L_{2o} = (x_{1o} \wedge e_1 \wedge e_\infty) I_c, \quad (25)$$
$$L_{3o} = (x_{2o} \wedge e_1 \wedge e_\infty) I_c, \quad (26)$$

when the hand is initialized the fingers moves away to home position, this is $\Phi_2 = 2.46^o$ in union two and $\Phi_3 = 50^o$ degrees in union three. In order to move the finger from this hypothetical position to its home position the appropriate transformation need to be obtained.

$$M_{2o} = cos(\Phi_2/2) - sin(\Phi_2/2) L_{2o}, \quad (27)$$
$$M_{3o} = cos(\Phi_3/2) - sin(\Phi_3/2) L_{3o}, \quad (28)$$

Having obtained the transformations, then we apply them to the points and lines to them that must move.

$$x_2 = M_{2o} x_{2o} \widetilde{M}_{2o}, \quad (29)$$
$$x_3 = M_{2o} M_{3o} x_{3o} \widetilde{M}_{3o} \widetilde{M}_{2o}, \quad (30)$$
$$L_3 = M_{2o} L_{3o} \widetilde{M}_{2o}. \quad (31)$$

The point $x_1 = x_{1o}$ is not affected by the transformation, as are for the lines $L_1 = L_{1o}$ and $L_2 = L_{2o}$ see figure 6.

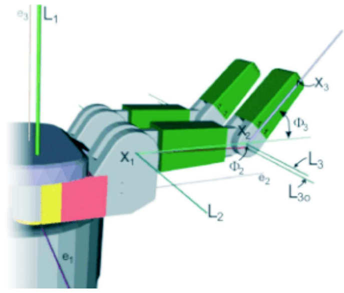


Figure 6: Barrett hand at home position.

Since the rotation angle of both axis L_2 and L_3 are related, we will use fractions of the angle q_1 to describe their individual rotation angles. The motors of each joint are computed using $\frac{2}{35} 2q_4$ to rotate around L_1, $\frac{1}{125} q_1$ around L_2 and $\frac{1}{375} q_1$ around L_3, the angles coefficients were taken from the Barrett hand user manual.

$$M_1 = cos(q_4/35) + sin(q_4/35) L_1, \quad (32)$$
$$M_2 = cos(q_1/250) - sin(q_1/250) L_2, \quad (33)$$
$$M_3 = cos(q_1/750) - sin(q_1/750) L_3. \quad (34)$$

The position of each point is related to the angles q_1 and q_4 as follows:

$$x'_1 = M_1 x_1 \widetilde{M}_1, \quad (35)$$
$$x'_2 = M_1 M_2 x_2 \widetilde{M}_2 \widetilde{M}_1, \quad (36)$$
$$x'_3 = M_1 M_2 M_3 x_3 \widetilde{M}_3 \widetilde{M}_2 \widetilde{M}_1, \quad (37)$$
$$L'_3 = M_1 M_2 L_3 \widetilde{M}_2 \widetilde{M}_1, \quad (38)$$
$$L'_2 = M_1 L_2 \widetilde{M}_1. \quad (39)$$

Since we already know x'_3, L'_1, L'_2 and L'_3 we can calculate the speed of the end of the finger using

$$\dot{X}'_3 = X'_3 \cdot \left(-\frac{2}{35} L'_1 \dot{q}_4 + \frac{1}{125} L'_2 \dot{q}_1 + \frac{1}{375} L'_3 \dot{q}_1\right). \quad (40)$$

6 POSE ESTIMATION

There are many approaches to solve the pose estimation problem ((Hartley and Zisserman, 2000)). In our approach we project the known mathematical model of the object on the camera's image. This is possible because after calibration we know the intrinsic parameters of the camera, see fig 7. The image of the mathematical projected model is compared with the image of the segmented object. If we find a match between them, then this means that the mathematical object is placed in the same position and orientation as the real object. Otherwise we follow a descendant gradient

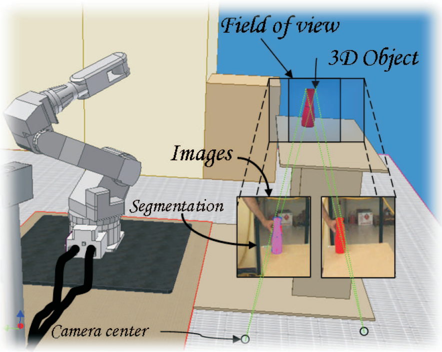


Figure 7: Mathematical model of the object.

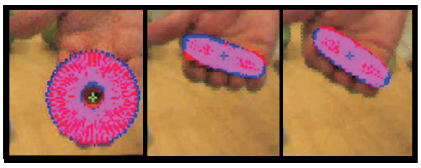


Figure 8: Pose estimation of a disk with a fixed camera.

based algorithm to rotate and translate the mathematical model in order to reduce the error between them. This algorithm runs very fast

Figure 8 shows the pose estimation result. In this case we have a maximum error of 0.4^o in the orientation estimation and 5mm of maximum error in the position estimation of the object. The problem becomes more difficult to solve when the stereoscopic system is moving. Figure 9 shows how well the stereo system track the object. If we want to know the real object's position with respect to the world coordinate system, of course we must know the extrinsic camera's parameters. Figure 10 illustrates the object's position and orientation with respect to the robot's hand. In the upper row of this figure we can see an augmented reality position sequence of the object. This shows that we can add the mathematical object in the real image. Furthermore, in the second row of the same image we can see the virtual reality pose estimation result.

Figure 9: Pose estimation of a recipient.

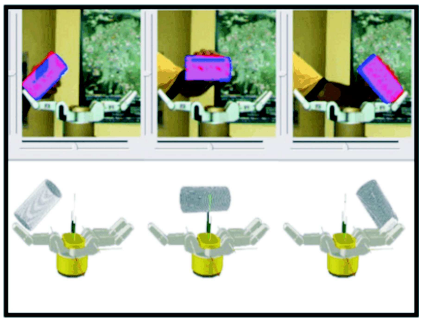


Figure 10: Object presented in augmented and virtual reality.

7 GRASPING THE OBJECTS

Considering that in using cameras we can only see the surface of the observed objects, in this work we consider them as bidimensional surfaces which are embed in a 3D space, and are described by the following function

$$\vec{H}(s,t) = h_x(s,t)e_1 + h_y(s,t)e_2 + h_z(s,t)e_3, \qquad (41)$$

where s and t are real parameters in the range $[0,1]$. Such parametrization allows us to work with different objects like points, conics, quadrics, or even more complex real objects like cups, glasses, etc.

Table 1: Functions of some objects.

Particle	$\vec{H} = 3e_1 + 4e_2 + 5e_3$
Cylinder	$\vec{H} = cos(t)e_1 + sin(t)e_2 + se_3$
Plane	$\vec{H} = te_1 + se_2 + (3s + 4t + 2)e_3$

There are many styles of grasping, however we are taking into account only three principal styles. Note that also for each style of grasping there are many possible solutions, for another approach see (Ch Borst and Hirzinger, 1999).

7.1 Style of Grasp One

Since our objective is to grasp such objects with the Barrett Hand, we must consider that it has only three fingers, so the problem consists in finding three points of grasping for which the system is in equilibrium by holding; this means that the sum of the forces are equal to zero, and also the sum of the moments.

We know the surface of the object, so we can compute its normal vector in each point using

$$N(s,t) = \left(\frac{\partial \vec{H}(s,t)}{\partial s} \wedge \frac{\partial \vec{H}(s,t)}{\partial t} \right) I_e. \qquad (42)$$

In surfaces with low friction the value of F tends to its projection over the normal ($F \approx F_n$). To maintain equilibrium, the sum of the forces must be zero $\sum_{i=1}^{3} \|F_n\| N(s_i,t_i) = 0$, (Fig. 11).

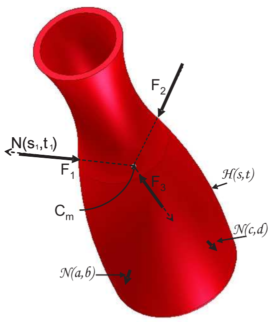


Figure 11: Object and his normal vectors.

This fact restricts the points over the surface in which the forces can be applied. This number of points is more reduced if we consider that the forces over the object are equal.

$$\sum_{i=1}^{3} N(s_i,t_i) = 0. \tag{43}$$

Additionally, in order to maintain the equilibrium of the system, it must be accomplished that the sum of the moments is zero

$$\sum_{i=1}^{3} H(s,t) \wedge N(s,t) = 0. \tag{44}$$

The points on the surface with the maximum and minim distance to the mass's center of the object fulfill $H(s,t) \wedge N(s,t) = 0$. The normal vector in such points crosses the center of mass (C_m) and it does not produce any moment. Before determining the external and internal points, we must compute the center of mass as

$$C_m = \int_0^1 \int_0^1 \vec{H}(s,t) ds dt \tag{45}$$

Once C_m is calculated we can establish the next restriction

$$(H(s,t) - C_m) \wedge N(s,t) = 0. \tag{46}$$

The values s and t satisfying (46), form a subspace and they fulfill that $H(s,t)$ are critical points on the surface (being maximums, minimums or inflections)

The constraint imposing that the three forces must be equal is hard to fulfill because it implies that the three points must be symmetric with respect to the mass center. When such points are not present, we can relax the constraint to allow that only two forces are equal in order to fulfill the hand's kinematics equations. Then, the normals $N(s_1,t_1)$ and $N(s_2,t_2)$ must be symmetric with respect to $N(s_3,t_3)$

$$N(s_3,t_3)N(s_1,t_1)N(s_3,t_3)^{-1} = N(s_2,t_2) \tag{47}$$

7.2 Style of Grasp Two

In the previous style of grasping three points of contact were considered. In this section we are taking into account a greater number of contact points, this fact generates a style of grasping that take the objects more secure. To increment the number of contact points is taken into account the base of the hand.

Since the object is described by the equation $H(s,t)$ it is possible to compute a plane π_b that divides the object in the middle, this is possible using lineal regression and also for the principal axis L_p. See figure 12.

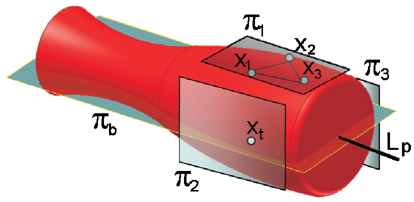


Figure 12: Planes of the object.

One Select only the points from locations with normal parallels to the plane π_b

$$N(s,t) \wedge \pi_b \approx 0 \tag{48}$$

Now we chose three points separated by 25 mm to generate a plane in the object. In this style of grasping the position of the hand relative to the object is trivial, because we just need to align the center of these points with the center of the hand's base. Also the orientation is the normal of the plane $\pi_1 = x_1 \wedge x_2 \wedge x_3 \wedge e_\infty$.

7.3 Style of Grasp Three

In this style of grasping the forces F_1, F_2 and F_3 do not intersect the mass center. They are canceled by symmetry, because the forces are parallel.

$$N(s_3,t_3)F_3 = N(s_1,t_1)F_1 + N(s_2,t_2)F_2. \tag{49}$$

Also the forces F_1, F_2 and F_3 are in the plane π_b and they are orthogonal to the principal axis L_p ($\pi_b = L_p \cdot N(s,t)$) as you can see in the figure 13.

A new restriction is then added to reduce the subspace of solutions

$$F_3 = 2F_1 = 2F_2, \tag{50}$$

$$N(s_1,t_1) = N(s_2,t_2) = -N(s_3,t_3). \tag{51}$$

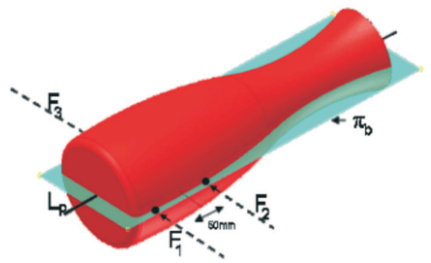


Figure 13: Forces of grasping.

Finally the direct distance between the parallels apply to x_1 and x_2 must be equal to 50 mm and between x_1, x_2 to x_3 must be equal to 25 mm.

Now we search exhaustively three points changing s_i and t_i. Figure 14 shows the simulation and result of this grasping algorithm. The position of the object relative to the hand must be computed using a coordinate frame in the object and other in the hand.

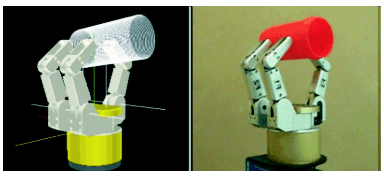


Figure 14: Simulation and result of the grasping.

8 TARGET POSE

Once the three grasping points ($P_1 = H(s_1,t_1)$, $P_2 = H(s_2,t_2)$, $P_3 = H(s_3,t_3)$) are calculated, for each finger it is really easy to determine the angles at the joints. To determine the angle of the spread ($q_4 = \beta$),

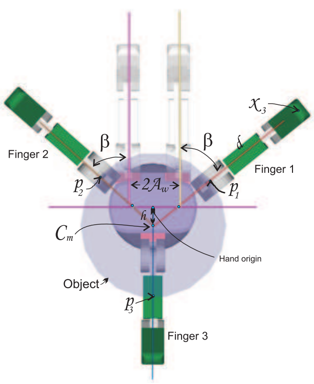


Figure 15: Object's position relative to the hand.

we use

$$cos\beta = \frac{(p_1 - C_m)\cdot(C_m - p_3)}{|p_1 - c_m|\,|C_m - p_3|}. \quad (52)$$

To calculate each one of the finger angles, we determine its elongation as

$$x_3' \cdot e_2 = |(p_1 - C_m)| - \frac{A_w}{\sin(\beta)} - A_1, \quad (53)$$

$$x_3' \cdot e_2 = |(p_2 - C_m)| - \frac{A_w}{\sin(\beta)} - A_1, \quad (54)$$

$$x_3' \cdot e_2 = |(p_3 - C_m)| + h - A_1, \quad (55)$$

where $x_3' \cdot e_2$ determines the opening distance of the finger

$$x_3' \cdot e_2 = (M_2 M_3 x_3 \widetilde{M}_3 \widetilde{M}_2)\cdot e_2 \quad (56)$$

$$x_3' \cdot e_2 = A_1 + A_2 \cos\left(\tfrac{1}{125}q + I_2\right) + \\ + A_3 \cos\left(\tfrac{4}{375}q + I_2 + I_3\right). \quad (57)$$

Solving for the angle q we have the opening angle for each finger. These angles are computed off line for each style of grasping of each object. They are the target in the velocity control of the hand.

8.1 Object Pose

We must find the transformation M which allows us to put the hand in a such way that each finger-end coincides with the corresponding contact point. For the sake of simplicity transformation M is divided in three transformations (M_1, M_2, M_3). With the same purpose we label the finger ends as X_1, X_2 and X_3, and the contact points as P_1, P_2 and P_3.

The first transformation M_1 is the translation between the object and the hand, which is equal to the directed distance between the centers of the circles called $Z_h^* = X_1 \wedge X_2 \wedge X_3$ y $Z_o^* = P_1 \wedge P_2 \wedge P_3$, and it can be calculated as

$$M_1 = e^{-\frac{1}{2}\left(\frac{Z_h^*}{Z_h^* \wedge e_\infty} \wedge \frac{Z_o^*}{Z_o^* \wedge e_\infty} \wedge e_\infty\right) I_c}. \quad (58)$$

The second transformation allows the alignment of the planes $\pi_h^* = Z_h^* = X_1 \wedge X_2 \wedge X_3 \wedge e_\infty$ and $\pi_o^* = Z_o^* \wedge e_\infty$, which are generated by the new points of the hand and the object. This transformation is calculated as $M_2 = e^{-\frac{1}{2}\pi_h \wedge \pi_o}$. The third transformation allows that the points overlap and this can be calculated using the planes $\pi_1^* = Z_o \wedge X_3 \wedge e_\infty$ and $\pi_2^* = Z_o \wedge P_3 \wedge e_\infty$, which are generated by the circle's axis and any of the points $M_3 = e^{-\frac{1}{2}\pi_1 \wedge \pi_2}$.

These transformations define also the pose of the object relative to the hand. They are computed off line in order to know the target position and orientation of the object with respect to the hand, it will be used to design a control law for visually guided grasping

9 VISUALLY GUIDED GRASPING

Once the target position and orientation of the object is known for each style of grasping and the hand's posture (angles of joints), it is possible to write a control law using this information and the equation of differential kinematics of the hand that it allows by using visual guidance to take an object.
Basically the control algorithm takes the pose of the object estimated as shown in the Section 6 and compares with the each one of the target poses computed in the Section 8 in order to choose as the target the closest pose, in this way the style of grasping is automatically chosen.

Once the style of grasping is chosen and target pose is known, the error ε between the estimated and computed is used to compute the desired angles in the joints of the hand

$$\alpha_d = \alpha_t e^{-\varepsilon^2} + (1 - e^{-\varepsilon^2})\alpha_a \tag{59}$$

where α_d is the desired angle of the finger, α_t is the target angle computed in the section 8 and α_a is the actual angle of the finger. Now the error between the desired and the actual position is used to compute the new joint angle using the equation of differential kinematics of the Barrett hand given in the Section 5.

9.1 Results

Next we show the results of the combination of the algorithms of pose estimation, visual control and grasping to create a new algorithm for visually guided grasping. In the Figure 16 a sequence of images of the grasping is presented. When the bottle is approached by the hand the fingers are looking for a possible point of grasp.

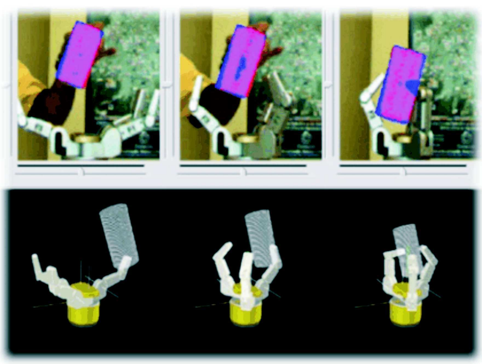


Figure 16: Visually guided grasping.

Now we can change the object or the pose of the object and the algorithm is computing a new behavior of grasping. The figure (17) shows a sequence of images changing the pose of the object.

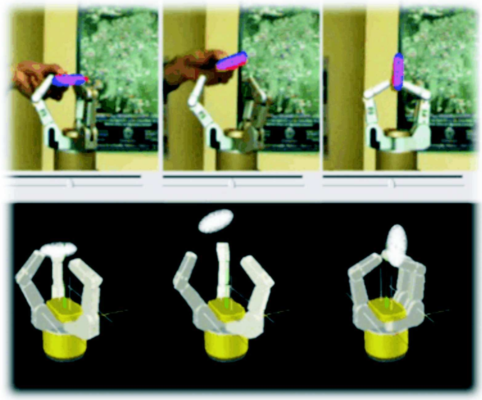


Figure 17: Changing the object's pose.

10 CONCLUSION

In this paper the authors used conformal geometric algebra to formulate grasping techniques. Using stereo vision we are able to detect the 3D pose and the intrinsic characteristics of the object shape. Based on this intrinsic information we developed feasible grasping strategies.

This paper emphasizes the importance of the development of algorithms for perception and grasping using a flexible and versatile mathematical framework

REFERENCES

Bayro-Corrochano, E. and Kahler, D. (2000). Motor algebra approach for computing the kinematics of robot manipulators. In *Journal of Robotic Systems*. 17(9):495-516.

Ch Borst, M. F. and Hirzinger, G. (1999). A fast and robust grasp planner for arbitrary 3d objects. In *ICRA99, International Conference on Robotics and Automation*. pages 1890-1896.

H. Li, D. Hestenes, A. R. (2001). *Generalized Homogeneous coordinates for computational geometry*. pages 27-52, in ((Somer, 2001)).

Hartley and Zisserman, A. (2000). *Multiple View Geometry in Computer Vision*. Cambridge University Press, UK, 1st edition.

Ruf, A. (2000). Closing the loop between articulated motion and stereo vision: a projective approach. In *PhD. Thesis, INP, GRENOBLE*.

Somer, G. (2001). *Geometric Computing with Clifford Algebras*. Springer-Verlag, Heidelberg, 2nd edition.

GEOMETRIC CONTROL OF A BINOCULAR HEAD

Eduardo Bayro-Corrochano and Julio Zamora-Esquivel
Department of Electrical Engineering and Computer Science
CINVESTAV, unidad Guadalajara. Jalisco, Mexico
jzamora@gdl.cinvestav.mx, edb@gdl.cinvestav.mx

Keywords: Conformal Geometry, Kinematics, Tracking.

Abstract: In this paper the authors use geometric algebra to formulate the differential kinematics of a binocular robotic head and reformulate the interaction matrix in terms of the lines that represent the principal axes of the camera. This matrix relates the velocities of 3D objects and the velocities of their images in the stereo images. Our main objective is the formulation of a kinematic control law in order to close the loop between perception and action, which allows to perform a smooth visual tracking.

1 INTRODUCTION

In this work we formulate the problem of visual tracking and design a control law by velocity feedback that allows us to close the loop between perception and action. Geometric algebra allow us to work with geometric entities like points, lines and planes and helps in the representation of rigid transformations. In this mathematical framework we straightforwardly formulate the direct and differential kinematics of robotic devices like the binocular robot head. On the other hand we show a reformulation of visual Jacobean which relates the velocity of a tridimensional object with the velocity of its projection onto the stereo camera images. Finally we write an expression that relates the joint velocities in the pan-tilt unit and velocities of the points in the camera image.

We start this work presenting a brief description of the geometric entities and conformal transformations them we show the kinematics of a pan-tilt unit and formulate its control law.

In contrast to other authors like (C. Canudas de Wit and Bastin, 1996), (Ruf, 2000) or (Kim Jung-Ha., 1990) we will use multivectors instead of matrices to formulate the control law it reduces the computation and improve the performance of the controller.

2 GEOMETRIC ALGEBRA: AN OUTLINE

Let $\mathcal{G}_n$ denote the geometric algebra of n-dimensions, this is a graded linear space. As well as vector addition and scalar multiplication we have a non-commutative product which is associative and distributive over addition – this is the *geometric* or *Clifford product*.

The inner product of two vectors is the standard *scalar* or *dot* product and produces a scalar. The outer or wedge product of two vectors is a new quantity which we call a *bivector*. We think of a bivector as a oriented area in the plane containing a and b, formed by sweeping a along b.

Thus, $b \wedge a$ will have the opposite orientation making the wedge product anti-commutative. The outer product is immediately generalizable to higher dimensions – for example, $(a \wedge b) \wedge c$, a *trivector*, is interpreted as the oriented volume formed by sweeping the area $a \wedge b$ along vector c. The outer product of k vectors is a k-vector or k-blade, and such a quantity is said to have *grade k*. A *multivector* (linear combination of objects of different type) is *homogeneous* if it contains terms of only a single grade.

2.1 The Geometric Algebra of n-D Space

In this paper we will specify a geometric algebra $\mathcal{G}_n$ of the n dimensional space by $\mathcal{G}_{p,q,r}$, where p, q and r

stand for the number of basis vector which squares to 1, -1 and 0 respectively and fulfill $n = p + q + r$.

We will use e_i to denote the vector basis i. In a Geometric algebra $G_{p,q,r}$, the geometric product of two basis vector is defined as

$$e_i e_j = \begin{cases} 1 & for \quad i = j \in 1, \cdots, p \\ -1 & for \quad i = j \in p+1, \cdots, p+q \\ 0 & for \quad i = j \in p+q+1, \cdots, p+q+r. \\ e_i \wedge e_j & for \quad i \neq j \end{cases}$$

This leads to a basis for the entire algebra:

$$\{1\}, \{e_i\}, \{e_i \wedge e_j\}, \{e_i \wedge e_j \wedge e_k\}, \ldots, \{e_1 \wedge e_2 \wedge \ldots \wedge e_n\} \quad (1)$$

Any multivector can be expressed in terms of this basis.

3 CONFORMAL GEOMETRY

Geometric algebra $G_{4,1}$ can be used to treat conformal geometry in a very elegant way. To see how this is possible, we follow the same formulation presented in (H. Li, 2001) and show how the Euclidean vector space $\mathbb{R}^3$ is represented in $\mathbb{R}^{4,1}$. This space has an orthonormal vector basis given by $\{e_i\}$ and $e_{ij} = e_i \wedge e_j$ are bivectorial basis and e_{23}, e_{31} and e_{12} correspond to the Hamilton basis. The unit Euclidean pseudo-scalar $I_e := e_1 \wedge e_2 \wedge e_3$, a pseudo-scalar $I_c := I_e E$ and the bivector $E := e_4 \wedge e_5 = e_4 e_5$ are used for computing the inverse and duals of multivectors.

3.1 The Stereographic Projection

The conformal geometry is related to a stereographic projection in Euclidean space. A stereographic projection is a mapping taking points lying on a hypersphere to points lying on a hyperplane. In this case, the projection plane passes through the equator and the sphere is centered at the origin. To make a projection, a line is drawn from the north pole to each point on the sphere and the intersection of this line with the projection plane constitutes the stereographic projection.

For simplicity, we will illustrate the equivalence between stereographic projections and conformal geometric algebra in $\mathbb{R}^1$. We will be working in $\mathbb{R}^{2,1}$ with the basis vectors $\{e_1, e_4, e_5\}$ having the usual properties. The projection plane will be the x-axis and the sphere will be a circle centered at the origin with unitary radius.

Given a scalar x_e representing a point on the x-axis, we wish to find the point x_c lying on the circle that projects to it (see Figure 1). The equation of the line passing through the north pole and x_e is given by $f(x) = -\frac{1}{x_e}x + 1$ and the equation of the circle $x^2 + f(x)^2 = 1$. Substituting the equation of the line on the circle, we get the point of intersection x_c which can be represented in homogeneous coordinates as the vector

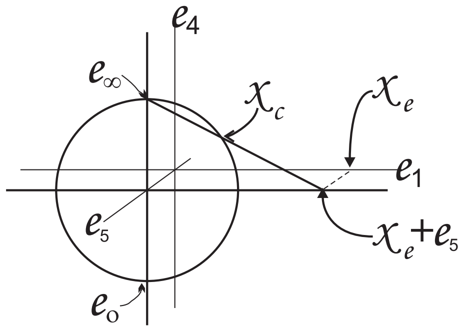


Figure 1: Stereographic projection for 1-D.

$$x_c = 2\frac{x_e}{x_e^2+1}e_1 + \frac{x_e^2-1}{x_e^2+1}e_4 + e_5. \quad (2)$$

From (2) we can infer the coordinates on the circle for the point at infinity as

$$e_\infty = \lim_{x_e \to \infty} \{x_c\} = e_4 + e_5, \quad (3)$$

$$e_o = \frac{1}{2} \lim_{x_e \to 0} \{x_c\} = \frac{1}{2}(-e_4 + e_5), \quad (4)$$

Note that (2) can be rewritten to

$$x_c = x_e + \frac{1}{2}x_e^2 e_\infty + e_o, \quad (5)$$

3.2 Spheres and Planes

The equation of a sphere of radius ρ centered at point $p_e \in \mathbb{R}^n$ can be written as $(x_e - p_e)^2 = \rho^2$. Since $x_c \cdot y_c = -\frac{1}{2}(\mathbf{x}_e - \mathbf{y}_e)^2$ and $x_c \cdot p_c = -\frac{1}{2}\rho^2$., we can rewrite the formula above in terms of homogeneous coordinates as. Since $x_c \cdot e_\infty = -1$ we can factor the expression above to

$$x_c \cdot (p_c - \frac{1}{2}\rho^2 e_\infty) = 0. \quad (6)$$

Which finally yields the simplified equation for the sphere as $s = p_c - \frac{1}{2}\rho^2 e_\infty$. Note from this equation that a point is just a sphere with zero radius. Alternatively, the dual of the sphere is represented as 4-vector $s^* = sI_c$. The advantage of the dual form is that the sphere can be directly computed from four points as

$$s^* = x_{c_1} \wedge x_{c_2} \wedge x_{c_3} \wedge x_{c_4}. \quad (7)$$

If we replace one of these points for the point at infinity we get the equation of a plane

$$\pi^* = x_{c_1} \wedge x_{c_2} \wedge x_{c_3} \wedge e_\infty. \quad (8)$$

So that π becomes in the standard form

$$\pi = I_c \pi^* = n + d e_\infty \quad (9)$$

Where n is the normal vector and d represents the Hesse distance.

3.3 Circles and Lines

A circle z can be regarded as the intersection of two spheres s_1 and s_2 as $z = (s_1 \wedge s_2)$. The dual form of the circle can be expressed by three points lying on it

$$z^* = x_{c_1} \wedge x_{c_2} \wedge x_{c_3}. \quad (10)$$

Similar to the case of planes, lines can be defined by circles passing through the point at infinity as:

$$L^* = x_{c_1} \wedge x_{c_2} \wedge e_\infty. \quad (11)$$

The standard form of the line can be expressed by

$$L = l + e_\infty (t \cdot l), \quad (12)$$

the line in the standard form is a bivector, and it has six parameters (Plucker coordinates).

4 RIGID TRANSFORMATIONS

We can express rigid transformations in conformal geometry carrying out reflections between planes.

4.0.1 Reflection

The reflection of conformal geometric entities help us to do any other transformation. The reflection of a point x respect to the plane π is equal x minus twice the direct distance between the point and plane see the image 2, that is $x = x - 2(\pi \cdot x)\pi^{-1}$ to simplify this expression recalling the property of Clifford product of vectors $2(b \cdot a) = ab + ba$.

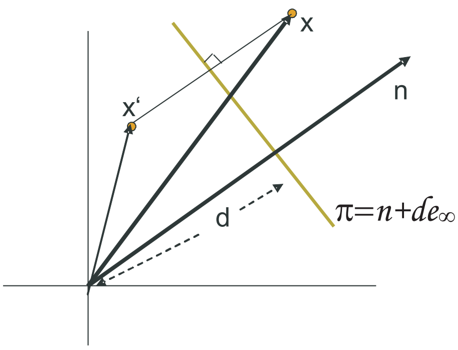


Figure 2: Reflection of a point x respect to the plane π.

For any geometric entity Q, the reflection respect to the plane π is given by

$$Q' = \pi Q \pi^{-1} \quad (13)$$

4.0.2 Translation

The translation of conformal entities can be done by carrying out two reflections in parallel planes π_1 and π_2 see the image (3), that is

$$Q' = \underbrace{(\pi_2 \pi_1)}_{T_a} Q \underbrace{(\pi_1^{-1} \pi_2^{-1})}_{\widetilde{T_a}} \quad (14)$$

$$T_a = (n + de_\infty)n = 1 + \frac{1}{2} a e_\infty = e^{-\frac{a}{2} e_\infty} \quad (15)$$

With $a = 2dn$.

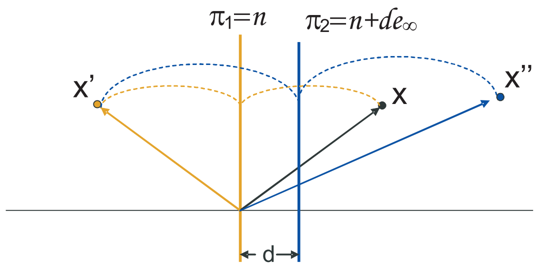


Figure 3: Reflection about parallel planes.

4.0.3 Rotation

The rotation is the product of two reflections between nonparallel planes, (see image (4))

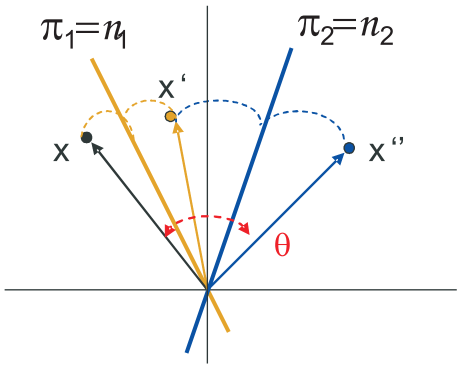


Figure 4: Reflection about nonparallel planes.

$$Q' = \underbrace{(\pi_2 \pi_1)}_{R_\theta} Q \underbrace{(\pi_1^{-1} \pi_2^{-1})}_{\widetilde{R_\theta}} \quad (16)$$

Or computing the conformal product of the normals of the planes.

$$R_\theta = n_2 n_1 = Cos(\frac{\theta}{2}) - Sin(\frac{\theta}{2}) l = e^{-\frac{\theta}{2} l} \quad (17)$$

With $l = n_2 \wedge n_1$, and θ twice the angle between the planes π_2 and π_1. The screw motion *called motor* related to an arbitrary axis L is $M = TR\widetilde{T}$

$$Q' = \underbrace{(TR\widetilde{T})}_{M_\theta}Q(\underbrace{(T\widetilde{R}\widetilde{T})}_{\widetilde{M_\theta}}) \quad (18)$$

$$M_\theta = TR\widetilde{T} = Cos(\frac{\theta}{2}) - Sin(\frac{\theta}{2})L = e^{-\frac{\theta}{2}L} \quad (19)$$

4.1 Kinematic Chains

The direct kinematics for serial robot arms is a succession of motors as you can see in (Bayro-Corrochano and Kahler, 2000) and it is valid for points, lines, planes, circles and spheres.

$$Q' = \prod_{i=1}^{n} M_i Q \prod_{i=1}^{n} \widetilde{M}_{n-i+1} \quad (20)$$

5 DIRECT KINEMATICS OF A PAN-TILT UNIT

We implement algorithm for the velocity control of a pan-tilt unit (PTU Fig. 5) assuming three degree of freedom. We consider the stereo depth as one virtual D.O.F. thus the PTU has a similar kinematic behavior as a robot with three D.O.F.

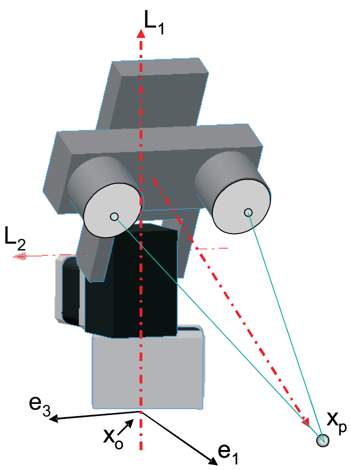


Figure 5: Pan tilt unit.

In order to carry out a velocity control, we need first to compute the direct kinematics, this is very easy to do, as we know the axis lines:

$$L_1 = -e_{31} \quad (21)$$
$$L_2 = e_{12} + d_1 e_1 e_\infty \quad (22)$$
$$L_3 = e_1 e_\infty \quad (23)$$

Since $M_i = e^{-\frac{1}{2}q_i L_i}$ and $\widetilde{M}_i = e^{\frac{1}{2}q_i L_i}$, The position of end effectors is computed as

$$x_p(q) = x'_p = M_1 M_2 M_3 x_p \widetilde{M}_3 \widetilde{M}_2 \widetilde{M}_1, \quad (24)$$

The state variable representation of the system is as follows

$$\begin{cases} \dot{x}'_p = x' \cdot \begin{pmatrix} L'_1 & L'_2 & L'_3 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \\ y = x'_p \end{cases} \quad (25)$$

where the position of end effector at home position x_p is the conformal mapping of $x_{p_e} = d_3 e_1 + (d_1 + d_2) e_2$ (see eq. 5), the line L'_i is the current position of L_i and u_i is the velocity of the i-junction of the system. As L_3 is an axis at infinity M_3 is a translator, that is, the virtual component is a prismatic junction.

5.1 Exact Linearization via Feedback

Now the following state feedback control law is chosen in order to get a new linear an controllable system.

$$\begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \begin{pmatrix} x'_p \cdot L'_1 & x'_p \cdot L'_2 & x'_p \cdot L'_3 \end{pmatrix}^{-1} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} \quad (26)$$

Where $V = (v_1, v_2, v_3)^T$ is the new input to the linear system, then we rewrite the equations of the system

$$\begin{cases} \dot{x}'_p = V \\ y = x'_p \end{cases} \quad (27)$$

5.2 Asymptotical Output Tracking

The problem of follow a constant reference x_t is solved computing the error between end effectors position x'_p and the target position x_t as $e_r = (x'_p \wedge x_t) \cdot e_\infty$, the control law is then given by.

$$V = -ke \quad (28)$$

This error is small if the control system is doing it's job, it is mapped to an error in the joint space using the inverse Jacobian.

$$U = J^{-1} V \quad (29)$$

Doing the Jacobian $J = x'_p \cdot \begin{pmatrix} L'_1 & L'_2 & L'_3 \end{pmatrix}$

$$j_1 = x'_p \cdot (L_1) \quad (30)$$
$$j_2 = x'_p \cdot (M_1 L_2 \widetilde{M}_1) \quad (31)$$
$$j_3 = x'_p \cdot (M_1 M_2 L_3 \widetilde{M}_2 \widetilde{M}_1) \quad (32)$$

Once that we have the Jacobian is easy to compute the dq_i using the crammer's rule.

$$\begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = (j_1 \wedge j_2 \wedge j_3)^{-1} \cdot \begin{pmatrix} V \wedge j_2 \wedge j_3 \\ j_1 \wedge V \wedge j_3 \\ j_1 \wedge j_2 \wedge V \end{pmatrix} \quad (33)$$

This is possible because $j_1 \wedge j_2 \wedge j_3 = det(J)I_e$. Finally we have dq_i which will tend to reduce these errors.

5.3 Visual Jacobian

A point in the image is given by $s = (x,y)^T$ whereas a 3-D point is represented as X. The relationship between $\dot{s}$ and $\dot{S}$ is called visual Jacobian.

Taking a camera in general position his projection matrix is represented by the planes π_1, π_2 y π_3 more details in (Hartley and Zisserman, 2000).

$$P = \begin{pmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \end{pmatrix}, \quad (34)$$

The point X is projected in the image in the point

$$s = \begin{pmatrix} \frac{\pi_1 \cdot X}{\pi_3 \cdot X} \\ \frac{\pi_2 \cdot X}{\pi_3 \cdot X} \end{pmatrix} \quad (35)$$

To simplify the explanation the x variable is introduced and his time derivative $\dot{x}$ defined as

$$x = \begin{pmatrix} \pi_1 \cdot X \\ \pi_2 \cdot X \\ \pi_3 \cdot X \end{pmatrix} \qquad \dot{x} = \begin{pmatrix} \pi_1 \cdot \dot{X} \\ \pi_2 \cdot \dot{X} \\ \pi_3 \cdot \dot{X} \end{pmatrix} \quad (36)$$

Now s is given by $s_1 = \frac{x_1}{x_3}$ and his derivative

$$\dot{s}_1 = \dot{x}_1 \frac{1}{x_3} + x_1 \left(\frac{-\dot{x}_3}{x_3^2} \right) \quad (37)$$

$$\dot{s}_1 = \frac{x_3 \dot{x}_1 - x_1 \dot{x}_3}{x_3^2} \quad (38)$$

By sustitution of x and $\dot{x}$ in the equation 38 is obtained

$$\dot{s}_1 = \kappa [(\pi_3 \cdot X)\pi_1 - (\pi_1 \cdot X)\pi_3] \cdot \dot{X} \quad (39)$$

$$\dot{s}_1 = \kappa [X \cdot (\pi_3 \wedge \pi_1)] \cdot \dot{X} \quad (40)$$

where $\kappa = \frac{1}{x_3^2}$. Doing the same steps for s_2 it possible to write the equation

$$\dot{s} = \kappa X \cdot \begin{pmatrix} \pi_1 \wedge \pi_3 \\ \pi_2 \wedge \pi_3 \end{pmatrix} \cdot \dot{X} \quad (41)$$

Geometrically $\pi_1 \wedge \pi_3$ represents a line of intersection of the planes π_1 and π_3. Denoting by L_x and L_y the lines of this intersection as

$$L_x = \pi_1 \wedge \pi_3 \quad (42)$$

$$L_y = \pi_2 \wedge \pi_3 \quad (43)$$

It is posible to rewrite 41 as

$$\dot{s} = \kappa X \cdot \begin{pmatrix} L_x \\ L_y \end{pmatrix} \cdot \dot{X} \quad (44)$$

In order to close the loop between the perception and action, the relationship between velocities in the points of the image and the velocities in the joints of the pan-tilt unit is computed.

Taking the equation of differential kinematics 25 and visual Jacobian 44 it is possible to write a new expression

$$\dot{s} = \kappa \begin{pmatrix} (X' \cdot L'_x) \cdot (X' \cdot L'_1) & (X' \cdot L'_x) \cdot (X' \cdot L'_2) \\ (X' \cdot L'_y) \cdot (X' \cdot L'_1) & (X' \cdot L'_y) \cdot (X' \cdot L'_2) \end{pmatrix} \dot{q} \quad (45)$$

We can write a similar expression using the differential kinematics of the Barrett Hand. The equation 45 is very useful to design a control law to track an object or to grasp it.

5.4 Exact Linearization via Feedback

Now the following state feedback control law is chosen in order to get a new linear an controllable system.

$$\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} (X' \cdot L'_x) \cdot (X' \cdot L'_1) & (X' \cdot L'_x) \cdot (X' \cdot L'_2) \\ (X' \cdot L'_y) \cdot (X' \cdot L'_1) & (X' \cdot L'_y) \cdot (X' \cdot L'_2) \end{pmatrix}^{-1} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$$

Where $V = (v_1, v_2)^T$ is the new input to the linear system, then we rewrite the equations of the system

$$\begin{cases} \dot{s}'_p = V \\ y = s'_p \end{cases} \quad (46)$$

5.5 Experimental Results

In this experiment the binocular head should smoothly track a target. The figure (6) show the 3D coordinates of the focus of attention. The figure (7) show examples of the image sequence. We can see that the curves of the 3D object trajectory are very rough, however the control rule manages to keep the trajectory of the pan-tilt unit smooth.

In the experiment the coordinate system is in the center of the camera. Then the principal planes of the camera are given by

$$\pi_1 = f_x e_1 + x_o e_3 \quad (47)$$

$$\pi_2 = f_y e_2 + y_o e_3 \quad (48)$$

$$\pi_3 = e_3 \quad (49)$$

whre f_x, f_y, x_o y y_o are the camera's parameters. Using this planes we compute the lines L_x y L_y, by the way the axis of the pan-tilt are known.

$$L_1 = e_{23} + d_1 e_2 \quad (50)$$

$$L_2 = e_{12} + d_2 e_2 \quad (51)$$

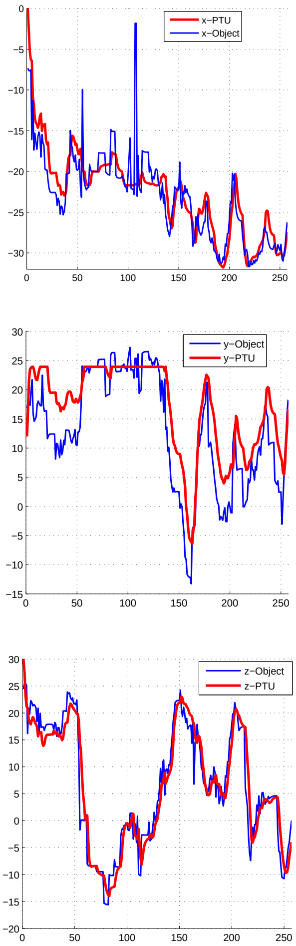


Figure 6: *x*, *y* and *z* coordinate of the focus of attention.

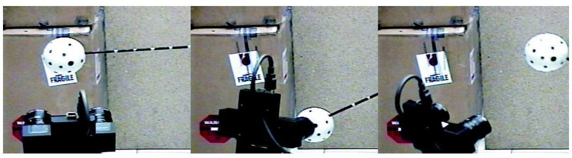


Figure 7: Sequence of tracking.

Note that the tilt axis is called L_1 and the pan axis is L_2, because the coordinate system is in the camera. also L_2' is a function of the tilt angle $L_2' = M_1 L_2 \tilde{M}_1$ with $M_1 = cos(\theta_{tilt}) + sen(\theta_{tilt})L_1$. In this experiment a point over the boar was selected and using the KLT algorithm was tracked the displacement in the image is transformed to velocities of the pan-tilt's joint using the visual-mechanical Jacobean Eq. 45.

As result in the image (8) we can see a sequence of pictures captured by the robot. In these images the position of the board do not change while the background is in continuous change.

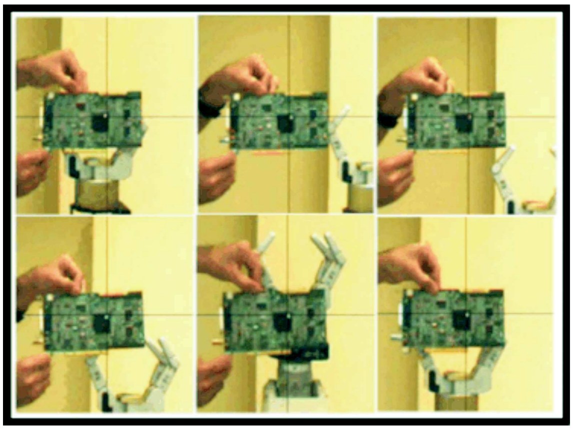


Figure 8: Sequence of tracking.

6 CONCLUSION

The authors show that is possible and easy to write a control law using the lines of the camera's axes as bivectors (Plcker coordinates) in the conformal geometry instead of the interaction matrix. This formulation combines the information of the camera's parameters with the axes of the pan tilt unit in order to create a matrix of the visual-mechanical Jacobian useful to write a velocity control law. The experiments confirm the effectiveness of our approach.

REFERENCES

Bayro-Corrochano, E. and Kahler, D. (2000). Motor algebra approach for computing the kinematics of robot manipulators. In *Journal of Robotic Systems*. 17(9):495-516.

C. Canudas de Wit, B. S. and Bastin, G. (1996). *Theory of Robot Control*. Springer, Berlin, 1st edition.

H. Li, D. Hestenes, A. R. (2001). *Generalized Homogeneous coordinates for computational geometry*. pages 27-52, in (Somer, 2001).

Hartley and Zisserman, A. (2000). *Multiple View Geometry in Computer Vision*. Cambridge University Press, UK, 1st edition.

Kim Jung-Ha., K. V. R. (1990). Kinematics of robot manipulators via line transformations. In *Journal of Robotic Systems*. pp. 647-674.

Ruf, A. (2000). Closing the loop between articulated motion and stereo vision: a projective approach. In *PhD. Thesis, INP, GRENOBLE*.

Somer, G. (2001). *Geometric Computing with Clifford Algebras*. Springer-Verlag, Heidelberg, 2nd edition.

DESIGN OF LOW INTERACTION DISTRIBUTED DIAGNOSERS FOR DISCRETE EVENT SYSTEMS

J. Arámburo-Lizárraga, E. López-Mellado and A. Ramírez-Treviño
CINVESTAV Unidad Guadalajara; Av. Científica 1145, Col. El Bajío; 45010 Zapopan, Jal, México
(phone: (33)37-70-37-00; fax: (33)37-70-37-09; email: jarambur@gdl.cinvestav. mx)

Keywords: Discrete Event Systems, Petri Nets, Distributed Diagnosis.

Abstract: This paper deals with distributed fault diagnosis of discrete event systems (*DES*). The approach held is model based: an interpreted Petri net (*IPN*) describes both the normal and faulty behaviour of *DES* in which both places and transitions may be non measurable. The diagnoser monitors the evolution of the *DES* outputs according to a model that describes the normal behaviour of the *DES*. A method for designing a set of distributed diagnosers is proposed; it is based on the decomposition of the *DES* model into reduced sub-models which require low interaction among them; the diagnosability property is studied for the set of resulting sub-models.

1 INTRODUCTION

Most of works study the diagnosability property and fault detection schemes based on a centralised approach using the global model of the *DES*. Recently, fault diagnosis of *DES* has been addressed through a distributed approach allowing breaking down the complexity when dealing with large and complex systems (Benveniste, et al., 2003; O. Contant, et al., 2004; Debouk, et al., 2000; Genc and Lafortune, 2003; Jiroveanu and Boel, 2003; Pencolé, 2004; Arámburo-Lizárraga, et al., 2005).

In (Debouk, et al., 2000) it is proposed a decentralised and modular approach to perform failure diagnosis based on Sampath's results (Sampath, et al., 1995). In (Contant, et al., 2004) and (Pencolé, 2004) the authors presented incremental algorithms to perform diagnosability analysis based on (Sampath, et al., 1995) in a distributed way; they consider systems whose components evolve by the occurrence of events; the parallel composition leads to a complete system model intractable. In (Genc and Lafortune, 2003) it is proposed a method that handles the reachability graph of the *PN* model in order to perform the analysis similarly to (Sampath, et al., 1995); based on design considerations the model is partitioned into two labelled *PN* and it is proven that the distributed diagnosis is equivalent to the centralised diagnosis; later, (Genc and Lafortune, 2005) extend the results to systems modelled by several labelled *PN* that share places, and present an algorithm to determine distributed diagnosis.

Our approach considers the system modelled as an interpreted *PN* (*IPN*) allowing describing the system with partially observable states and events; the model includes the possible faults it may occur. A structural characterisation and a diagnoser scheme was presented in (Ramírez-Treviño, et al., 2004); then in (Arámburo-Lizárraga, et al., 2005) we proposed a methodology for designing reduced diagnosers and presented an algorithm to split a global model into a set of communicating sub-models.

In this paper we present the formalisation of the distributed system model. The proposed distributed diagnoser scheme consists of communicating diagnoser modules, where each diagnoser can handle two kind of reduced models; the choice of the reduced models depends on some considerations of the system behaviour. In some cases the communication between modules is not necessary.

This paper is organised as follows. In section 2 basic definitions of *PN* and *IPN* are included. Section 3 summarises the concepts and results for centralised diagnosis. Section 4 presents the results related to distributed diagnosis analysis. Section V presents the method to get reduced sub-models that have low interaction among them.

2 BACKGROUND

We consider systems modelled by Petri Nets and Interpreted Petri Nets. A Petri Net is a structure $G = (P, T, I, O)$ where: $P = \{p_1, p_2, ..., p_n\}$ and $T = \{t_1, t_2, ... ,t_m\}$ are finite sets of nodes called respectively places and transitions, I (O): $P \times T \rightarrow \mathbb{Z}^+$ is a function representing the weighted arcs going from places to transitions (transitions to places), where $\mathbb{Z}^+$ is the set of nonnegative integers.

The symbol ${}^\bullet t_j$ ($t_j{}^\bullet$) denotes the set of all places p_i such that $I(p_i,t_j)\neq 0$ ($O(p_i,t_j)\neq 0$). Analogously, ${}^\bullet p_i$ ($p_i{}^\bullet$) denotes the set of all transitions t_j such that $O(p_i,t_j)\neq 0$ ($I(p_i,t_j)\neq 0$) and the incidence matrix of G is $C=[c_{ij}]$, where $c_{ij} = O(p_i,t_j) - I(p_i,t_j)$.

A marking function M: $P\rightarrow\mathbb{Z}^+$ represents the number of tokens (depicted as dots) residing inside each place. The marking of a PN is usually expressed as an n-entry vector.

A Petri Net system or Petri Net (PN) is the pair $N=(G,M_0)$, where G is a PN structure and M_0 is an initial token distribution. $R(G,M_0)$ is the set of all possible reachable markings from M_0 firing only enabled transitions.

In a PN system, a transition t_j is enabled at marking M_k if $\forall p_i \in P$, $M_k(p_i) \geq I(p_i,t_j)$; an enabled transition t_j can be fired reaching a new marking M_{k+1} which can be computed as $M_{k+1} = M_k + Cv_k$, where $v_k(i)=0$, $i\neq j$, $v_k(j)=1$.

This work uses Interpreted Petri Nets (IPN) (Ramírez-Treviño, et al., 2003) an extension to PN that allow to associate input and output signals to PN models. An IPN (Q, M_0) is an Interpreted Petri Net structure $Q = (G, \Sigma, \lambda, \varphi)$ with an initial marking M_0, where G is a PN structure, $\Sigma = \{\alpha_1, \alpha_2, ... ,\alpha_r\}$ is the input alphabet of the net, where α_i is an input symbol, λ: $T\rightarrow\Sigma \cup\{\varepsilon\}$ is a labelling function of transitions with the following constraint: $\forall t_j,t_k \in T, j \neq k$, if $\forall p_i$ $I(p_i,t_j) = I(p_i,t_k) \neq 0$ and both $\lambda(t_j) \neq \varepsilon$, $\lambda(t_k) \neq \varepsilon$, then $\lambda(t_j) \neq \lambda(t_k)$, in this case ε represents an internal system event, and $\varphi: R(Q,M_0)\rightarrow(\mathbb{Z}^+)^q$ is an output function that associates to each marking an output vector. Here q is the number of outputs. In this work φ is a $q\times n$ matrix. If the output symbol i is present (turned on) every time that $M(p_j)\geq 1$, then $\varphi(i,j)=1$, otherwise $\varphi(i,j)=0$.

A transition $t_j \in T$ of an IPN is enabled at marking M_k if $\forall p_i \in P$, $M_k(p_i) \geq I(p_i,t_j)$. When t_j is fired in a marking M_k, then M_{k+1} is reached, i.e., $M_k \xrightarrow{t_j} M_{k+1}$; M_{k+1} can be computed using the state equation:

$$\begin{aligned} M_{k+1} &= M_k + Cv_k \\ y_k &= \varphi(M_k) \end{aligned} \tag{1}$$

where C and v_k are defined as in PN and $y_k \in (\mathbb{Z}^+)^q$ is the k-th output vector of the IPN.

Let $\sigma = t_i t_j ... t_k ...$ be a firing transition sequence of an $IPN(Q,M_0)$ s.t. $M_0 \xrightarrow{t_i} M_1 \xrightarrow{t_j} ... M_x \xrightarrow{t_k} ...$ The set £(Q,M_0) of all firing transition sequences is called the firing language £(Q,M_0)=$\{\sigma = t_i t_j ... t_k ... \wedge M_0 \xrightarrow{t_i} M_1 \xrightarrow{t_j} ... M_x \xrightarrow{t_k} ...\}$.

According to functions λ and φ, transitions and places of an IPN (Q,M_0) if $\lambda(t_i) \neq \varepsilon$ the transition t_i is said to be manipulated. Otherwise it is non-manipulated. A place $p_i\in P$ is said to be measurable if the i-th column vector of φ is not null, i.e. $\varphi(\bullet,i) \neq 0$. Otherwise it is non-measurable.

The following concepts are useful in the study of the diagnosability property. A sequence of input-output symbols of (Q,M_0) is a sequence $\omega = (\alpha_0,y_0)(\alpha_1,y_1)...(\alpha_n,y_n)$, where $\alpha_j \in \Sigma \cup\{\varepsilon\}$ and α_{i+1} is the current input of the IPN when the output changes from y_i to y_{i+1}. It is assumed that $\alpha_0 = \varepsilon$, $y_0 = \varphi(M_0)$. The firing transition sequence $\sigma \in$ £(Q,M_0) whose firing actually generates ω is denoted by σ_ω. The set of all possible firing transition sequences that could generate the word ω is defined as $\Omega(\omega) = \{\sigma \mid \sigma \in$ £$(Q,M_0) \wedge$ the firing of σ produces $\omega\}$.

The set $\Lambda(Q,M_0) = \{\omega \mid \omega$ is a sequence of input-output symbols$\}$ denotes the set of all sequences of input-output symbols of (Q,M_0) and the set of all input-output sequences of length greater or equal than k will be denoted by $\Lambda^k(Q,M_0)$, i.e. $\Lambda^k(Q,M_0) = \{\omega \in \Lambda(Q,M_0) \mid |\omega| \geq k\}$ where $k \in \mathbb{N}$.

The set $\Lambda_B(Q,M_0)$, i.e., $\Lambda_B(Q,M_0) = \{\omega \in \Lambda(Q,M_0) \mid \sigma\in\Omega(\omega)$ such that $M_0 \xrightarrow{\sigma} M_j$ and M_j enables no transition, or when $M_j \xrightarrow{t_i}$ then $C(\bullet,t_i)=0\}$ denotes all input-output sequences leading to an ending marking in the IPN (markings enabling no transition or only self-loop transitions).

The following lemma (Ramírez-Treviño, et al., 2004) gives a polynomial characterisation of event-detectable IPN.

Lemma 1: A live IPN given by (Q,M_0) is event-detectable if and only if:

1. $\forall t_i, t_j \in T$ such that $\lambda(t_i) = \lambda(t_j)$ or $\lambda(t_i) = \varepsilon$ it holds that $\varphi C(\bullet,t_i) \neq \varphi C(\bullet,t_j)$ and
2. $\forall t_k \in T$ it holds that $\varphi C(\bullet,t_k) \neq 0$.

3 CENTRALISED DIAGNOSIS

The main results on diagnosability and diagnoser design in a centralised approach presented in (Ramírez-Treviño, et al., 2007) are outlined below.

3.1 System Modelling

The sets of nodes are partitioned into faulty (P^F and T^F) and normal functioning nodes (P^N and T^N); so $P = P^F \cup P^N$ and $T = T^F \cup T^N$. p_i^N denotes a place in P^N of the normal behaviour (Q^N, M_0^N). Since $P^N \subseteq$

P then p_i^N also belongs to (Q,M_0). The set of risky places of (Q,M_0) is $P^R = {}^\bullet T^F$. The post-risk transition set of (Q,M_0) is $T^R = P^{R\bullet} \cap T^N$.

Example. Figure 1 presents an *IPN* model of a system. The model has three faulty states, represented by places p_{16}, p_{17}, p_{18}. Function λ is defined as $\lambda(t_1)$=a, $\lambda(t_3)$=b, $\lambda(t_4)$=x, $\lambda(t_7)$=y, $\lambda(t_9)$=c, $\lambda(t_{10})$=z, for others transitions $\lambda(t_i)=\varepsilon$. Measurable places are p_3, p_5, p_8, p_{12}, p_{15}, $P^R = \{p_4, p_7, p_{12}\}$, $T^R = \{t_4, t_7, t_{10}\}$, $T^F = \{t_{13}, t_{14}, t_{15}\}$ and $P^F = \{p_{16}, p_{17}, p_{18}\}$.

3.2 Reduced Models

In a previous work (Arámburo-Lizárraga, et al., 2005) we stated that the condition of event-detectability is needed only on $t_j \in {}^\bullet P^R$ and $t_j \in P^{R\bullet}$. This fact can be exploited in order to obtain a reduced model containing the pertinent parts of $\left(Q^N, M_0^N\right)$ regarding the modelled faults in (Q,M_0).

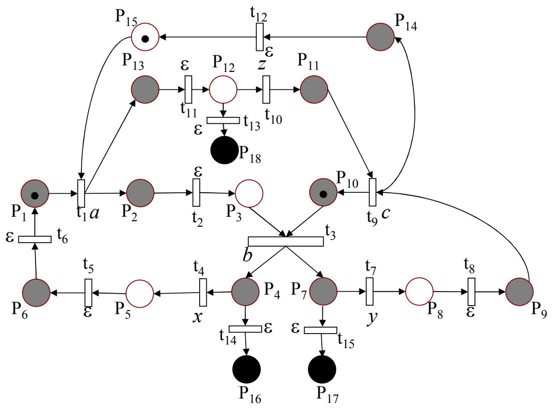


Figure 1: Global model.

Definition 1. Let $\left(Q^N, M_0^N\right)$ be the embedded normal behaviour included in (Q,M_0). The reduced model $\left(Q^{RM}, M_0^{RM}\right)$ of $\left(Q^N, M_0^N\right)$ is the subnet induced by:

- $P^{RM} = P_a \cup P_b \cup P_c$, where $P_a = \{p_i \mid p_i \in P^R\}$, $P_b = \{p_j \mid p_j \in P^{R\bullet\bullet}\}$, and $P_c = \{p_k \mid p_k \in {}^{\bullet\bullet}P^R$, p_k is a measurable place$\}$. The sets P_b and P_c are necessary only when $\exists p_i \in P^R$, such that p_i is non-measurable.
- $T^{RM} = T_{in} \cup T_{out}$, where $T_{in} = \{{}^\bullet p_i \mid p_i \in P^{RM}\}$, $T_{out} = \{p_i{}^\bullet \mid p_i \in P^{RM}\}$.
- $\lambda^{RM}: T^{RM} \rightarrow \Sigma \cup \{\varepsilon\}, \forall t_i' \in T^{RM}, \lambda(t_i') = \lambda(t_i), t_i \in T^N$, $t_i' = t_i$.
- $\varphi^{RM} = \varphi|_{R(Q^{RM}, M_0^{RM})}$
- $M_0^{RM} = M_0|_{P^{RM}}$.

The firing rules of $\left(Q^{RM}, M_0^{RM}\right)$ are defined:

- If $t_j \in T^{RM}$ is fired in (Q,M_0) then it must be fired in $\left(Q^{RM}, M_0^{RM}\right)$.
- If the input symbol $\lambda(t_k)$, $t_k \in P^{R\bullet}$ is activated in the system then it must be activated in $\left(Q^{RM}, M_0^{RM}\right)$.
- If $\exists t_j \in T^{RM}$, s.t., t_j is not event detectable then t_j is fired automatically when ${}^\bullet t_j$ was marked.

The reduced model nodes (places and transitions) are a copy of the original ones, and they have associated the same input-output symbols.

Figure 2 presents the reduced model of the global system model depicted in figure 1. Notice that in this example the number of places is reduced and T^{RM} are only event-detectable transitions.

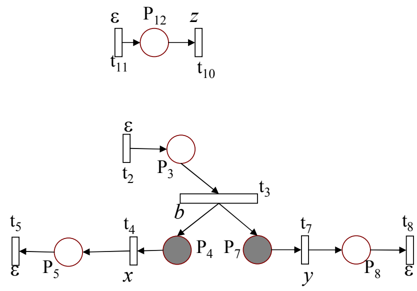


Figure 2: Diagnoser reduced model.

3.3 Characterisation of Diagnosability

The characterisation of input-output diagnosable *IPN* is based on the partition of $R(Q,M_0)$ into normal and faulty markings; all the faulty markings must be distinguishable from other reachable markings.

Definition 2: An *IPN* given by (Q,M_0) is said to be input - output diagnosable in $k < \infty$ steps if any marking $M_f \in F$ is distinguishable from any other $M_k \in R(Q,M_0)$ using any word $\omega \in \Lambda^k(Q,M_f) \cup \Lambda_B(Q,M_f)$, where $F = \{M \mid \exists p_k \in P^F$ such that $M(p_k)>0, M \in R(Q,M_0)\}$.

The following result extends that presented in (Ramírez-Treviño, et al., 2007).

Theorem 1: Let (Q,M_0) be a binary *IPN*, such that $\left(Q^N, M_0^N\right)$ is live, strongly connected and event detectable on $t_j \in {}^\bullet P^R$ and $t_j \in P^{R\bullet}$. Let $\{X_1,\ldots,X_\tau\}$ be the set of all T-semiflows of (Q,M_0). If $\forall\, p_i^N \in P^N$, $\left(p_i^N\right)^\bullet \cap T^F \neq \theta$ the following conditions hold:

1. $\forall r$, $\exists j$ $X_r(j) \geq 1$, where $t_j \in \left(p_i^N\right)^\bullet - T^F$,
2. $\forall t_k \in \left(p_i^N\right)^\bullet - T^F$, ${}^\bullet(t_k) = \{p_i^N\}$ and $\lambda(t_k) \neq \varepsilon$.

then the *IPN* (Q,M_0) is input-output diagnosable.

Proof: It is similar to that included in (Ramírez-Treviño, et al., 2007).

4 DISTRIBUTED DIAGNOSIS

4.1 Model Partition

In order to build a distributed diagnoser, the *IPN* model (Q, M_0) can be conveniently decomposed into m interacting subsystems where different modules share common nodes.

Definition 3. Let (Q,M_0) be an *IPN*. The distributed Interpreted Petri Net model *DN* of (Q,M_0) is a finite set of modules $\mathcal{M}=\{\mu_1, \mu_2,\ldots,\mu_m\}$ such that:

each $\mu_k \in \mathcal{M}$ is an *IPN* subnet: $\mu_k = (N_k, \Sigma_k, \lambda_k, \varphi_k)$, $k \in \{1,2,\ldots,m\}$ modules.

- $N_k = (P_k, T_k, I_k, O_k, M_{0k})$ where $P_k \subseteq P$, $T_k \subseteq T$, $I_k(O_k) : P_k \times T_k \rightarrow Z^+$, s.t., $I_k(p_i,t_j) = I(p_i,t_j)$ $(O_k(p_i,t_j) = O(p_i,t_j))$, $\forall\ p_i \in P_k$ and $\forall t_j \in T_k$ and $M_{0k} = M_0\,|_{Pk}$
- $\Sigma_k = \{\alpha\in\Sigma \mid \exists t_i,\ t_i \in T_k,\ \lambda(t_i) = \alpha\}$
- $\lambda_k : T_k \rightarrow \Sigma_k \cup \{\varepsilon\}$, s.t. $\lambda_k(t_i) = \lambda(t_i)$ and $t_i \in T_k$
- $\varphi_k : R(m_k, M_{0k}) \rightarrow (Z^+)^q$, q is restricted to the outputs associated to P_k. $\varphi_k = \varphi\big|_{Pk}$

For each μ_k the following conditions hold:

a) $\exists\mu_l \in\mathcal{M}$, s.t. $T_k \cap T_l \neq\varnothing$, $P_k\cap P_l = \{{}^\bullet t_i \cup t_i^\bullet \mid t_i \in \{T_k \cap T_l\}\}$, $P_k\cap P_l$ are measurable places.
b) $\forall p_i \in \{P_k - (P_k\cap P_l)\}$ if $p_i \in P^R$ then $p_i{}^{\bullet\bullet} \subset P_k$.
c) $ICom(OCom)$: $P_k \times T_l \rightarrow Z^+$, s.t. $I_k(p_i,t_j) = I_l(p_i,t_j)$ $(O_k(p_i,t_j) = O_l(p_i,t_j))$, $\forall p_i \in P_k$ and $\forall t_j \in T_l$. *ICom* and *OCom* represent the communication between modules. The arcs are depicted as a dashed line.

The obtained *DN* captures the firing language $\pounds(Q,M_0)$ in a distributed way, $\forall t_x \in\ \sigma = t_i t_j \ldots t_k \ldots$ and for every (α_x,y_x) in $\omega = (\alpha_0,y_0)(\alpha_1,y_1)\ldots(\alpha_n,y_n)$ $\exists\mu_k \in\mathcal{M}$ where t_x is fired and (α_x,y_x) is also generated in *DN*.

Consider the *IPN* system model depicted in the Figure 1 (for the sake of simplicity, we use in the examples the same names for duplicated nodes (places or transitions) belonging to different modules). Figure 3 presents the distributed *IPN*, m = 3 modules, *ICom* and *OCom* are represented by the dashed arcs. For example we can get the sets $T_1 \cap T_2 = \{t_3\}$ and $T_1 \cap T_3 = \{t_1\}$, $P_1 \cap P_2 = \{p_3\}$ and $P_1 \cap P_3 = \{p_{15}\}$.

We are preserving the property of event detectability using duplicated measurable places, which they establish the outputs that each module needs from others modules.

4.2 Local Reduced Models

The local models can be reduced following the steps of sub-section 3.2 and obtaining a simpler distributed model considering the local nodes.

Definition 4. Let $\mu_i \in\mathcal{M}$ be an *IPN* module. The local reduced model $\left(Q^{RM}, M_0^{RM}\right)_i$ is the subnet induced as in definition 1.

Consider the *DN* distributed model depicted in figure 3, the figure 4 presents the local reduced models where the place p_3 is duplicated in module 2 for detecting the firing of t_3. The communication between modules is represented by the dashed arcs.

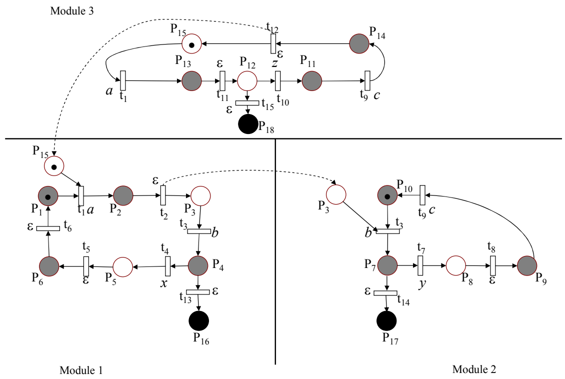


Figure 3: Distributed Interpreted Petri Net.

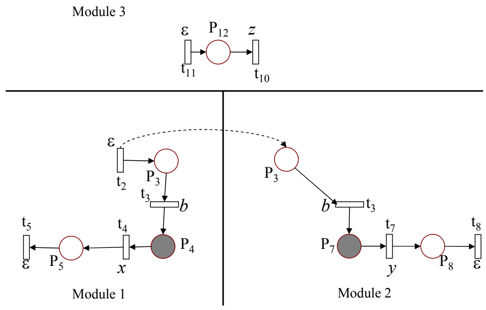


Figure 4: Local reduced models.

It is possible to obtain local reduced models where the communication is eliminated, since T^{RM}_n can be event-detectable only by the local outputs.

4.3 Modular Fault Detection

The error between the system output and the local diagnoser model output is $E_{kn}=\varphi\left(M_k\right)-\varphi_n\left(M_k^{RM}\right)$. The following algorithm, devoted to detect which local faulty marking was reached in *DN*, is executed when $E_{kn}\neq 0$ in $\mu_n \in\mathcal{M}$.

Algorithm 1. Detecting Local Faulty Markings

Inputs: $\varphi_n\left(M_k^{RM}\right)$, M_n^{RM}, $\lambda(t_i)$, $t_i \in T^{RM}_n$, E_{kn}

Outputs: p_n^F

1.Constants: φC_n^{RM} -- local reduced normal behaviour

2.Repeat

2.a. Read $\varphi_n(M_k^{RM})$ and $\lambda(t_i)$
2.b. If $\lambda(t_j) \in \lambda(P^{R\bullet})$ then computes
$\delta = \varphi_n(M_k^{RM})$ - $\varphi_n(M_{k-1}^{RM})$ (a column of φC_n^{RM})
2.c. i = index of the column of φC_n^{RM} , s.t.,
φC_n^{RM} $(\bullet,i) = \delta$, i.e. t_i was fired;
2.d. If $E_{kn} \neq 0$ then
- $\forall p_n \in ({}^{\bullet}t_i)^{\bullet\bullet} \cap P^F{}_n$, $M_{fn}(p_n)=1$
- Return $(p_n{}^F)$
- Sends to all modules the message "A fault occurred in module μ_n in place $(p_n{}^F)$".

Since $(Q^{RM}, M_0^{RM})_n$ is event detectable in ${}^{\bullet}P^R$ and $P^{R\bullet}$, then step 2.b. will compute just one column index; moreover, since $(Q^N, M_0^N)_n$ fulfils the conditions of theorem 1, then step 2.c. will compute just one place.

4.4 Distributed Input-output Diagnosability

The results of centralised diagnosability are applied to the modules issued from the partition.

The nodes of every $\mu_k \in \mathcal{M}$ are partitioned into local faulty nodes and normal nodes, i.e., $P_k = P^F{}_k \cup P^N{}_k$ and $T_k = T^F{}_k \cup T^N{}_k$.

$R(\mu_k, M_{0k})$ denotes the reachability set of a module μ_k and $LF = \{M_k \mid \exists p_j \in P^F{}_k$, such that $M_k(p_j)>0$, $M_k \in R(\mu_k, M_{0k})\}$ denotes the set of the local faulty markings.

$\Lambda^{int}{}_k(\mu_k, M_{0k})$ denotes the set of all input-output sequences that lead to a marking which puts a token into a duplicated place in other module μ_n, $\Lambda^{int}{}_k(\mu_k, M_{0k}) = \{\omega \mid \exists\sigma_m$, such that σ_m generates ω, and $M_{0m} \xrightarrow{\sigma_m} M_{jm}$ then M_{jm} marks a p_j s.t. $p_j \in P^{RM}{}_m$ in some module $\mu_m\}$.

Now, we introduce two notions for describing degrees of diagnosability in the modules of a distributed model.

A module is locally diagnosable if, for every local fault we can detect it only through local information, else it is conditionally diagnosable.

Definition 5. (Local Diagnosability) A module $\mu_n \in \mathcal{M}$ given by *DN* is said to be locally input-output diagnosable in $k < \infty$ steps if any marking $M_{fn} \in LF$ is distinguishable from any other $M_{kn} \in R(\mu_n, M_{0n})$ using any local word $\omega_n \in \Lambda^k{}_n(\mu_n, M_{0n}) \cup \Lambda_{Bn}(\mu_n, M_{0n})$.

Definition 6. (Conditional Diagnosability) A module $\mu_n \in \mathcal{M}$ given by *DN* is said to be conditional input-output diagnosable in $k < \infty$ steps if any marking $M_{fn} \in LF$ is distinguishable from any other $M_{kn} \in R(\mu_n, M_{0n})$ using any local word $\omega_m \in \Lambda^k{}_n(\mu_n, M_{0n}) \cup \Lambda_{Bn}(\mu_n, M_{0n})$ and any word $\omega_m \in \Lambda^{int}{}_n(\mu_n, M_{0n})$.

Proposition 1. Let (Q,M_0) be an *IPN* and *DN* its corresponding distributed *IPN* as stated in definition 3. If (Q,M_0) is input-output diagnosable as in theorem 1 then *DN* is distributed input-output diagnosable.

Proof. Assume that (Q,M_0) is input-output diagnosable. There exists a finite sequence of input-output symbols ω, s.t., $\omega \in \Lambda^k(Q,M_f) \cup \Lambda_B(Q,M_f)$, and $\sigma = t_i t_j t_k \ldots t_m$ is the firing transition sequence whose firing generates ω s.t. $M_0 \xrightarrow{\sigma_m} M_k$, $M_k \in F$. By theorem 1 M_k is distinguishable from any other $M_k \in R(Q,M_0)$ and (Q,M_0) is input-output diagnosable.

Since *DN* is the distributed behaviour of (Q,M_0), we suppose that the sequence σ can be fired in some modules $\mu_k \ldots \mu_l, \mu_m \in \mathcal{M}$ of *DN*, and the sequence generates the following local markings $M_{ik} \cup \ldots \cup M_{il} \cup M_{im}$, then $M_k = M_{ik} \cup \ldots \cup M_{il} \cup M_{im}$, s.t. $M_{ik} \ldots M_{il} \in LN$ and $M_{im} \in LF$. Let $\sigma_1, \sigma_2, \ldots, \sigma_m$ sequences s.t. $\sigma = \sigma_1\sigma_2\ldots\sigma_m$, suppose that σ_1 is fired in a module $\mu_k \in \mathcal{M}$ s.t. $M_{0k} \xrightarrow{\sigma_1} M_{ik}$, σ_2 is fired in $\mu_l \in \mathcal{M}$, s.t. $M_{0l} \xrightarrow{\sigma_2} M_{il} \ldots$, and σ_m is fired in $\mu_m \in \mathcal{M}$, s.t. $M_{0m} \xrightarrow{\sigma_m} M_{im}$, and σ occurs if the sequence σ_1 followed by a sequence $\sigma_2, \ldots$ followed by a sequence σ_m occur in the corresponding modules. Then by definition 5 and 6 μ_m can distinguish any $M_{im} \in LF$ from any other $M_{km} \in R(\mu_m, M_{0m})$. Hence there exists a module $\mu_m \in \mathcal{M}$ that can distinguish the corresponding faulty marking M_{im}; as μ_m can be any module and μ_m can be local or conditional input-output diagnosable, therefore *DN* is distributed input-output diagnosable.
□

Proposition 1 considers both cases (local and conditional diagnosable modules) for establishing the distributed input-output diagnosability of *DN*.

5 REDUCING INTERACTIONS

In Section 3.2 we explained how to build reduced models. Now, let us consider the following assumption:

- The manipulated input symbols $\lambda(t_k) \neq \varepsilon$ are not activated arbitrarily, only when they are enabled at the marking $M_k(p_k)>0$, s.t. $p_k \in {}^{\bullet}t_k$.

This assumption regards for building smaller reduced models.

Definition 7. Let (Q^N, M_0^N) be the embedded normal behaviour included in (Q,M_0). When the following condition holds: $\forall\lambda(t_k) \neq \varepsilon$, $t_k \in P^{R\bullet}$ are fired only when it is necessary, then the reduced model (Q^{RM}, M_0^{RM}) of (Q^N, M_0^N) of definition 1 is modified considering the following sets:

- $P^{RM} = P_a \cup P_b$, where $P_a = \{p_i \mid p_i \in P^R\}$ and $P_b = \{p_j \mid p_j \in P^{R\bullet\bullet}\}$;

- $T^{RM} = T_{in} \cup T_{out} \cup T_{af}$, where $T_{in} = \{{}^\bullet p_i \mid p_i \in P^{RM}\}$, $T_{out} = \{p_i{}^\bullet \mid p_i \in P^{RM}\}$ and $T_{af} = \{t_{edx} \mid t_{edx} \in {}^\bullet p_i$ and/or $t_{edx} \in p_i{}^\bullet$, t_{edx} is a new transition, $x = 1, 2, \ldots, z$ transitions non event-detectable$\}$, T_{af} is necessary only when $p_i \in P^{RM}$, such that p_i is non-measurable.
- $\lambda^{RM}: T^{RM} \to \Sigma \cup \{\varepsilon\}, \forall t_i' \in \{T_{in} \cup T_{out}\}, \lambda(t_i') = \lambda(t_i)$, $t_i \in T^N$, $t_i' = t_i$. If $t_i' \in T_{af}$, t_i' has no input symbols.
- $\varphi^{RM} = \varphi|_{R(Q^{RM}, M_0^{RM})}$
- $M_0^{RM} = M_0|_{P^{RM}}$. If $\exists p_k \in P^{RM}$, s.t., $M_k(p_k) = 0$, but, $p_k \in t_{ed}{}^\bullet$ then $M_k(p_k) > 0$.

The firing rules of $\left(Q^{RM}, M_0^{RM}\right)$ are defined as in definition 1 besides the following new firing rule:

- The transitions that belongs to T_{af} are fired automatically, i.e, $M({}^\bullet t_{ed}) > 0$ or $M(t_{ed}{}^\bullet) = 0$.

Figure 5 presents the distributed reduced model when we consider that the input symbols are not activated of an arbitrary way. We can see that the transition t_3 is not part of the reduced model of module 2, it is replaced by a transition t_{ed1}, $\lambda(t_{ed1}) = \varepsilon$. The goal for building smaller reduced models is to guarantee the observation of the system in critical situations.

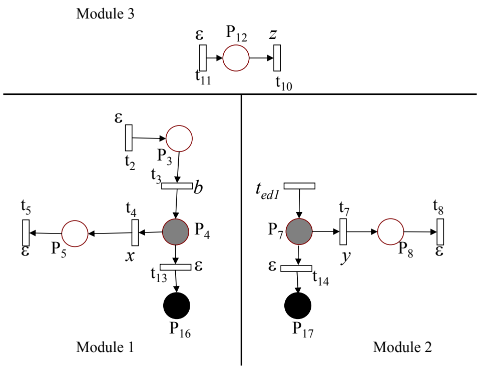


Figure 5: Reduced models for the centralised diagnoser.

6 CONCLUSIONS

A method for designing distributed diagnosers has been presented. The proposed model decomposition technique preserves the diagnosability of the global model into the distributed one and reduces the communication among the diagnosers. Current research addresses reliability of distributed diagnosers.

REFERENCES

Arámburo-Lizárraga J., E. López-Mellado, and A. Ramírez-Treviño (2005). "Distributed Fault Diagnosis using Petri Net Reduced Models". *Proc. of the IEEE International Conference on Systems, Man and Cybernetics*. pp. 702-707, October 2005.

Benveniste A., S. Haar, E. Fabre and C. Jara (2003). "Distributed and Asynchronous Discrete Event Systems Diagnosis". *42nd IEEE Conference on Decision and Control*. 2003.

Contant O, S. Lafortune and D. Teneketzis (2004). "Diagnosis of modular discrete event systems". *7th Int. Workshop on Discrete Event Systems* Reims, France. September, 2004.

Debouk R, S. Lafortune and D. Teneketzis (2000). "Coordinated Decentralized Protocols for Failure Diagnosis of Discrete Event Systems", Kluwer Academic Publishers, Discrete Event Systems: Theory and Applications, vol. 10, pp. 33-79, 2000.

Genc S. and S. Lafortune (2003). "Distributed Diagnosis of Discrete-Event Systems Using Petri Nets" *Proc. of the 24th. ATPN* pp. 316 - 336, June, 2003.

Genc S. and S. Lafortune (2005). "A Distributed Algorithm for On-Line Diagnosis of Place-Bordered Nets". 16th IFAC World Congress, Praha, Czech Republic, July 04-08, 2005.

Jalote P. (1994). *Fault Tolerance in distributed systems*. Prentice Hall. 1994

Jiroveanu G. and R. K. Boel (2003). "A Distributed Approach for Fault Detection and Diagnosis based on Time Petri Nets". *Proc. of CESA*. Lille, France, July 2003.

Pencolé Y. "Diagnosability analysis of distributed discrete event systems". *Proc. of the 15th International Workshop on Principles of Diagnosis*. Carcassonne, France. June 2004.

Ramírez-Treviño A., I. Rivera-Rangel and E. López-Mellado (2003). "Observability of Discrete Event Systems Modeled by Interpreted Petri Nets". *IEEE Transactions on Robotics and Automation*, vol. 19, no. 4, pp. 557-565, August 2003.

Ramírez-Treviño, E. Ruiz Beltrán, I. Rivera-Rangel, and E. López-Mellado (2004). A. Ramírez-Treviño, E. Ruiz Beltrán, I. Rivera-Rangel, E. López-Mellado. "Diagnosability of Discrete Event Systems. A Petri Net Based Approach". Proc. of the IEEE International Conference on Robotic and Automation. pp. 541-546, April 2004.

Ramírez-Treviño A, E. Ruiz Beltrán, I. Rivera-Rangel, E. López-Mellado (2007). "On-line Fault Diagnosis of Discrete Event Systems. A Petri Net Based Approach". *IEEE Transactions on Automation Science and Engineering*. Vo1. 4-1, pp. 31-39. January 2007.

HIGHER ORDER SLIDING MODE STABILIZATION OF A CAR-LIKE MOBILE ROBOT

F. Hamerlain, K. Achour
Laboratoire de Robotique et d'intelligence Artificielle, CDTA, Cité du 20 Août 1956, BP 17, Baba Hassen, Alger, Algeria
Hamerlainf@yahoo.fr, kachour@cdta.dz

T. Floquet, W. Perruquetti
LAGIS, UMR CNRS 8146, Ecole Centrale de Lille, BP 48, Cité Scientifique, 59651 Villeneuve-d'Ascq Cedex, France
thierry.floquet@ec-lille.fr, wilfrid.perruquetti@ec-lille.fr

Keywords: Nonholonomic systems, Higher order sliding modes, Finite time stabilization, Car-like mobile robot.

Abstract: This paper deals with the robust stabilization of a car-like mobile robot given in a perturbed chained form. A higher order sliding mode control strategy is developed. This control strategy switches between two different sliding mode controls: a second order one (super-twisting algorithm) and a new third order sliding mode control that performs a finite time stabilization. The proposed third sliding mode controller is based on geometric homogeneity property with a discontinuous term. Simulation results show the control performance.

1 INTRODUCTION

In the recent years, the control of nonholonomic systems has received a considerable attention, and in particular the stabilization problem. Due to the peculiar nature of nonholonomic kinematics, the stabilization problem addressed in control design for wheeled mobile robots (WMR) is in general quite difficult. In fact, it is known that nonholonomic WMR with restricted mobility (such as unicycle-type and car-like vehicles) cannot be stabilized to a desired configuration (or posture) via differentiable, or even continuous, pure-state feedback control despite they are open loop controllable (Brockett, 1983). Several nonlinear control designs have been proposed to achieve the stabilization for such systems. Time-varying feedbacks (Samson, 1995) or (open loop) sinusoidal and polynomial controls (Murray and Sastry, 1993) can be developed. Other alternatives consist in using the backstepping recursive techniques (Jiang and Nijmeijer, 1999), (Huo and Ge, 2001), flatness (M. Fliess *et al.*, 1995), or discontinuous approaches (Astolfi, 1996), (Floquet *et al.*, 2003). The robustness property is an important aspect for stabilizing tasks of uncertain systems, especially when there exist disturbances or errors dynamics in the system. It is well known that the standard sliding mode features are high accuracy and robustness with respect to various internal and external disturbances. The basic idea is to force the state via a discontinuous feedback to move on a prescribed manifold called the sliding manifold. A specific drawback involved by sliding mode technique is the well known chattering effect (undesirable vibrations), which limits the practical relevance. To overcome this drawback, the Higher Order Sliding Mode (HOSM) approach has been proposed (Emel'yanov *et al.*, 1993). The main objective is to keep the sliding variable and a finite number of its successive time derivatives to zero through a discontinuous function acting on some high order time derivative of the sliding variable. This technique generalizes the basic sliding mode idea and can be implemented for systems with arbitrary relative degree. Keeping the main advantages of the standard sliding mode control, the chattering effect is avoided and finite time convergence together with higher order precision are provided. Actually, the problem of higher order sliding mode control is equivalent to the finite time stabilization of an integrator chain with nonlinear uncertainties. In (Floquet *et al.*, 2003), it is shown that the HOSM theory is efficient to design control laws which robustly stabilizes in finite time a chained form system. Second order sliding mode controllers were proposed to stabilize a three-dimensional system (unicycle type vehicle). It should be pointed out that, in the case of the four dimensional car-like robot system, the proposed procedure requires the finite time stabilization of a third order integrator chain. Thus, a third order sliding mode control is at least necessary and a new type of third order sliding mode algorithm is introduced in this paper.

The aim of this paper is to present a high order sliding mode control strategy for the robust stabilization problem of a car-like mobile robot. First, the perturbed one-chain form of the robot is derived. Then, second order sliding mode controllers based on the so-called super twisting algorithm and a third order sliding mode controller are developed. The latter controller is a combination of a finite time controller based on geometric homogeneity and a discontinuous term that ensures robustness properties. By switching between these sliding mode controllers, a finite time stabilization to the origin is obtained.

The organization of this paper is as follows: Section 2 presents the perturbed chained form model of the car-like vehicle and states the problem under interest. Section 3 deals with the design of the hybrid control law strategy via higher order sliding mode technique. Simulation results are presented in Section 4.

2 CAR-LIKE ROBOT MODEL AND PROBLEM STATEMENT

As mentioned in (Murray and Sastry, 1993), many nonlinear mechanical systems that belong to the class of driftless nonholonomic systems (the knife-edge, articulated vehicles, a car towing several trailers, etc...) can be transformed via change of coordinates in the state and control spaces into a so-called chained form. In this paper, we are particularly concerned with nonholonomic systems whose trajectories can be written as the solution of the driftless system:

$$\dot{x} = g_1(x)u_1 + g_2(x)u_2 \tag{1}$$

where $x \in \Re^n$ is the state vector, u_1, $u_2 \in \Re$ are the two control inputs, g_1, g_2 are smooth linearly independent vector fields. We are interested in the case of $n = 4$ which is the example of a car-like mobile robot. The kinematic model of the robot in a single drive mode is given by:

$$\begin{cases} \dot{x} = \cos(\theta)\ u_1 \\ \dot{y} = \sin(\theta)\ u_1 \\ \dot{\theta} = \frac{\tan(\phi)}{L} u_1 \\ \dot{\phi} = u_2 \end{cases} \tag{2}$$

where (x,y) are the Cartesian coordinates of the center of the rear axle, θ is the orientation angle of the vehicle with respect to a fixed frame, ϕ is the steering angle relative of the car body and u_1, u_2 are respectively the driven and the steering velocities.

For $\theta \in \left]-\frac{\pi}{2}, \frac{\pi}{2}\right[$, let us consider the following transformation on the control input vector:

$$w = \begin{pmatrix} \cos(\theta) & 0 \\ 0 & 1 \end{pmatrix} u$$

and let us introduce perturbations in the model. Then, the behavior of the robot can be described by the following system:

$$\dot{q} = g_1 w_1 + g_2 w_2 + \gamma(x,t) \tag{3}$$

with $q = (x, y, \theta, \phi)^T$, $g_1(q) = \left(1, \tan(\theta), \frac{\tan(\phi)}{L\cos(\theta)}, 0\right)^T$, $g_2(q) = (0,0,0,1)^T$. $\gamma(x,t) \in \Re^n$ is an additive perturbation assumed to be smooth enough. In (Murray *et al.*, 1994), conditions are given for a nonholonomic systems (1) to be transformed into a so-called one-chained form.

By using the diffeomorphism and the control input space transformation given in (Murray *et al.*, 1994)

$$\begin{cases} z_1 = x \\ z_2 = \frac{\tan(\phi)}{L\cos(\theta)^3} \\ z_3 = \tan(\theta) \\ z_4 = y \end{cases} \tag{4}$$

$$\begin{cases} v_1 = w_1 = u_1 \cos(\theta) \\ v_2 = \frac{3\tan(\phi)^2 \sin(\theta)}{L^2\cos(\theta)^3} w_1 + \frac{1}{L\cos(\theta)^3\cos(\phi)^2} w_2 \end{cases}, \tag{5}$$

for $\phi \in \left]-\frac{\pi}{2}, \frac{\pi}{2}\right[$, the system (3) is transformed in the following perturbed one-chained form

$$\begin{cases} \dot{z}_1 = v_1 + \gamma^1(z,t) \\ \dot{z}_2 = v_2 + \gamma^2(z,t) \\ \dot{z}_3 = z_2\ (v_1 + \gamma^1(z,t)) \\ \dot{z}_4 = z_3\ (v_1 + \gamma^1(z,t)) \end{cases} \tag{6}$$

if and only if $\gamma(x,t)$ belongs to the distribution spanned by $g_1(x)$ and $g_2(x)$ (Floquet *et al.*, 2000).

In the following, γ^1 and γ^2 are supposed to be bounded for all $z \ \varepsilon\ \Omega$ (an open set in $\Re^4$) and for all t as follows:

$$\begin{aligned} \left|\gamma^1(z,t)\right| &\leq \sigma_1 \\ \left|\gamma^2(z,t)\right| &\leq \sigma_2 \\ \left|\dot{\gamma}^1(z,t)\right| &\leq \sigma_1^{'} \end{aligned}$$

where σ_1, σ_2, $\sigma_1^{'}$ are positive constants.

Problem: Find a robust control law for the car-like robot model (6) guaranteeing the finite time stabilization at the origin $(x = 0,\ y = 0,\ \theta = 0,\ \phi = 0)$ in the presence of matched disturbances.

3 FINITE TIME STABILIZATION VIA HIGHER ORDER SLIDING MODES

3.1 Higher Order Sliding Modes

In this approach, the designed control law will switch between second order and third order sliding mode algorithms in order to obtain the finite time stabilization of (6). It is assumed that the reader is familiar with the sliding mode theory (see (Emel'yanov *et al.*, 1993), (Fridman and Levant, 2002) or (Perruquetti and Barbot, 2002) for further details). Let us just briefly recall that the principle of higher order sliding mode control is to constrain, by the mean of a discontinuous control acting on the r^{th} time derivative of a suitably chosen sliding variable $S : \Re^+ \times \Re^n \rightarrow \Re$, the system trajectories to reach and stay, after a finite time, on a given sliding manifold S^r in the state space defined by:

$$\mathcal{S}^r = \left\{ S(t,x) = \dot{S}(t,x) = ... = S^{(r-1)}(t,x) = 0 \right\},$$

where $x \in \Re^n$ is the state system. A control law leading to such a behavior will be called a r^{th} order ideal sliding mode algorithm with respect to S.

Arbitrary-order sliding mode controllers with finite time convergence have been proposed in (Levant, 2001) and (Levant, 2003). As the control algorithm proposed in (Levant, 2001) requires the knowledge of high order time derivatives of the output, the author in (Levant, 2003) proposes the use of a robust exact finite time convergence differentiators based on the super twisting algorithm. The implementation of these controllers is not easy since some singularities in the time derivatives of the sliding variable may appear. In (Laghrouche *et al.*, 2004), a third order sliding mode controller that combines a standard sliding mode control with a linear quadratic one has been proposed. However, it directly depends on the initial conditions of the system and complex off-line computations are needed before starting the control action. A higher order sliding mode control strategy with smooth manifold leading to a practical convergence was developed in (Djemai and Barbot, 2002). Based on this strategy, a real third order sliding mode controller with time varying smooth manifolds was designed for the practical stabilization of a unicycle-type mobile robot (Barbot *et al.*, 2003).

3.2 Finite Time Stabilization of the Car-like Robot

The stabilization of (6) is made in three steps by switching between different types of sliding mode controllers:

First step:

The control algorithm is first to constrain the subsystem:

$$\dot{z}_1 = v_1 + \gamma^1(z,t) \tag{7}$$

to evolve after a finite time on the sliding manifold

$$s_1 = z_1 - at = 0,\ a > 0.$$

The subsystem (7) has relative degree one with respect to s_1 and the second time derivative of s_1 is given by:

$$\ddot{s}_1 = \dot{v}_1 + \dot{\gamma}^1(z,t).$$

The chosen sliding mode algorithm is the super twisting algorithm which has been developed for systems with relative degree one to avoid chattering. The control law v_1 is given as follows:

$$\begin{array}{l} v_1 = -\lambda_1 |s_1|^{\frac{1}{2}} \operatorname{sign}(s_1) + v_{11}, \\ \dot{v}_{11} = -\alpha_1 \operatorname{sign}(s_1), \end{array} \tag{8}$$

where α_1, λ_1 are positive constants that satisfy the following conditions (Levant, 2003):

$$\begin{array}{rcl} \alpha_1 & > & \sigma_1^{'} \\ \lambda_1^2 & > & 4\sigma_1^{'} \dfrac{\alpha_1 + \sigma_1^{'}}{\alpha_1 - \sigma_1^{'}} \end{array}$$

This ensures that the trajectories reach the sliding manifold $\Gamma_1 = \{ z \in \Omega : s_1 = \dot{s}_1 = 0 \}$ in a finite time T_1 and stay it after T_1. Thus, for $t \geq T_1$, the resulting dynamics, in sliding motion, is given by:

$$\left\{ \begin{array}{l} \dot{z}_1 = a, \\ \dot{z}_2 = v_2 + \gamma^2(z,t), \\ \dot{z}_3 = az_2, \\ \dot{z}_4 = az_3. \end{array} \right. \tag{9}$$

Second step:

When the state trajectory evolves in Γ_1, the dynamics of the subsystem

$$\left\{ \begin{array}{l} \dot{z}_2 = v_2 + \gamma^2(z,t), \\ \dot{z}_3 = az_2, \\ \dot{z}_4 = az_3. \end{array} \right. \tag{10}$$

is equivalent to a perturbed triple chain of integrator. The finite time stabilization of (10) can be obtained using a 3^{rd} order sliding mode algorithm for v_2. A new kind of algorithm is presented here. The control law v_2 is made of two terms:

$$v_2 = v_{2,id} + v_{2,vss}$$

where $v_{2,id}$ is an ideal control, based on the geometric homogeneity approach, and that ensures the finite time stabilization of the system (9) without perturbations. $v_{2,vss}$ is a discontinuous part of the control v_2 allowing to reject the uncertainties.

a. Control design of $v_{2,id}$

Consider the system (10) without perturbations:

$$\begin{cases} \dot{z}_2 = v_{2,id} \\ \dot{z}_3 = az_2 \\ \dot{z}_4 = az_3 \end{cases} \tag{11}$$

and let us define a control law $v_{2,id}$ stabilizing $\bar{z}=(z_2,\ z_3,\ z_4)^T$ to zero in finite time.

To this end, let $k_1, k_2, k_3 > 0$ be such that the polynomial $p^3 + k_3 p^2 + k_2 p + k_1$is Hurwitz. From the works (Bhat and Bernstein, 2005), there exists $\varepsilon \in (0,1)$ such that, for every, $\beta \in (1-\varepsilon, 1)$, the origin is a globally finite time stable equilibrium for (11) via the state feedback:

$$\begin{aligned} v_{2,id} &= -k_1 \text{sign}(z_4)\,|z_4|^{\beta_1} - k_2 \text{sign}(z_3)\,|z_3|^{\beta_2} \\ &\quad -k_3 \text{sign}(z_2)\,|z_2|^{\beta_3} \end{aligned} \tag{12}$$

with

$$\begin{cases} \beta_{i-1} = \frac{\beta_i \beta_{i+1}}{2\beta_{i+1} - \beta_i},\ i = 2,3 \\ \beta_3 = \beta,\ \beta_4 = 1 \end{cases}$$

This ensures that the following equalities hold after a finite time $T_{2,i}$:

$$z_2 = z_3 = z_4 = 0.$$

b. Control design of $v_{2,vss}$

For perturbation rejection, the following sliding variable $s \in \Re$ is introduced:

$$s = s_0(z_2, z_3, z_4) + s_2 \tag{13}$$

Here, s_0 is a quite conventional sliding mode variable, selected such that $\frac{\partial s_0}{\partial z}[1,0,0]^T \neq 0$ (relative degree one requirement). One can choose

$$s_0 = z_2.$$

s_2 is an additional term that enables integral control to be included such that

$$\dot{s}_2 = -v_{2,id}.$$

Let us show that a sliding motion can be induced on $s = 0$ by using the discontinuous control

$$v_{2,vss} = -D\text{sign}(s) \tag{14}$$

where the switching gain satisfies:

$$D > \sigma_2.$$

For this, define a Lyapunov function V as follows:

$$V = \frac{1}{2}s^2$$

The time derivative of this function is given by:

$$\begin{aligned} \dot{V} &= s(v_2 + \gamma^2 + \dot{s}_2) \\ &= s(v_{2,id} + v_{2,vss} + \gamma^2(z,t) - v_{2,id}) \\ &= s(\gamma^2(z,t) - D\text{sign}(s)) \\ &\leq \sigma_2\,|s| - D\ |s| \\ &\leq -G\sqrt{V}, \quad G > 0 \end{aligned}$$

Hence, the trajectories of the system converge in a finite time $T_{2,v}$ on the sliding manifold given by $\{s = 0\}$. When sliding, the equivalent control denoted by $v_{2,eq}$ (see (Edwards and Spurgeon, 1998), p. 34 for a definition), required to maintain the sliding motion on the surface $s = 0$, is obtained by writing that $\dot{s} = 0$:

$$\begin{aligned} \dot{s} &= \dot{s}_0 + \dot{s}_2 \\ &= v_{2,eq} + v_{2,id} + \gamma^2(z,t) - v_{2,id} \\ &= 0 \end{aligned} \tag{15}$$

Thus:

$$v_{2,eq} = -\gamma^2(z,t)$$

and the equivalent dynamics on $s = 0$ is given by the system (11). This implies that the trajectory enters the set

$$\Gamma_2 = \{\ z \in \Omega : z_2 = z_3 = z_4 = 0\}$$

after a finite time $T_2 = T_{2,i} + T_{2,v}$.

Third step:

When $z \in \Gamma_1 \cap \Gamma_2$, the controls are switched to:

$$\begin{cases} v_1 = -\lambda_{11}\,|z_1|^{\frac{1}{2}}\,\text{sign}(z_1) + w_{11} \\ \dot{w}_{11} = -\alpha_{11}\text{sign}(z_1) \end{cases} \tag{16}$$

$$\begin{cases} v_2 = -\lambda_{22}\,|z_2|^{\frac{1}{2}}\,\text{sign}(z_2) + w_{22} \\ \dot{w}_{22} = -\alpha_{22}\text{sign}(z_2) \end{cases} \tag{17}$$

With a suitable choice of the positive constants α_{11}, λ_{11}, α_{22} and λ_{22} (see (Levant, 2003)), a sliding motion is obtained on the manifolds $\{z_1 = \dot{z}_1 = 0\}$ and $\{z_2 = \dot{z}_2 = 0\}$ after a finite time T_3. This allows to maintain z_2 equals to zero (and so $z_3 = z_4 = 0$), and to reach the manifold $z_1 = 0$, so that the global system is stabilized in a finite time lower than $T_1 + T_2 + T_3$.

4 SIMULATION RESULTS

In this section, simulation results for the finite time stabilization of the car-like robot are presented. The sampling time is set to be $\tau = 0.01s$ with the physical parameter $L = 1.2m$. The design parameters of the second order sliding mode controllers are:

$$a = 3,\ \lambda_1 = 5,\ \lambda_{11} = \lambda_2 = 1,\ \alpha_1 = \alpha_{11} = \alpha_{22} = 0.5$$

while for the third order one, they are:

$$\begin{aligned} k_1 &= 1, k_2 = k_3 = 1.5, D = 0.97, \\ \beta_1 &= 2/3, \beta_2 = 3/5, \beta_3 = 3/4. \end{aligned}$$

The perturbations are taken as band limited white noises with unit variance. The results of the stabilization are given in Figures 1, 2, 3 with the initial conditions:

$$z_1 = x = 0.5m, z_2 = z_3 = 0, z_4 = y = 1.4m.$$

Figures 1, 2 and 3 show that the control problem is fulfilled since the state trajectory of the robot converges to zero in a robust manner. One can note (Figure 1) that z_4 tends to zero ($\approx 5s$) faster than z_3 ($\approx 5.2s$), and z_2 ($\approx 5.4s$). Once $z_2 = 0$, the control v_1 switches in the third step and z_1 reaches the origin in finite time $(13s)$. Figure 2 gives the angles behaviour and the trajectory of the robot in the phase plan (x, y), while Figure 3 shows the movement of the robot. The choice of a second order sliding mode controller (super twisting algorithm) in the first and third steps allows to overcome the chattering phenomenon since the discontinuity is acting on the first time derivative of the control v_1. Indeed, the trajectory of the system is smoother as there are few uncertainties on the information injected in the equivalent dynamics in the second step. Also, it can be seen on the behavior of the actual control inputs (Figure 4), that the driven velocity is not affected by the chattering effect. However, the steering velocity still exhibits some chattering in the second step due to the discontinuous part of the control v_2. In practice, the chattering phenomenon can be reduced by using sigmoïd functions instead of the signum function. Another solution would be the use of a second order sliding mode algorithm in (14).

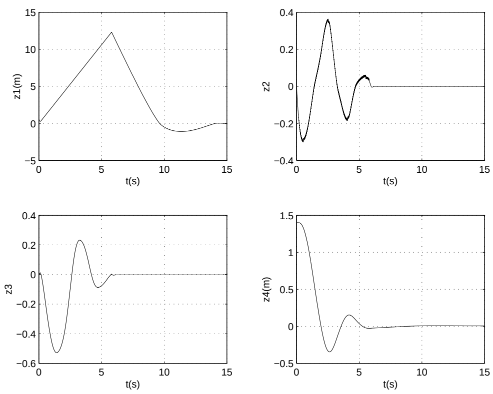


Figure 1: Coordinates z_1, z_2, z_3, z_4.

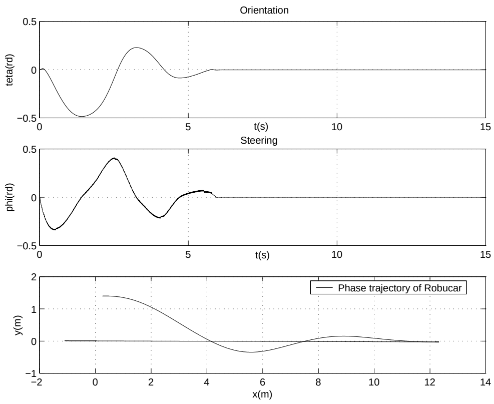


Figure 2: Angles and phase trajectory.

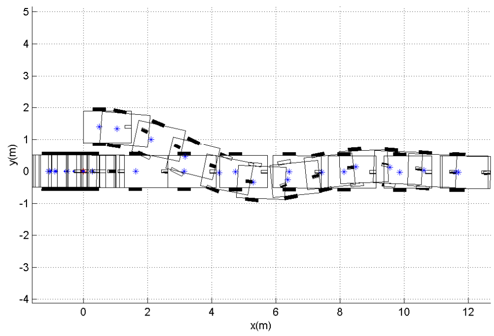


Figure 3: Movement of the Robot.

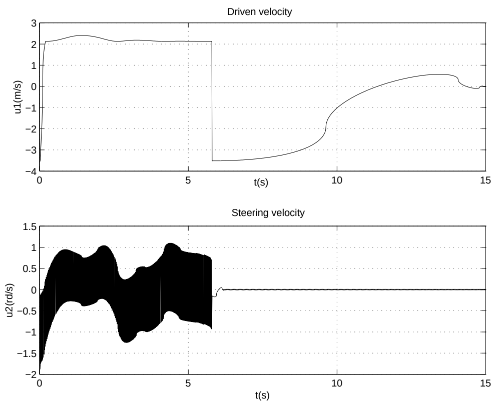


Figure 4: Actual control inputs.

5 CONCLUSION

This paper has presented a higher order sliding mode control solution for the robust stabilization problem applied to a car-like robot. Based on the perturbed chain form of the robot, control laws, switching between different higher order sliding mode controllers, have been developed to obtain a robust finite time stabilization. One contribution of this paper is the design of a 3^{rd} order sliding mode control based on geometric homogeneity property with a discontinuous term. Future work concerns the experimental test of the proposed control approach.

REFERENCES

Astolfi A. (1996). Discontinuous control of nonholonomic systems, *Systems & Control Letters* Vol. 27, pp. 37–45.

Barbot J.-P., Djemai M., Floquet T. and Perruquetti W. (2003). Stabilization of a unicycle-type mobile robot using higher order sliding mode control. In *Proc. of the 41st IEEE Conference on Decision and Control*.

Bhat, S. and Bernstein, D. (2005). Geometric Homogeneity with applications to finite time stability. *Mathematics of Control, Signals and Systems*, Vol. 17, pp. 101–127.

Brockett, R. (1983). Asymptotic stability and feedback stabilization. In *Differential geometric control theory*, pp. 181–191, Birkhauser.

Djemai M. and Barbot J.-P. (2002). Smooth manifolds and high order sliding mode control. In *Proc. of the 41st IEEE Conference on Decision and Control*.

Edwards C. and Spurgeon S. (1998). *Sliding mode control: theory and applications*. Taylor and Francis Eds.

Emel'yanov S.V., Korovin S.V., and Levantovsky L.V. (1993). Higher order sliding modes in control systems. *Differential Equations*, Vol. 29, pp. 1627–1647.

Floquet T., Barbot J.-P. and Perruquetti W. (2000). One-chained form and sliding mode stabilization for a nonholonomic perturbed system. *American Control Conference*, Chicago, USA.

Floquet T., Barbot J.-P. and Perruquetti W. (2003). Higher order sliding mode stabilization for a class of non holonomic perturbed system. *Automatica*, Vol. 39, pp. 1077–1083.

Fliess M., Levine J., Martin P. and Rouchon P. (1995). Flatness and defect of non-linear systems: Introductory theory and examples. *International Journal on Control*, Vol.61, pp. 1327–1361.

Fridman L. and Levant A. (2002). Higher order sliding modes. In *Sliding Mode Control in Engineering*, W. Perruquetti and J. P. Barbot (Eds), Marcel Dekker, pp. 53–101.

Huo W. and Ge S. (2001). Exponential stabilization of nonhononomic systems: an ENI approach. *International Journal of Control*, Vol. 74, 1492–1500.

Jiang, Z. and Nijmeijer, H. (1999). A recursive technique for tracking control of nonholonomic systems in chained form. *IEEE Tansactions on Automatic Control*, Vol. 44, pp. 265–279.

Laghrouche S., Smaoui M. and Plestan F. (2004). Third-order sliding mode controller for electropneumatic actuators. In *Proc. of the 43rd IEEE Conference on Decision and Control*.

Levant, A. (2001). Universal SISO sliding-mode controllers with finite-time convergence. *IEEE Transactions on Automatic Control*, Vol. 46, pp. 1447–1451.

Levant, A. (2003). Higher-order sliding modes, differentiation and output-feedback control. *International Journal of Control*, Vol. 76, pp. 924–941.

Murray R. and Sastry S. (1993). Nonholonomic motion planning :steering using sinusoids. *IEEE. Tansactions On Automatic Control*, Vol. 38, pp. 77–716.

Murray R., Li Z. and Sastry S. (1994). *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Inc., Florida, USA.

Perruquetti W. and Barbot J.-P. (Editors) (2002), *Sliding Mode Control in Engineering*, Marcel Dekker.

Samson C. (1995). Control of chained systems. Application to path following and time-varying point stabilization of mobile robots. *IEEE Tansactions on Automatic Control*, Vol. 40, pp. 64–77.

DIRECTIONAL MANIPULABILITY FOR MOTION COORDINATION OF AN ASSISTIVE MOBILE ARM

K. Nait-Chabane, P. Hoppenot and E. Colle
IBISC Université d'Evry Val d'Essonne, 40 rue de plevoux 91020, Evry, France
chabane@iup.univ-evry.fr , hoppenot@iup.univ-evry.fr, colle@iup.univ-evry.fr

Keywords: Mobile manipulator, manipulability, redundancy systems, robotic assistance.

Abstract: In this paper, we address the problem of coordinated motion control of a manipulator arm embarked on a mobile platform. The mobile manipulator is used for providing assistance for disabled people. In order to perform a given task by using mobile manipulator redundancy, we propose a new manipulability measure that incorporates both arm manipulation capacities and the end-effector imposed task. This measure is used in a numerical algorithm to solve system redundancy and then compared with other existing measures. Simulation and real results show the benefit and efficiency of this measure in the field of motion coordination.

1 INTRODUCTION

In assistive robotics, a manipulator arm constitutes one possible solution for restoring some manipulation functions to victims of upper limb disabilities. The literature proposes three distinct manipulator arm configurations. The first one consists of a workstation in which a manipulator arm evolves within a structured environment (RAID, AFMASTER (Busnel, 2001)). In the second configuration, a manipulator arm is added to an electrical wheelchair ((Kwee, 1993), (Evers, 2001)). The third configuration aims at expanding the field of application of manipulator arm by making it mobile, MoVAR (Van der Loos, 1995), URMAD-MOVAID (Dario, 1999) and ARPH (Hoppenot, 2002), that offers many advantages. Our research deals with this third configuration. Such a system possesses more degrees of freedom than necessary for task execution. Any given point in the workspace may be reached by moving the manipulator arm or moving the mobile platform or by a combination of both. To facilitate the use of the system by the handicapped person, the idea is that the operator pilots the gripper and that the remainder of the articulated system follows. We have focused attention on the use of redundancy for controlling a mobile arm.

Manipulability measures play an important role in the design, analysis, evaluation and optimization in manipulation robotics; it is a scalar that quantifies how well the system behaves with respect to force and motion transmission. These measures however do not include information either on the task imposed or on the direction of end-effector motion. We propose an additional measure that takes the task to be performed into account.

This paper is organized as follows. Section 2 presents the work conducted towards devising different solutions to the redundancy problem. Section 3 discusses the mobile arm and its kinematic models before Section 4 introduces the manipulability concept and primary set of related measures used in the literature on robotics. Section 5 then lays out a new measure that takes both the system manipulation capacities and task in progress into account. We will recall the principle behind the kinematic control scheme used to solve redundancy problems in Section 6, followed by an illustration of the benefit of our new measure by means of simulation and real results (in Section 7).

2 RELATED RESEARCH WORK

A considerable amount of interest has been shown over the past few years in mobile manipulators. Seraji (Seraji, 1993) presents a simple online approach for the motion control of mobile manipulators using augmented Jacobian matrices. This kinematic approach requires additional constraints to be satisfied for the manipulator configuration. The approach proposed may be applied with equal ease to both nonholonomic and holonomic mobile robots. Yamamoto and Yun (Yamamoto, 1987) set out to decompose the motion of the mobile manipulator into two subsystems based on the concept of preferred operating region. The mobile platform is controlled so as to bring the manipulator into a preferred operating region/configuration, with respect to the mobile platform, as the manipulator performs a variety of unknown manipulation tasks. The authors used the manipulability measure of the manipulator arm to define this preferred operating region. The principal advantage of this approach lies in its decentralized planning and control of the mobile platform and manipulator arm. However, the case when the manipulator is mounted at the center of the axis between the two driving wheels lies at a singularity in the method proposed by Yamamoto and Yun (Yamamoto, 1987). Nagatani (Nagatani, 2002) developed an approach to plan mobile base's path which satisfies manipulator's manipulability. The controllers used for manipulation and locomotion differ from one another.

Khatib (Khatib, 1995) Khatib [10] proposed to use a joint limit constraint in mobile manipulation in the form of potential function while his approach is to use the inherent dynamics characteristics of mobile manipulator in operational space. Additionally, he analyzed the inertial properties of a redundant arm with macro-micro structure. Kang (Kang, 2001) derived a combined potential function algorithm to determine a posture satisfying both the reduced inertia and joint limit constraints for a mobile manipulator. The author then integrated the inertia property algorithm into a damping controller in order to reduce the impulse force upon collision as well as to regulate contact.

Yoshikawa (Yoshikawa, 1990) introduced the arm manipulators manipulability and used it to solve the redundancy of such systems. The manipulability of mobile manipulator has been studied by few authors. Yamamoto and Yun (Yamamoto, 1999) have treated both locomotion and manipulation within the same framework from a task space perspective. They have presented the kinematic and dynamic contributions to manipulators and platforms by means of the so-called task space ellipsoid.

Bayle (Bayle, 2001) extended the definition of manipulability to the case of a mobile manipulator and then applied it in an inversion process for solving redundancy.

3 DESCRIPTION OF ROBOTIZED ASSISTANT

The mobile manipulator consists of a Manus arm mounted on a mobile platform powered by two independent drive wheels. Let's start by defining a fixed world reference frame $\{W\}$, a moving platform frame $\{P\}$ attached to the midway between the two drive wheels, a moving arm frame $\{A\}$ related to the manipulator base, and a moving end-effector frame $\{E\}$ attached to the arm end-effector (see Fig. 1).

We will adopt the following assumptions in modeling the mobile manipulator system: no slipping between wheel and floor; a platform incapable of moving sideways in order to maintain the nonholonomic constraint; and a manipulator rigidly mounted onto the platform.

The forward kinematics of a serial chain manipulator relating joint space and task space variables is expressed by:

$$X_a = f_a(q_a) \tag{1}$$

where $X_a = [x_{a1}, x_{a2}, \cdots x_{am}]^T \in R^m$ is the vector of task variables in an m-dimensional task space, $q_a = [q_{a1}, q_{a2}, \cdots q_{an}]^T \in R^n$ is the vector of joint variables in n-dimensional variables (called the generalized coordinates), and f_a is the nonlinear function of the forward kinematic mapping

$$\dot{X}_a = J_a(q_a)\dot{q}_a \tag{2}$$

where $\dot{X}_a$ is the task velocity vector, $\dot{q}_a$ the joint velocity vector and $J_a(q_a)$ the Jacobian matrix.

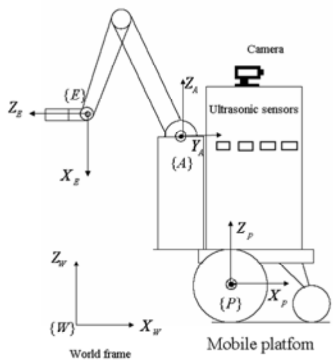


Figure 1: Mobile manipulator system.

For the kinematic modeling of the considered manipulator arm, we make use of Denavit-Hartenberg parameters (Sciavicco, 1996). Manus arm possesses six rotoid joints, with 3 DOF for gripper positioning and 3 DOF for gripper orientation. The Cartesian coordinates of the end-effector relative to the arm base frame $\{A\}$ are given by:

$$X_a = \begin{cases} x_{a1} = (L_4 c_3 + L_3 c_2) c_1 - L_2 s_1 \\ x_{a2} = (L_4 c_3 + L_3 c_2) s_1 + L_2 c_1 \\ x_{a3} = L_4 s_3 + L_3 s_2 \\ x_{a4} = \phi \\ x_{a5} = \theta \\ x_{a6} = \psi \end{cases} \quad (3)$$

where $c_i = \cos(q_{ai})$, $s_i = \sin(q_{ai})$ and L_2, L_3, L_4 represent the length of shoulder, upper arm and lower arm, respectively.

$[x_{a1}, x_{a2}, x_{a3}]^T$ and $[\phi, \theta, \psi]^T$ represent the Cartesian coordinates and Euler angles of the end-effector, respectively. In this paper, we will only consider the three main joints of the arm, as given by the generalized vector $q_a = [q_{a1}, q_{a2}, q_{a3}]^T$.

The platform location is given by three operational coordinates x_p, y_p and θ_p, which define its position and orientation. The generalized coordinate vector is thus: $q_p = [x_p, y_p, \theta_p]^T$ and the generalized velocity vector is: $\dot{q}_p = [\dot{x}_p, \dot{y}_p, \dot{\theta}_p]$.

The constraint equation applied to the platform has the following form:

$$A(q_p)\dot{q}_p = 0 \quad (4)$$

in which $A(q_p) = [\sin(\theta_p) \; -\cos(\theta_p) \; 0]$.

The kinematic model of the mobile platform is given in (Campion, 1996):

$$\dot{q}_p = S(q_p) u_P \quad (5)$$

where $S(q_p) = \begin{bmatrix} \cos(\theta_p) & 0 \\ \sin(\theta_p) & 0 \\ 0 & 1 \end{bmatrix}$ and $u_p = [v, \omega]^T$, with v and ω being the linear and angular velocities of the platform, respectively.

The forward kinematic model of the mobile manipulator may be expressed in the following form:

$$X = f(q_p, q_a) \quad (6)$$

where q_p is the generalized coordinates of the mobile platform and q_a the joint variables of the arm, as defined above.

The configuration of the mobile manipulator is therefore defined by the N generalized coordinates ($N=n+3$):

$$q = [q_p^T, q_a^T]^T = [x_p, y_p, \theta_p, q_{a1}, \cdots, q_{an}]^T \quad (7)$$

The direct kinematic model for the positioning task of the considered mobile arm relative to world frame $\{W\}$ is given by:

$$X = [x_1, x_2, \cdots, x_6]^T = f(q_a, q_p) \quad (8)$$

$$X = \begin{cases} x_1 = x_p + (x_{a2} + a) c_{\theta_p} - (b - x_{a1}) s_{\theta_p} \\ x_2 = y_p + (x_{a2} + a) s_{\theta_p} + (b - x_{a1}) c_{\theta_p} \\ x_3 = x_{a3} + c \\ x_4 = x_{a4} + \theta_p - \dfrac{\pi}{2} = \phi + \theta_p - \dfrac{\pi}{2} \\ x_5 = x_{a5} = \theta \\ x_6 = x_{a6} = \psi \end{cases} \quad (9)$$

where $c_{\theta_p} = \cos(\theta_P)$, $s_{\theta_p} = \sin(\theta_P)$; a, b and c are the Cartesian coordinates of the manipulator arm base with respect to the mobile platform frame $\{P\}$.

The instantaneous kinematic model is then given by:

$$\dot{X} = J(q)\dot{q} \quad (10)$$

with: $J(q) = \dfrac{\partial f}{\partial q}$.

We can observe that generalized velocities $\dot{q}$ are dependent; they are linked by the nonholonomic constraint. The platform constraint described by (4) can be written in the following form:

$$[A(q_p) \;\; 0]\dot{q} = 0 \quad (11)$$

According to (5), the relation between the generalized velocity vector of the system and its control velocities can be written as follows:

$$\dot{q} = M(q) u \quad (12)$$

where $M(q)=\begin{bmatrix} S_p(q_p) & 0 \\ 0 & I_n \end{bmatrix}$.

I_n is an n-order identity matrix and $u=[v, w, \dot{q}_{a1}, \cdots, \dot{q}_{an}]^T$.

The instantaneous kinematic model does not include the nonholonomic constraint of the platform given by (11). The relationship between the operational velocities of the mobile manipulator and its control velocities takes the nonholonomic platform constraint into account and may be expressed by the reduced direct instantaneous kinematic model, i.e.:

$$\dot{X} = \bar{J}(q)u \tag{13}$$

with: $\bar{J}(q) = J(q)M(q)$.

4 MANIPULABILITY MEASURES

A well-established tool used for the motion analysis of manipulators is known as the manipulability ellipsoid approach. The concept of *manipulability* was originally introduced by Yoshikawa ((Yoshikawa, 1985), (Yoshikawa, 1990)) for arm manipulators, in order to denote a measure for the ability of a manipulator to move in certain directions. The set of all end-effector velocities that may be attained by joint velocities such that the Euclidean norm of $\dot{q}_a$, $\|\dot{q}_a\| = (\dot{q}_{a1}^2 + \dot{q}_{a2}^2 + \cdots \dot{q}_{an}^2)^{1/2}$, satisfying $\|\dot{q}_a\| \le 1$ is an ellipsoid in m-dimensional Euclidean space. This ellipsoid represents the manipulation capability and is called the "manipulability ellipsoid".

Yoshikawa defines the manipulability measure w as follows:

$$w = \sqrt{\det(J_a(q_a)J_a^T(q_a))} \tag{14}$$

which can be simplified into $w = |\det(J_a(q_a))|$ when $J_a(q_a)$ is a square matrix.

Let's now consider the singular value decomposition of J_a, as given by:

$$J_a = U_a \Sigma_a V_a^T \tag{15}$$

where $U_a \in R^{m\times m}$ and $V_a \in R^{n\times n}$ are orthogonal matrices, and:

$$\Sigma_a = \begin{bmatrix} \sigma_{a1} & & & & 0 & \vdots & \\ & \sigma_{a2} & & & & \vdots & \\ & & . & & & \vdots & 0 \\ & & & . & & \vdots & \\ 0 & & & & \sigma_{am} & \vdots & \end{bmatrix} \in R^{m\times n} \tag{16}$$

in which: $\sigma_{a1} \ge \sigma_{a2} \ge \cdots \ge \sigma_{am}$.

The value of $w = \sigma_{a1}.\sigma_{a2}.\cdots\sigma_{am}$ is proportional to the ellipsoid volume.

Another measure has been proposed for characterizing the distance of a configuration from a singularity (Salisbury, 1982). This measure is given by:

$$w_2 = \frac{\sigma_{am}}{\sigma_{a1}} \tag{17}$$

where σ_{a_1} and σ_{a_m} are the maximum and minimum singular values of the Jacobian matrix, respectively.

Bayle (Bayle, 2001) defined a measure w_5 that extended the notion of eccentricity of the ellipse, i.e.:

$$w_5 = \sqrt{1 - \frac{\sigma_{am}^2}{\sigma_{a1}^2}}. \tag{18}$$

The structure of the manipulator arm consists of an arm portion with three joints and a wrist portion with three joints whose axes intersect at a single point. The arm portion concerns the positioning task, while the wrist portion focuses on gripper orientation. It proves quite useful to divide this study into wrist and arm singularities. We present herein the manipulability of the considered system for positioning tasks.

5 DIRECTIONAL MEASURE

All of the abovementioned measures describe system manipulability in general terms, without taking the task the manipulator is being asked to perform into account. One key factor behind the failure of these measures is the fact that they do not include information either on the task or on the direction the end-effector is required to move. A new measure should therefore be introduced to address this situation.

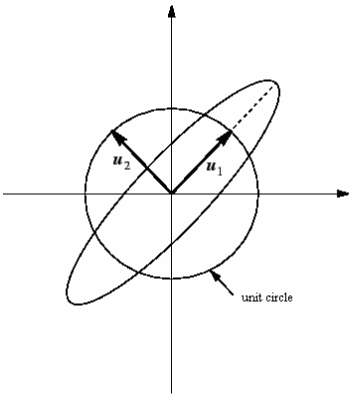


Figure 2: Manipulability ellipse in the two-dimensional case.

The Singular Value Decomposition (15) of the Jacobian matrix and its geometric relationship offer further insights into characterizing the manipulability of mechanical systems. Let u_{ai} be the i^{th} column vector of U_a. The primary axes of the manipulability ellipsoid are then: $\sigma_{a1}u_{a1}, \sigma_{a2}u_{a2}, \cdots \sigma_{am}u_{am}$. Figure 2 provides an illustration of the two-dimensional case, according to which u_1 and u_2 yield the major and minor ellipse axes, respectively. We propose to include information on the direction of the task wished to precisely know the manipulation capacity of the arm manipulator for the execution of this operational task.

Let $\dot{X}_d$ be the desired task. We now define a unit vector $d = \frac{\dot{X}_d}{\left\|\dot{X}_d\right\|}$, which gives the direction of the imposed task.

By using properties of the scalar product and the singular values that represent radius of the ellipsoid, we define a new manipulability measure as being the sum of the absolute values of the scalar products of the directional vector of the task by the singular vectors pondered by their corresponding singular values. This new measure is noted w_{dir}.

$$w_{dir} = \sum_{i=1}^{m} \left|(d^T.u_{ai})\sigma_{ai}\right| \qquad (19)$$

This measure is maximized when the arm capacity of manipulation according to the direction of the task imposed is maximal. It is equal to zero if there is no possibility of displacement according to this direction.

6 CONTROL SCHEME

Whitney (Whitney, 1969) first proposed using the pseudo-inverse of the manipulator Jacobian in order to determine the minimum norm solution for the joint rates of a serial chain manipulator capable of yielding a desired end-effector velocity. A weighted pseudo-inverse solution approach also allows incorporating the various capabilities of different joints, as discussed in Nakamura (Nakamura, 1991) and Yoshikawa (Yoshikawa, 1984). One variant of this approach includes superposition of the Jacobian null space component on the minimum norm solution to optimize a secondary objective function (Baerlocher, 2001).

This same notion can then be extended to the case of a nonholonomic mobile manipulator. The inverse of the system given by equation (12) exhibits the following form:

$$u = \bar{J}^+ \dot{X}_d + (I - \bar{J}^+ \bar{J})Z \qquad (20)$$

where Z is a (N-1)-dimensional arbitrary vector.

The solution to this system is composed of both a specific solution $\bar{J}^+ \dot{X}_d$ that minimizes control velocities norm and a homogeneous solution $(I - \bar{J}^+ \bar{J})Z$ belonging to the null space $N(\bar{J})$. By definition, these latter added components do not affect task satisfaction and may be used for other purposes. For this reason, the null space is sometimes called the redundant space in robotics.

The Z vector can be utilized to locally minimize a scalar criterion. Along the same lines, Bayle (Bayle,2001b) proposed the following scheme:

$$u = \bar{J}^+ \dot{X}_d - W(I - \bar{J}^+ \bar{J})M^T (\frac{\partial P}{\partial q})^T \qquad (21)$$

where $\dot{X}_d$ is the desired task vector, W a positive weighting matrix, and *P(q)* the objective function dependent upon manipulator arm configuration. To compare the advantage of our manipulability measure with those presented in the literature, the control scheme whose objective function depends on various measures (w, w_5 and w_d) is to be applied. For manipulation tasks involving a manipulator arm, it is helpful to consider manipulability functions whose minima correspond to optimal configurations, e.g. (-w), (-w_d) or w_5.

As for the calculus, we used the numerical gradient of $P(q)$.

7 RESULTS

7.1 Simulation Results

In this section, we will consider a Manus arm mounted on a nonholonomic mobile platform powered by two independent-drive wheels, as described in Section 3. The mobile platform is initially directed toward the positive X-axis at rest (q_p=[0, 0, 0]^T) and the initial configuration of the manipulator arm is: q_a= [4.71, 2.35, 4.19]^T (*rad*). The arm is fixed on the rear part of the platform. The coordinates of the arm base with respect to the platform frame are: [-0.12, -0.12, 0.4]^T (*m*). The imposed task consists of following a straight line along a Y-axis of the world frame {W}. The velocity along a path is constant and equal to 0.05 m.s^{-1}. Results obtained in the following cases have been reported:

- in optimizing arm manipulability measure w;
- in optimizing arm manipulability measure w_{dir}.

The comparison criteria are thus:

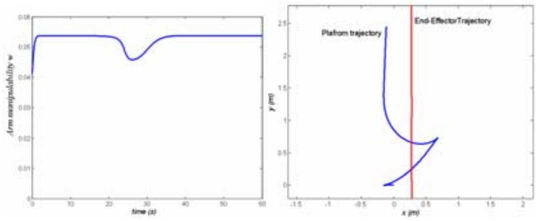
- Platform trajectory profile,
- indicator of energy spent E by the platform,
- manipulation capacity of the arm at the end of the task, measured by w.

w is the most widely used indicator of manipulability found in the literature. In our case, w serves as a reference to evaluate the efficiency of the control algorithm in terms of arm manipulability; its values range between 0, which corresponds to singular configurations, and 0.06, which corresponds to good manipulability. In addition, we are looking for forward displacements of the platform and smooth trajectories. End-effector trajectories enable checking if the task has been performed adequately.

E is defined by $E = \sum v_l^2 + v_r^2$, with v_l and v_r the linear velocities respectively of the left and right wheels of the platform.

Before presenting each case separately, it should be noted that, for each one of them, the task is carried out correctly.

Figure 3 shows simulation results in which arm manipulability w is used as the optimizing criterion for solving mobile arm redundancy. As depicted in Figure 3a, the arm manipulability w quickly improves up to a threshold corresponding to acceptable configurations. Around 25 seconds, local degradation of the manipulability measure is shown to be quite low.

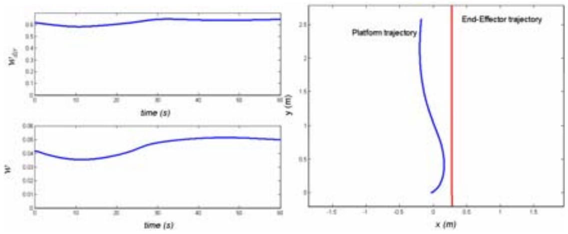


(a) (b)

Figure 3: Simulation result when optimizing arm manipulability measure w.

To quickly improve arm manipulability while performing the imposed task, the arm extends and the platform retracts (see Fig. 3b). The platform stops retracting once arm manipulability has been optimized; afterwards, it advances so that the unit carries out the imposed task. This evolution corresponds to the first graining of the platform trajectory. Since the platform is poorly oriented with respect to the task direction, its contribution is limited by the nonholonomic constraint, which does cause slight degradation to the manipulability, as shown in Figure 3a. The reorientation of the platform, which corresponds to the second point of graining, allows for improvement and optimization of the manipulability measure. The mobile arm achieves the desired task with an acceptable arm configuration from a manipulation perspective; the platform moves in reverse gear however, which counters our intended aim. As there are two graining points, the platform trajectory is not smooth. The energy indicator E for this trajectory is E=7.15 m^{2}s^{-2}.

In Figure 4, we have used the proposed directional manipulability of the arm to solve mobile arm redundancy. Figure 4a shows the evolution of the directional manipulability measure $w_{dir,}$ and arm manipulability w for comparison. The directional manipulability of the arm is initially good; it decreases slightly then improves progressively. Corresponding measure w does not reach its maximum value, but remains in a beach of acceptable values, far from the singular configurations. In this case, no local degradation of the manipulability measure is detected. Figure 4b presents the trajectory of the middle axis point on the platform. This figure indicates that the mobile platform retracts during a short period of time at the very beginning in order to improve arm manipulability. The platform reorients itself according to the imposed task without changing its motion direction. In executing a desired task, the platform thus follows a smoother trajectory and displays forward displacements. The energy indicator E for this trajectory is E=3.1 $m^2.s^{-2}$. Energy expenditure is lower than the preceding case.



(a) (b)

Figure 4: Simulation result when optimizing arm directional manipulability measure $w_{dir.}$

7.2 Results on Real System

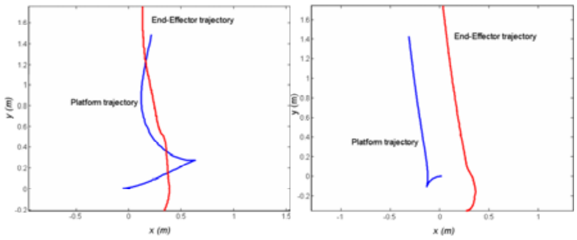
To illustrate the results presented in theory, we implemented on the real robot the algorithm. Starting from a given configuration q_i, collected by the sensors data, we impose an operational task on the end effector of the arm manipulator which consists in following a straight line according to the direction perpendicular to the axis longitudinal of the platform (Y axis of the world frame). Imposed velocity is 0.005 m per 60 ms cycle.

It should be noted that for each case, the task is carried out correctly with good configuration from manipulability point of view.

Figure 5 presents the platform and end effector trajectories respectively in the cases of optimizing arm manipulability *w* (figure 5a) and arm directional manipulability w_{dir} (figure 5b).

Platform and OT trajectories presented on Figure 5a show that the platform carries out most part of its movement in reverse gear. The end effector follows a straight line with an error which reaches 21 cm in end of the task. This error includes a set of measurement errors and the tracking error.

In the case of directional manipulability optimization, figure 5b indicates that the platform moves back a little at the beginning and moves according to the direction of the task. End-effector follows a straight line with a weak error at the beginning (less than 3cm) better than the case of the optimization of w. The tracking error increases at the end of the execution of the task. Indeed, as the calculation of the gripper position is done on the basis of odometric data, which generates not limited errors, the tracking error increases.



(a) (b)

Figure 5: End-Effector and platform trajectories in the real case.

7.3 Discussion

For all the cases studied, the task has been performed adequately. The end-effector follows the desired trajectory, as represented by a straight line on the above figures. When a criterion is optimized, manipulability is maintained up to a certain level. Nevertheless, in the case of w optimization, local deteriorations are observed; these correspond to graining points in the platform trajectories. In the case of w_d optimization, local degradation does not appear. Moreover, since w_d takes task direction into account, the platform advances normally. The arm is more heavily constrained by w_d, which adds a supplementary condition on the task direction. The platform seeks to replace the arm more quickly in those configurations better adapted to following the direction imposed by the task in progress.

This more natural behavior offers the advantage of not disorienting the individual, an important feature in assistive robotics, which calls for the robot to work in close cooperation with the disabled host.

8 CONCLUSION

The purpose of this paper is to take advantage of system redundancy in order to maximize arm manipulability. Arm manipulability measures serving as performance criteria within a real-time control scheme make task execution possible with the best arm configuration from manipulation ability point of view. However, platform trajectories contain graining points, especially when the system is poorly-oriented with respect to the operational task. The platform moves in reverse gear for the most part of task execution. We have proposed a new measure that associates information on task direction with a manipulation capability measure. As shown in the paper, thanks to the new criterion, the imposed task is performed with human-like smooth movements and good manipulation ability. Both characteristics play an important part for implementing efficient man machine cooperation.

The number of platform trajectory graining points is reduced and, for the most part, the platform moves forward.

Work in progress is focusing on the inclusion of obstacle avoidance in the control scheme in order to improve coordination between the two subsystems. Another work relates to the development of a control strategy for seizure. This strategy takes into account both of human-machine cooperation and the presence of obstacles in the environment.

REFERENCES

P. Baerlocher, "Inverse kinematics techniques for the interactive posture control of articulated figures," PhD thesis, Lausanne, EPFL 2001.

B. Bayle, J.Y. Fourquet, M. Renaud, "Manipulability Analysis for Mobile Manipulators," In ICRA'2001, Seoul, South Korea, pp. 1251-1256, May 2001.

B. Bayle, J. Y. Fourquet, M. Renaud. Using Manipulability with Nonholonomic Mobile Manipulators, 3rd International Conference on Field and Service Robotics (FSR'2001), Helsinki (Finlande), 11-13 Juin 2001, pp.343-348

M. Busnel, R. Gelin and B. Lesigne, "Evaluation of a robotized MASTER/RAID workstation at home: Protocol and first results", Proc. ICORR 2001, Vol. 9, pp. 299-305, 2001.

G. Campion, G. Bastin and B. D'Andréa-Novel, "Structural proprieties and classification of kinematic and dynamic models of wheeled mobile robots," IEEE Transaction on Robotics and Automation, Vol. 12, no. 1, pp. 47-62, February 1996.

P. Dario, E. Guglielmelli, B. Allotta, "MOVAID: a personal robot in everyday life of disabled and elderly people," Technology and Disability Journal, no 10, IOS Press, 1999, pp. 77-93.

H. G. Evers, E. Beugels and G. Peters, "MANUS toward new decade", Proc. ICORR 2001, Vol. 9, pp. 155-161, 2001.

P. Hoppenot, E. Colle, "Mobile robot command by man-machine co- operation - Application to disabled and elderly people assistance," Journal of Intelligent and Robotic Systems, Vol. 34, no 3, pp 235-252, July 2002.

S. Kang, K. Komoriya, K. Yokoi, T. Koutoku, K. Tanie, "Utilization of Inertial Effect in Damping-based Posture Control of Mobile Manipulator," International Conference on Robotic and Automation, pp. 1277-1282, Seoul, South Korea, May 2001.

O. Khatib, "Inertial properties in robotic manipulation: an object-level framework," Int J. Robotics Research, Vol. 13, No. 1, pp. 19-36, 1995.

H. Kwee, C.A. Stanger, "The Manus robot arm", Rehabilitation Robotics News letter, Vol. 5, No 2, 1993.

K. Nagatani, T. Hirayama, A. Gofuku, Y. Tanaka, "Motion planning for Mobile Manipulator with keeping Manipulability," IROS 2002, , pp 1663-1668, Lausane, Switzerland, October 2002.

Y. Nakamura, "Advanced robotics, redundancy and optimization," Addison Wesley Publishing, 1991.

J. K. Salisbury and J.J. Craig, "Articulated hands: Force Control and Kinematic Issues," Intl. J. Robotics Research, Vol. 1, no. 1, pp. 4-17, 1982.

L. Sciavicco and B. Siciliano, "Modeling and control of robot manipulators," The McGraw-Hill companies, Inc., 1996.

H. Seraji, "An on-line approach to coordinated mobility and manipulation," In ICRA'1993, Atlanta, USA, May 1993, pp. 28-35.

D.E Whitney, "Resolved Motion Rate Control of Manipulators and Human Prosthetics," IEEE Trans on Man Machine Systems, Vol 10, pp. 47-53, 1969.

Y. Yamamoto and X. Yun, "Coordinating locomotion and manipulation of mobile manipulator," Recent Trends in Mobile Robots Zheng Y. Ed., 1987.

Y. Yamamoto and X. Yun, "Unified analysis an mobility and manipulability of mobile manipulators," IEEE Int. Conf. Robotics and Automation, Detroit, pp. 1200-1206, USA, 1999.

T. Yoshikawa, "Manipulability of Robotic Mechanisms," International Journal of Robotics Research, 1985, Vol. 4, no. 2, pp. 3-9.

Yoshikawa, T.: Foundation of robotics: Analysis and control, The MIT Press, 1990.

T. Yoshikawa, "Analysis and control of Robot manipulators with redundancy," In M. Brady & R. Paul, editors, Robotics Research: The First International Symposium, MIT Press, pp. 735-747, 1984.

H.F.M. Van der Loos, "VA/Stanford Rehabilitation Robotics Research and Development Program: Lessons Learned in the Application of Robotics Technology to the Field of Rehabilitation," IEEE Trans. Pn Rehabilitation Engineering, Vol. 3, n°1, pp. 46-55, March 1995.

HYBRID MOTION CUEING ALGORITHMS FOR REDUNDANT ADVANCED DRIVING SIMULATORS

Hatem Elloumi, Nadia Maïzi and Marc Bordier
Centre de Mathématiques Appliquées, École des Mines de Paris
2004 route des Lucioles, 06902 Sophia Antipolis Cedex, France
Email: {hatem.elloumi,nadia.maizi,marc.bordier}@ensmp.fr

Keywords: Driving simulation, motion cueing, hybrid control.

Abstract: Redundant Advanced Driving Simulators (hexapods mounted on rails) present an extra capability to reproduce motion sensations. The exploitation of this capability is currently done by frequency separation methods without taking into account the frequency overlapping between the hexapod and the rails. Within this bandwidth, these two degrees of freedom could be considered as equivalent. Our aim is to use this equivalence to improve the motion restitution. We offer two algorithms based on the hybrid systems framework which deal with the longitudinal mode. Their goal is to improve the restitution of motion sensations by reducing false cues (generated by actuators braking) and decreasing null cues (due to actuators blocking). Our algorithms include and treat all steps of motion cueing: motion tracking (restitution), braking before reaching the displacement limits, washout motion, and switching rules.

1 INTRODUCTION

Driving simulators are advanced devices composed of four components: a virtual scene projected on a wide screen to imitate the road and the traffic, an audio system to play the driving sounds (horn, squeal of brakes, etc.), a car cockpit (including a real dashboard, the pedals and the seat of the driver) to copy the body position and the interaction of the driver with a real vehicle and finally a robot carrying the car cockpit to provide its motion. While the first three components could be considered as offering a sufficiently high degree of realism, the robot presents a very low capacity of displacement, thus preventing it from performing the real car motions.

In fact, the aim of a driving simulator is not tracking real trajectories produced by outdoors driving but reproducing the corresponding motion *sensations*. How could we then, generate realistic motion *sensations* in simulation despite the constrained robot motion? It is the aim of Motion Cueing Algorithms (MCA) to give heuristically an answer to this problem. This paper uses the hybrid systems framework (Zaytoon, 2001; Van der Shaft and Schumacher, 1999) to build two MCA designed specifically for redundant simulation robots. These robots are made up of two parts:

- A Gough-Stewart parallel robot or hexapod (this parallel robot is composed of three parts: a moving body called the *platfom* (carrying the car cockpit) linked to the *base* through six extensible legs. Each leg is composed of a prismatic joint (i.e. an electro hydraulic jack) and two passive spherical joints making the connection with the base and the platform. For an excellent overview of parallel robots the reader is referred (Merlet, 2000)). The jacks' excursions are of ± 20cm allowing a six-dimension motion of the car cockpit up to ± 15cm (in linear directions) and up to 30deg (for rotations).
- A rail system carrying the base of the hexapod to provide an extra motion in the horizontal plane. In this paper the rails' limits are: ± 2m.

(Elloumi, 2006) shows that rails and the hexapod present overlapping bandwidths in the high frequency domain. So how could we benefit from this redundancy? Our approach is based on the *classical* MCA which will be presented in section 3. But before addressing this point, section 2 will deal with the notion

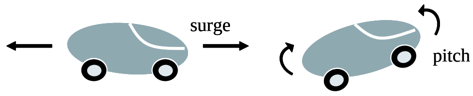


Figure 1: The longitudinal mode.

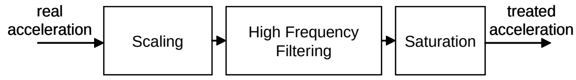


Figure 2: Preliminary treatments for the classical MCA.

of sensation to close the description of the simulator objective.

Remark: The results presented here are done for the longitudinal mode. However, they could be extended for the other space directions. The longitudinal mode is composed of the surge and pitch motions as depicted in figure 1.

2 MOTION PERCEPTION

Even in the absence of visual information (closed-eye subject), humans detect motion thanks to their inertial receptor: the *vestibular system*. Located in the inner ear, this biological apparatus measures both linear and angular motion of the head (a thorough description is given in (Elloumi, 2006; Telban et al., 2000; Angelaki and Dickman, 2004)) if they are beyond detection thresholds (on acceleration and speed respectively).

The *motion sensation* is built at the level of the brain not only from the vestibular system information but also from all the perception receptors (most particularly: the eyes) cues. In this paper, as commonly done in driving simulation, we consider that apart from the vestibular system, all the other sensors receive coherent and well adapted cues. As a consequence, in this paper the motion sensation will be considered as the interpretation of head displacements by the inertial receptor.

One remarkable gain of working with motion sensations instead of real trajectories (accelerations) is illustrated by the tilt coordination. In driving (or flight) simulation, a simultaneous rotation of the driver's head and the visual scene at a very slow rate happens to create an illusion of linear acceleration: "*When a visual scene representing an accelerated translation is presented to the driver while the simulation cockpit is tilted at an undetectable rate, the relative variation of the gravity vector will be partly interpreted as an actual linear acceleration*" (Reymond et al., 2002). Thus from a control point of view, the tilt coordination leads to a low-frequency motion sensation through a very small variation of the jacks' displacement as we shall see in the next section.

3 CLASSICAL MOTION CUEING ALGORITHM

This scheme was developed in 1970 by (Parrish et al., 1975). Despite its simplicity, this algorithm displays the importance of tilt coordination to restitute longitudinal accelerations. This scheme is based on the simple observation that the simulator translation is very limited so that only fast (*onset*) accelerations could be tracked. Consequently, the principle of this method is to use filtering to extract from the real car acceleration the high frequency component and address it to the robot translation. Hopefully, the tilt coordination enables the reproduction of slow (*sustained*) accelerations. Filtering (low frequencies) is performed to supply the tilt rotation as well. As for the restitution of the rotation speed, high pass filtering is performed to deal with angular limits.

The classical MCA is a linear approach which is commonly preceded by some preliminary treatments of the real accelerations to cope with robot motion limits (see figure 2).

4 THE REDUNDANCY PROBLEM

Restituting longitudinal acceleration on redundant simulators could be done thanks to three degrees of freedom (dof) as depicted in Fig.3: the base translation (X) (performed by the rails), the hexapod translation (x) and the tilt coordination (θ: tilt angle) (both performed by the jacks). As shown in (Elloumi, 2006), the behavior of the last dof is independent from the first two as the rotation due to the tilting is limited by a very low detection threshold.

As a consequence, in order to improve the quality of motion cueing only the translations behaviors should be considered. The considered linear acceleration[1] provided to the driver by the simulation robot is then: $\ddot{X} + \ddot{x}$.

Besides as the rails and jacks bandwidths are overlapping in the high frequencies domain, these two dof could be considered as equivalent. How could we

[1]The tilt coordination contribution $g\theta$ (where g is the gravity magnitude) is omitted from the hybrid algorithms that we shall present (but could be added outside these algorithms).

then exploit this equivalence? We offer two algorithms based on the hybrid systems framework which use only one translation at a time and a switching strategy to cope with the limits.

Figure 3 shows these two translations: hexapod translation (x) and base translation (X) each constrained by three levels of limitation: position, speed and acceleration $\pm\xi_L, \pm\dot{\xi}_L, \pm\ddot{\xi}_L$ ($\xi \in \{x, X\}$). The models ruling the variation of $\xi \in \{x, X\}$ are linearized models (double integrators):

$$\ddot{\xi} = u_\xi, \ \ \xi \in \{x, X\} \tag{1}$$

where u_ξ is the reference acceleration (control). As these dof are limited, we have to define two strategies: a braking strategy (triggered when nearing the limits) and a washout strategy (going back to a neutral position) strategy (once braking has been done).

4.1 Braking Strategy

In this paper we adopted the parabolic braking (constant braking acceleration $\ddot{\xi}_b$) in order to stop the translation at its limits $\pm\xi_L$ (null speed $\dot{\xi} = 0$). The triggering condition is then:

$$\dot{\xi}^2 - 2\ddot{\xi}_b(\xi_L - |\xi|) \geq 0, \ \ \xi \in \{x, X\} \tag{2}$$

The braking acceleration $\ddot{\xi}_b$ determines the free zone (braking-free zone) size. The higher $\ddot{\xi}_b$ is the bigger is the free zone. Nevertheless, the incoherent sensations would be strong in this case. At the opposite, choosing this acceleration to be as low as the detection threshold (0.05ms^{-2}) would considerably reduce the braking sensations and would noticeably reduce the free zone at the same time.

We have studied the influence of this parameter on the ratio between the free zone volume and the theoretically available one. As the phase profile (speed and position) is independent from the acceleration value, this ratio is equal to the ratio between the surfaces of the phase profiles. The theoretical surface is $S_{theo} = 4\xi_L\dot{\xi}_L$ and the free zone surface is:

$$S_{free} = \begin{cases} \frac{16}{3}\xi_L^{\frac{3}{2}}\sqrt{\ddot{\xi}_b} \text{ if } 0 \leq \ddot{\xi}_b < \frac{1}{4}\frac{\dot{\xi}_L^2}{\xi_L} \\ 4\left[\xi_L - \frac{1}{4}\dot{\xi}_L^2\ddot{\xi}_b^{-1}\right]\dot{\xi}_L + \frac{2}{3}\dot{\xi}_L^3\ddot{\xi}_b^{-1} \text{ otherwise} \end{cases} \tag{3}$$

The ratio S_{free}/S_{theo} is saturated starting from a certain braking acceleration $\ddot{\xi}_b$. In other words, starting from this point, the magnitude of $\ddot{\xi}_b$ wouldn't have a significant impact on the free zone size. However, the braking duration $\dot{\xi}_0\ddot{\xi}_b^{-1}$ (bounded by $\dot{\xi}_L\ddot{\xi}_b^{-1}$) will keep decreasing.

4.2 Washout Strategies

The goal of the washout is to bring the translation to its neutral position $(\xi, \dot{\xi}) = (0, 0)$. We present two washout strategies:

4.2.1 Known Starting Point

In this case the backward motion starts at $(\xi, \dot{\xi}) = (\pm\xi_L, 0)$, the chosen washout control is:

$$u_\xi = -\text{sign}(\xi)\begin{cases} a_r \text{ if } \frac{\xi_L}{2} \leq |\xi| \leq \xi_L \\ -a_r \text{ if } 0 \ \leq |\xi| \leq \frac{\xi_L}{2} \end{cases} \tag{4}$$

Taking $a_r = a_{threshold} = 0.05\text{ms}^{-2}$ would make this motion imperceptible. Finally, the duration of this strategy is: $2\sqrt{\xi_L a_r^{-1}}$.

4.2.2 Unknown Starting Point

In this case the backward motion starts at an unknown point (within the limits). The control is then a Proportional Derivative (PD):

$$\ddot{\xi} = -\mu\dot{\xi} - k\xi \tag{5}$$

The parameters (μ, k) have to be chosen so that the motion limits are respected.

These definitions of braking and washout techniques enable us to present our hybrid algorithms.

5 SYMMETRIC ALGORITHM

The principle of the symmetric algorithm is to use only one translation at a time. When the active translation reaches its limits, switching will be performed to activate the idle dof. In other words, both translations reproduce the reference acceleration as *relay runners*.

In non redundant simulators (without rails), when the hexapod translation is close to its limits, braking and washout will be successively triggered. The operator has to wait until these two operations finish in order to get back coherent motion cues. The symmetric algorithm will speed up the reactivation of the acceleration restitution by using the idle translation during the washout (impercebtible) motion of the active one. This algorithm is called *symmetric* because both translations have the same role in the motion cueing process.

In order to represent the symmetric algorithm as a hybrid automaton, two points have to be defined: the working modes and the rules of correct operation.

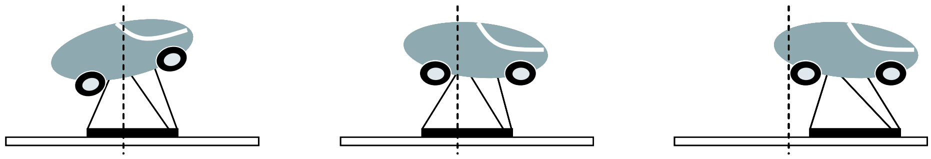


Figure 3: The redundancy of the longitudinal acceleration restitution.

Working Modes

In the case of the symmetric algorithm, both translations (X and x) have *the same* working modes:

1. *active*: the dof ξ tracks the *treated* reference acceleration
2. *brake*: parabolic braking
3. *washout*: known starting point
4. *idle*: null acceleration

Rules of Correct Operation

By defining these rules we characterize the way the hybrid automaton works. In our case, these rules are valid for *both* translations:

1. braking must lead to the limit position with a null speed
2. braking must be followed by a washout motion
3. washout must lead to the neutral position with a null speed
4. reactivating one dof could be done only starting from the neutral position (with a null speed)
5. if one dof is braking, the other one mustn't be active. The braking sensations could deteriorate indeed the quality of the free dof restitution

5.1 The Symmetric Automaton

Figure 4 shows the symmetric automaton (we observe a central symmetry around the state (**Hexapod:** washout, **Rail:** washout)). Three types of transition predicates appear:

- `hexa=0` (or `rail=0`) i.e. the dof has attained the neutral position $(\xi, \dot{\xi}) = (0, 0)$
- `decl_hexa` (or `decl_rail`) i.e. the braking condition is fulfilled (see (2))
- `lim_hexa` (or `lim_rail`) i.e. that an extreme position $(\xi, \dot{\xi}) = (\pm\xi_L, 0)$ has been attained

If we consider that braking is instantaneous then this automaton could be reduced to the four states outside the dashed box i.e. an alternation between the activation and the washout for both translations. The states inside the box take into account the parabolic braking and the subsequent activation of the washout.

5.2 Simulations

Matlab/Simulink and Stateflow were used to perform simulations. The initial state was chosen to be (**Rail:** active, **Hexapod:** idle). The reference acceleration profile was extracted from the Renault simulations in (Dagdelen, 2005). The parameters values used in these simulations are : $\ddot{X}_b = 1ms^{-2}$, $\ddot{x}_b = 0.2ms^{-2}$ and $a_r = 0.05ms^{-2}$ for both dof. Transition times are indicated by vertical lines. In figure 5, we can distinguish 6 working phases:

- (**Rail:** active, **Hexapod:** idle)
- (**Rail:** brake, **Hexapod:** idle)
- (**Rail:** washout, **Hexapod:** active)
- (**Rail:** washout, **Hexapod:** brake)
- (**Rail:** washout, **Hexapod:** washout)
- (**Rail:** active, **Hexapod:** washout)

6 MASTER-SLAVE ALGORITHM

In this algorithm the roles played by the two translations are *asymmetric*. One dof is the *master* i.e. responsible for restituting the motion. The other dof is the *slave* which has to counterbalance the "bad" master behaviors. It consists in producing opposite accelerations to the master's when the latter brakes or goes back to the neutral position (washout).

Working Modes

The working modes are different for each translation. The master's modes are:

1. *active*: the master dof tracks the treated reference acceleration

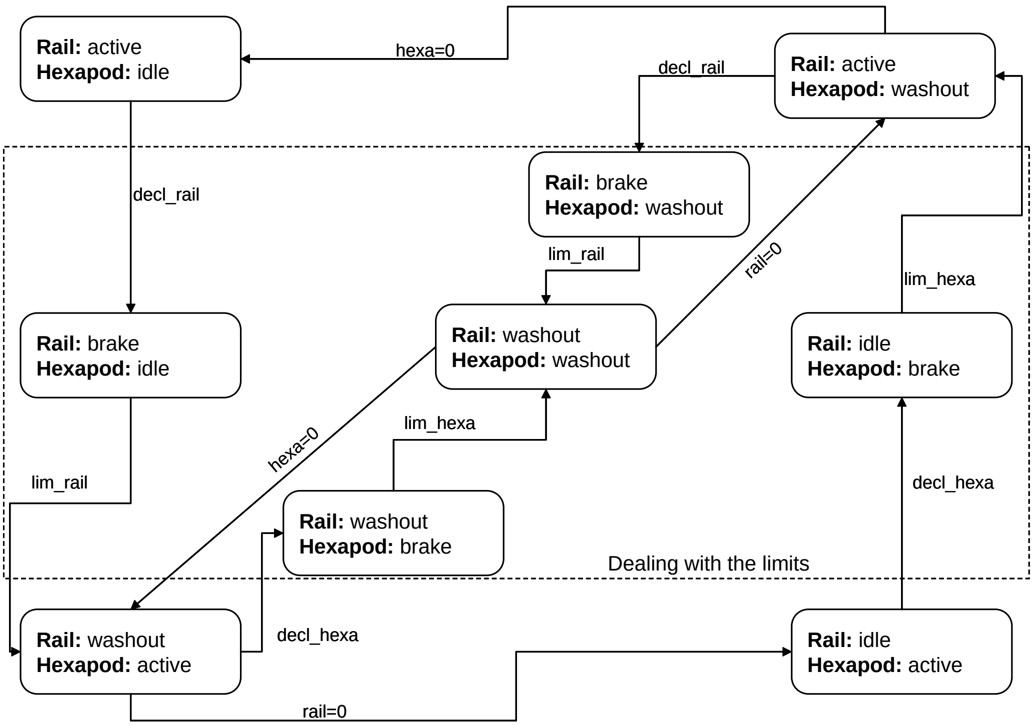


Figure 4: The symmetric automaton.

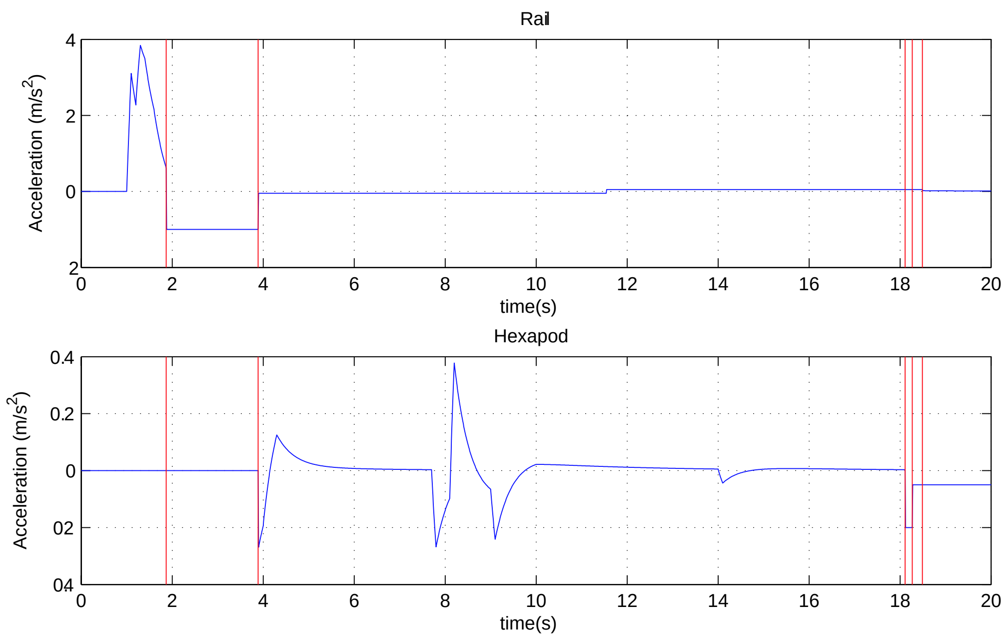


Figure 5: Simulation of the symmetric algorithm.

2. *brake*: parabolic braking
3. *quick-washout*: known starting point
4. *washout*: unknown starting point and undetectable
5. *idle*: null acceleration

The slave's modes are:

1. *counter-brake*: the slave dof tracks the acceleration opposite to the master's parabolic braking one
2. *counter-washout*: the slave dof tracks the acceleration opposite to the master's quick washout one
3. *brake*: parabolic braking
4. *washout*: unknown starting point and undetectable
5. *idle*: null acceleration

Rules of Correct Operation

1. braking of both translation (master and slave) must lead to the limit position with a null speed
2. braking must be followed by a washout motion. The master's washout is quick only if the slave is within its free zone
3. the master could be (re)activated only starting from its neutral position (the slave mustn't be in the braking mode)
4. after the counter-washout mode, the slave starts a washout motion to its neutral position

6.1 The Master-slave Automaton

Figure 6 shows the master-slave automaton. The transitions have the same signification as for the symmetric automaton. Similarly, if we consider that braking is instantaneous then this automaton could be reduced to these subsequent states:

- Initial state: (**Master:** active, **Slave:** idle)
- (**Master:** brake, **Slave:** counter-brake): when nearing the limits, the master brakes. The slave provides the opposite acceleration so that the total acceleration (perceived by the driver) is null.
- (**Master:** quick-washout, **Slave:** counter-washout): after the master's braking, a quick motion brings it to its neutral position. The slave counterbalances this motion so that the overall acceleration is null again.
- (**Master:** active, **Slave:** washout): the master is reactivated once reaching its neutral position. The slave starts the washout in order to improve its future capacity of compensation. From this state two permutations could occur: going back to the initial state if the slave washout has finished or switching to (**Master:** brake, **Slave:** counter-brake) if the master reaches once again its limits.

The dashed box integrates all the states that describe the automaton behavior when the slave couldn't perform its counterbalancing role. It happens when the slave reaches its limits and has to break. In this case, the hybrid automaton starts a backward motion of both dof that ends by going back to the initial state.

6.2 Simulations

The reference acceleration profile is the same as before (scaled at 50% for a better visualisation). Rails were chosen to be the slave whereas the hexapod translation is chosen to be the master. In fact, as the rails motion capacity is higher than the hexapod one, the former is better suited to play the compensation (slave) role.

The algorithm parameters are: braking (and counter-braking) acceleration $0.3ms^{-2}$, quick washout (and counter-washout) acceleration $0.5ms^{-2}$ and slave braking acceleration $0.6ms^{-2}$. As for the slave washout, μ and k were chosen to be τ^{-1} and τ^{-2} where $\tau = 1.45s$. Figure 7 shows the simulation results. We can distinguish 5 phases:

- (**Hexapod:** active, **Rail:** idle)
- (**Hexapod:** brake, **Rail:** counter-brake)
- (**Hexapod:** quick washout, **Rail:** counter-washout)
- (**Hexapod:** active, **Rail:** washout)
- (**Hexapod:** active, **Rail:** idle)

7 CONCLUSION

In this paper we have presented two motion cueing algorithms based on the hybrid systems framework. These two algorithms exploit the redundancy of the simulators to maintain the reproduction of motion sensations despite the robot displacement limitations.

The symmetric algorithm presents a reliable initial restitution. However it generates incoherent sensations due to significant braking magnitudes. The Master/Slave algorithm has a lesser restitution capacity but it reduces considerably bad sensations by providing a full compensation (null sensations at the level of the driver) of braking and washout motions.

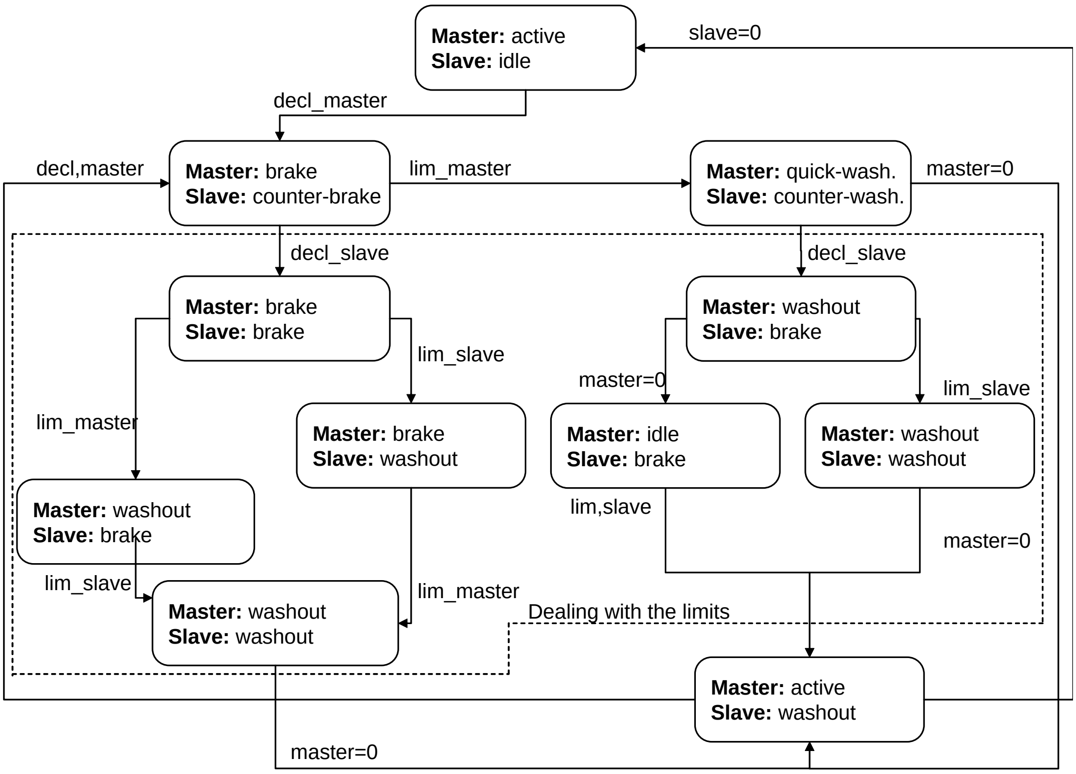


Figure 6: The master-slave automaton.

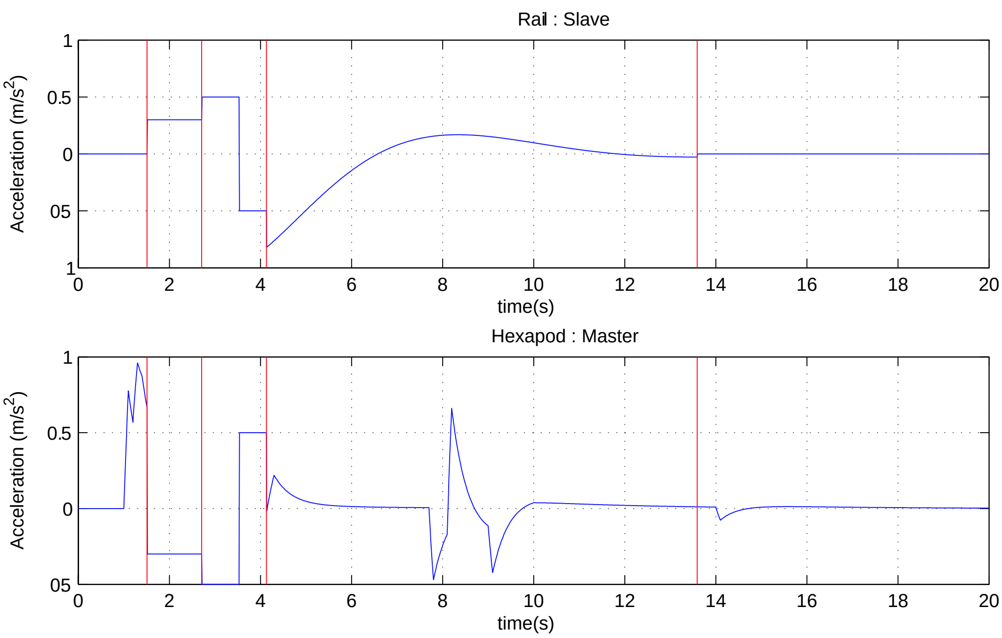


Figure 7: The master-slave simulations.

REFERENCES

Angelaki, D. and Dickman, J. (2004). Gravity or translation: Central processing of vestibular signals to detect motion. *Journal of Vestibular Research*.

Dagdelen, M. (2005). *Restitution des stimuli inertiels en simulation de conduite*. PhD thesis, Ecole des Mines de Paris - Renault.

Elloumi, H. (2006). *Commande des plate-formes avances de simulation de conduite*. PhD thesis, Ecole des Mines de Paris, Centre de Mathématiques Appliquées.

Merlet, J. (2000). *Parallel Robots*. Kluwer Academic Publishers.

Parrish, R., Dieudonne, J., and Martin Jr., D. (1975). Coordinated adaptive washout for motion simulators. *Journal of Aircraft*.

Reymond, G., Heidet, A., Canry, M., and Kemeny, A. (2002). Validation of Renault's dynamic simulator for adaptive cruise control experiments. *Driving Simulation Conference*.

Telban, R., Cardullo, F., and Guo, L. (2000). Investigation of mathematical models of otolith organs for human centered motion cueing algorithms. In *AIAA Modeling and Simulation Technologies Conference*, Denver CO.

Van der Shaft, A. and Schumacher, H. (1999). *An introduction to hybrid dynamical systems*. Springer-Verlag, Lecture Notes in Control, 251.

Zaytoon, J. (2001). *Systèmes dynamiques hybrides*. Hermes Sciences Publications.

IRREVERSIBILITY MODELING APPLIED TO THE CONTROL OF COMPLEX ROBOTIC DRIVE CHAINS

Omar Al Assad, Emmanuel Godoy
Ecole Supérieure d'Electricité (Supélec) Gif-sur-Yvette F-91192 Cedex, France
(Omar.AlAssad@ge.com, Emmanuel.Godoy@supelec.fr)

Vincent Croulard
General Electric Healthcare, 283 rue de la Minière, Buc F-78530 Cedex, France
(Vincent.Croulard@med.ge.com)

Keywords: Robotic, Modeling, Drive chain, Mechanical irreversibility, Efficiency, Gearbox.

Abstract: The phenomena of static and dry friction may lead to difficult problems during low speed motion (e.g. stick slip phenomenon). However, they can be used to obtain irreversible mechanical transmissions. The latter tend to be very hard to model theoretically. In this paper, we propose a pragmatic approach to model irreversibility in robotic drive chains. The proposed methodology consists of using a state machine to describe the functional state of the transmission. After that, for each state we define the efficiency coefficient of the drive chain. This technique gives conclusive results during experimental validation and allows reproducing a reliable robot simulator. This simulator is set up for the purpose of position control of a medical positioning robot.

1 INTRODUCTION

Modern control theories in robotics are more and more turned towards model-based controllers such as computed torque controllers, adaptive controllers or feedforward dynamics compensators. Therefore, dynamic modeling has become an inevitable step during controllers design. Besides, accurate dynamic modeling is a key point during simulations and the mechanism design process.

In the literature, the problem of robot dynamic modeling is treated in two steps. The first one concerns the mechanical behavior of the robot external structure considered often as a rigid structure. Many researchers have treated this problem and different techniques have been introduced to solve this issue. The two best-known methods in this matter are the Newton-Euler formulation and the Lagrange formulation (Khalil, 1999). The second step concerns the drive chain modeling which includes motors, gears and power loss modeling. Despite the advances made in the field of mechanical modeling, some issues are still without a convenient solution. We can mention, for instance, the phenomenon of irreversibility that characterizes certain types of mechanical transmissions such as worm gears (Henriot, 1991). This characteristic is often required for security reasons like locking the joint in case of motor failure or unexpected current cut-off. The purpose of this paper is to present a new modeling approach based on a state machine in order to simulate irreversible transmissions.

This paper is organized as follows. In section 2, we give a brief overview of the LCA vascular robot, which is used as an application for this study. Section 3 presents details about the modeling approach used for the robot structure and drive chain. Section 4 presents the irreversibility modeling issue and the proposed solution. Section 5 illustrates the experimental validation results. Finally, section 7 presents some concluding remarks.

2 LCA ROBOT PRESENTATION

The LCA vascular robot (figure 1) is used for medical X-ray imaging. It is a four-degrees-of-freedom open-chain robot composed of the following links: the L-arm (rotational joint), the Pivot (rotational joint), the C-arc (it has a translation movement in a circular trajectory template. Hence, it can be considered as a rotational joint around the virtual axis crossing the C-arc center) and the Lift (prismatic joint)

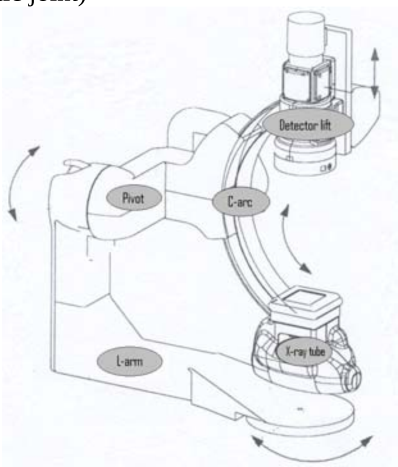
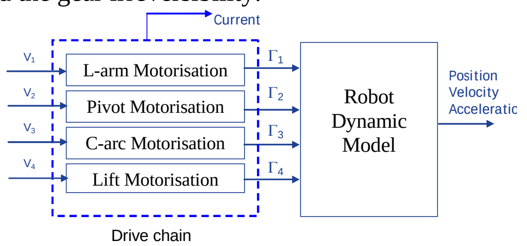


Figure 1: LCA robot.

3 MODELING APPROACH

The modeling of the LCA robot requires a clear distinction between the dynamic model of the mechanical structure and the drive chain model (figure 2). In fact, the dynamic model describes merely the relation between the applied torques and the ideal mechanical reaction of the gantry given by the joints acceleration.

The drive chain model takes into account the hard nonlinearities of the system such as the joint friction and the gear irreversibility.



V_i are the motors command voltage. Γ_i are the axes driving torques.

Figure 2: The robot model structure.

3.1 Dynamic Modeling

Two main methods can be used to calculate the dynamic model of the robot mechanical structure. We can mention the Newton-Euler formulation and the Lagrange formulation (Khalil, 1999).

Most authors use the Lagrange formulation that gives the mathematical expression of the model as:

$$\Gamma = A(q)\ddot{q} + C(q,\dot{q})\dot{q} + Q(q) \qquad (1)$$

Where $q, \dot{q}, \ddot{q}$ are respectively the vectors of joints position, velocity, and acceleration.

$A(q)$: the 4x4 robot inertia matrix.

$C(q,\dot{q}) \cdot \dot{q}$: the 4x1 Coriolis and centrifugal torque/ forces vector.

$Q(q)$: the 4x1 gravitational torques/ forces vector.

Γ : the 4x1 input torques/ forces vector.

To simulate the robot movement, we should use the inverse of the dynamic model as follow:

$$\ddot{q} = f(\Gamma, \dot{q}, q)$$

This model can be obtained directly using the recursive Newton-Euler equations, or it can be inferred from equation (1):

$$\ddot{q} = A(q)^{-1}(\Gamma - C(q,\dot{q})\dot{q} - Q(q)) \qquad (2)$$

The “A” matrix is inverted symbolically; this will result in a heavy mathematical expression, costly in term of computation time. Alternatively, this inverse can be calculated after numerical calculation of $A(q)$ which leads to faster simulations.

3.2 Drive Chain Modeling

The next step consists of modeling the drive chain, which includes the electrical motor (DC motor for this application), the mechanical transmission (gears) and the elements of power dissipation (friction) (figure 3).

We will describe briefly the first and the second elements and emphasize the third element, which is the purpose of this paper.

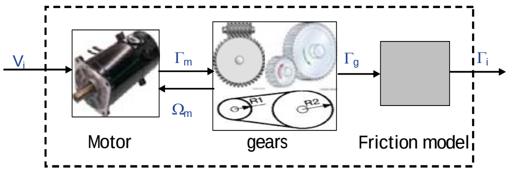


Figure 3: Drive chain model.

Actually, the phenomenon of irreversibility, obtained using specific transmissions and particular geometric dimensioning, is a complex problem and leads instinctively to non linear models. It can be treated using several approaches. In a microscopic point of view, the contacts among driving and driven elements are modeled as well as the applied forces. However, this rigorous approach leads to very complex analytical models, with serious difficulties in the implementation and simulations phases, particularly in the case of closed loop structures including controllers (Henriot, 1991). Besides, the identification of this type of models is very complicated due to the significant number of parameters. In a macroscopic point of view, the power transfer between the motor and the load is modeled with an efficiency coefficient taking into account the power transfer direction (load driving/driven) (Abba, 1999), (Abba, 2003). However, a proportional coefficient is insufficient to represent the irreversibility behavior. In our approach, we suggest the use of a state machine to define the current functional state of the transmission in order to reproduce the irreversibility.

3.2.1 DC Motor Modeling

The DC motor is a well-known electromechanical device. Its model has two inputs, the armature voltage and the shaft velocity. The output is the mechanical torque applied on the shaft. The DC motor behavior is modeled using two equations (Pinard, 2004) the electrical equation of the armature current (3) and the mechanical equation of the motor torque (4):

$$V - E = R \cdot I + L \cdot \frac{dI}{dt} \qquad (3)$$

where V is the motor voltage. I is the armature current. $E = K_{emf} \cdot \dot{q}_m$ is the electromotive force. $\dot{q}_m$ is the motor velocity; and the motor parameters are: K_{emf} (the back electromotive constant), R (the motor resistance) and L (the motor inductance).

$$\Gamma_m = K_t \cdot I \qquad (4)$$

where Γ_m is the motor torque and K_t is the motor torque constant ($K_t = K_{emf}$)

3.2.2 Gears Modeling

In this paper, we consider rigid gears' models. In this case, the model's mathematical expression depends only on the gear ratio *N*. Therefore, the output torque is obtained using the following relation: $\Gamma_g = N \cdot \Gamma_m$ and the speed of the motor shaft is obtained using: $\dot{q}_m = N \cdot \dot{q}$.

The gear's ratio is given by simple mathematical expressions (Henriot, 1991) or via the gears datasheet.

3.2.3 The Power Dissipation in Drive Chain

This section is the most essential in drive chain modeling. In fact, good power dissipation modeling helps to reproduce complex gear behaviors such as irreversibility. The power dissipation will be illustrated through the friction phenomenon.

In robotics, friction is often modeled as a function of joint velocity. It is based on static, dry and viscous friction (Khalil, 1999), (Abba, 2003). These models produce accurate simulation results with simple drive chain structures. However, in the presence of more complex mechanisms such as worm gears these models lack reliability.

To illustrate this phenomenon, we can compare the theoretical motor torque required to drive the LCA pivot axis in the case of a reversible transmission and the real measured motor torque. Figure 4 and 5 show the applied torques on the pivot axis during a 7°/sec and -7°/sec constant velocity movement.

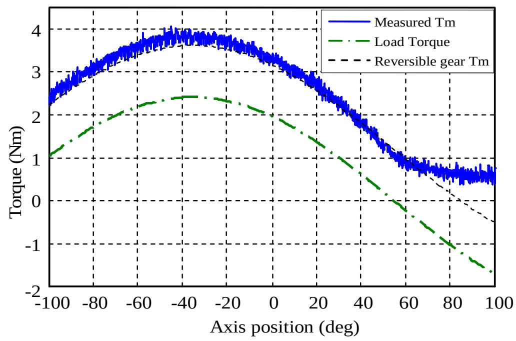


Figure 4: Motor and load torque variation during constant velocity rotation (7 °/s).

During this movement, the robot dynamic is represented by the following dynamic equation:

$$\Gamma_m = \Gamma_l + \Gamma_f \qquad (5)$$

where Γ_l is the load torque and Γ_f is the friction torque. Consequently, we expect that the motor torque will have the same behavior as the load torque because the friction torque is constant. However, these results reveal an important difference between the measured motor torque and the expected motor torque with a drive chain using only velocity friction model.

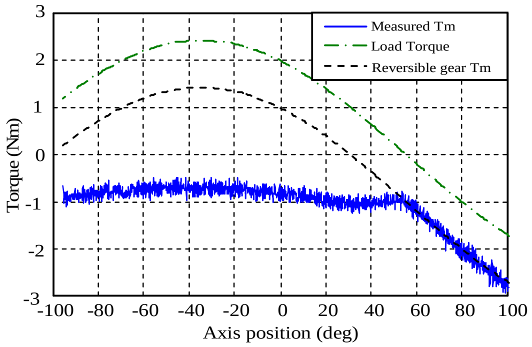


Figure 5: Motor and load torque variation during constant velocity rotation (-7 °/s).

We can see that the irreversibility seriously influences the motor torque. Actually, the irreversibility compensates the gravity torque when the load torque becomes driving. Therefore, it is essential to expand the friction model to take into consideration more variables such as motor torque and load torque in order to reproduce irreversibility in a simulation environment. Thus, the friction model Γ_f applied on motor shaft will have the following structure:

$$\Gamma_f = \Gamma_{fs}(\Gamma_m, \dot{q}_m) + \Gamma_{fv}(\dot{q}_m) + \Gamma_{fT}(\Gamma_m, \Gamma_l, \dot{q}_m) \qquad (6)$$

where :

$\Gamma_{fs}(\Gamma_m, \dot{q}_m)$: 4x1 vector of the static friction model

$\Gamma_{fv}(\dot{q}_m)$: 4x1 vector of the velocity friction model

$\Gamma_{fT}(\Gamma_m, \Gamma_l, \dot{q}_m)$: 4x1 vector of the torque friction model.

Γ_{fs} and Γ_{fv} are the classical friction terms used usually in drive chain modeling (Dupont, 1990), (Armstrong, 1998). While, Γ_{fT} presents the term that takes account of the irreversibility behavior.

$$\begin{aligned}\Gamma_{fTi}(\Gamma_{mi}, \Gamma_{li}, \dot{q}_{mi}) = \mu_{mi}(\Gamma_{mi}, \Gamma_{li}, \dot{q}_{mi}) \cdot \Gamma_{mi} \\ + \mu_{li}(\Gamma_{mi}, \Gamma_{li}, \dot{q}_{mi}) \cdot \Gamma_{li}\end{aligned} \qquad (7)$$

where $\mu_{mi}(\Gamma_{mi}, \Gamma_{li}, \dot{q}_{mi})$ and $\mu_{li}(\Gamma_{mi}, \Gamma_{li}, \dot{q}_{mi})$ are the motor and load friction dynamic coefficients.

Let's consider now the complete robot dynamic model:

$$\Gamma_m = J_m \cdot \ddot{q}_m + N^{-1} A(q)\ddot{q} + \Gamma_l + \Gamma_f \qquad (8)$$

where $\Gamma_l = N^{-1}(C(q,\dot{q})\dot{q} + Q(q))$ and J_m is the 4x4 motors and gears inertia matrix. By replacing (6) in (8) we obtain:

$$\begin{aligned}\Gamma_m = (J_m + N^{-2}A(q)) \cdot \ddot{q}_m + \Gamma_l + \Gamma_{fs}(\Gamma_m, \dot{q}_m) \\ + \Gamma_{fv}(\dot{q}_m) + \mu_m \cdot \Gamma_m + \mu_l \cdot \Gamma_l\end{aligned} \qquad (9)$$

where μ_m and μ_l are respectively 4x4 diagonal matrixes:

$$\mu_m = \mathrm{diag}\{[\mu_{mi}(\Gamma_{mi}, \Gamma_{li}, \dot{q}_{mi})]\,;\, i = 1,\ldots,4\}$$
$$\mu_l = \mathrm{diag}\{[\mu_{li}(\Gamma_{mi}, \Gamma_{li}, \dot{q}_{mi})]\,;\, i = 1,\ldots,4\}$$

By regrouping the terms of equation 11 we obtain:

$$\begin{aligned}\eta_m \cdot \Gamma_m = (J_m + N^{-2}A(q)) \cdot \ddot{q}_m + \eta_l \cdot \Gamma_l \\ + \Gamma_{fs}(\Gamma_m, \dot{q}_m) + \Gamma_{fv}(\dot{q}_m)\end{aligned} \qquad (10)$$

where $\eta_m = (I_{4\times 4} - \mu_m)$ and $\eta_l = (I_{4\times 4} + \mu_l)$.

The new terms η_m and η_l, which depend on Γ_m, Γ_l and $\dot{q}_m$, introduce the efficiency concept in the robot dynamic model. The next section will focus on the proposed approach used to calculate the drive chain efficiency coefficients.

4 EFFICIENCY COEFFICIENTS ESTIMATION

One of the complex issues in drive chain modeling is the estimation of the transmission efficiency coefficient. One technique consists of theoretically calculating the efficiency of each element of the drive chain using the efficiency definition (Henriot, 1991):

$$\eta = \frac{|\text{Received Power}|}{\text{Emitted Power}} = \frac{|P_{out}|}{P_{in}} \qquad (11)$$

The calculation of this coefficient requires the determination of the driving element whether it is the motor or the load. We talk then about the motor torque efficiency (η_m) or the load torque efficiency (η_l). Therefore, the received power "P_{in}" could be either from the motor or the load.

Actually, this method can be applied with simple gear mechanisms such as spur gears, whereas for complex gears, such as worm gears, the calculation of the efficiency coefficient using analytical formulas tends to be hard and inaccurate due to the lack of information concerning friction modeling as well as the complexity of the contact surface between gears' components (Henriot, 1991). The alternative that we propose is to experimentally identify the efficiency coefficient according to a functional state of the drive chain, for instance, when the load is driving the movement or when the motor is driving the movement. This leads us to create a state machine with the following inputs and outputs:

Table 1: State machine inputs and outputs.

Inputs	Outputs
Γ_m : motor torque (Tm) Γ_l : load torque (Tl) $\dot{q}_m$: motor velocity	η_m : motor efficiency η_l : load efficiency

Now, we will present the states and the criteria of states transitions that we have used for LCA robot drive chain modeling. The state machine includes two levels: the upper level that describes the motion (figure 6) and the lower level that describes the switch between motor driving and load driving states (figures 7, 8), and associates an efficiency coefficient for each state. In this level, the transition condition is the sign of the velocity.

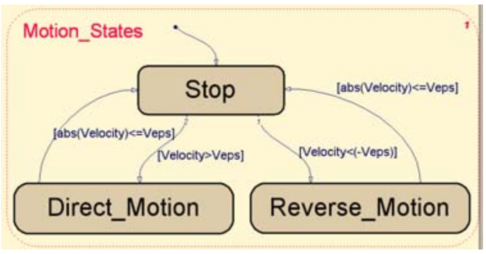


Figure 6: Motion state machine.

In the upper level, the transition condition is the sign of the velocity. In fact, for simulation convergence issue the drive chain is considered stopped when $|\dot{q}_m| < V_{eps}$, where V_{eps} is the stop velocity threshold.

In the lower level, a sub-state has been combined to each motion state:

- **The stop states (figure 7)**

During the stop phase, the drive chain is irreversible (the load torque cannot drive the movement). Motion is observed when the motor torque becomes superior to the load torque.

In the lower level, the state transition is based on the motor and load torque values. As for V_{eps} (figure 6), Tm_{eps} represents the motor torque threshold, it is used for simulation convergence issue ($Tm_{eps} = 10^{-5}$ Nm).

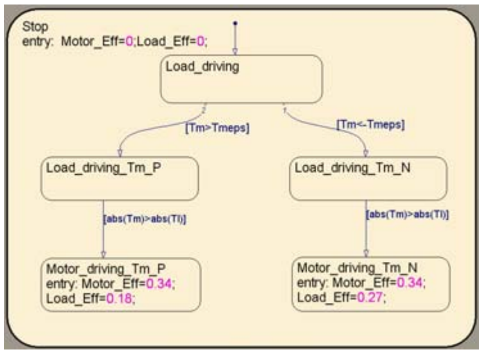


Figure 7: Stop state machine.

- **The direct motion states**

For the direct motion state (if V>0), we have four main states (figure 8), the states transitions are given by the following conditions: $\Gamma_m > 0$ and $\Gamma_l < 0$: the motor is driving; $\Gamma_m > 0$ and $\Gamma_l > 0$: we distinguish two states whether $\Gamma_l > \Gamma_m$ or not; and $\Gamma_m < 0$: the motor is braking (load driving)

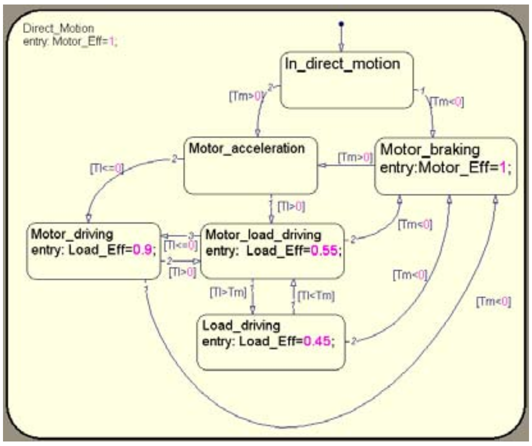


Figure 8: direct motion state machine.

- **The reverse motion states**

The reverse motion (V<0) state machine has the same structure as the direct motion one. We need to replace Γ_m and Γ_l by $-\Gamma_m$ and $-\Gamma_l$. The table 1 summaries the drive chain efficiency coefficients for each state: (Motor driving / Motor and load driving / Load driving).

Table 2: Drive chain efficiency coefficients.

States	Direct motion η_l	Reverse motion η_l
1-$\Gamma_m \cdot \Gamma_l < 0$	0.9	0.9
2-$\Gamma_m \cdot \Gamma_l > 0$ & $\|\Gamma_m\| < \|\Gamma_l\|$	0.55	0.16
3-$\Gamma_m \cdot \Gamma_l > 0$ & $\|\Gamma_m\| > \|\Gamma_l\|$	0.45	0.06

5 EXPERIMENTAL VALIDATION

The validation of the drive chain model has been done on the pivot axis. The efficiency coefficients have been identified using experimental measures.

We compare the open loop response of the pivot joint and the simulation results to a voltage input for both direct and reverse motion. Figure 9 shows the applied voltage on the motor pivot axis for direct motion. Figure 10 shows the experimental results (dashed curve) of current, velocity and position and those obtained in simulation (solid curve). We notice in that the simulation response represents the same behavior as the real mechanism. In this figure we distinguish four main phases: the starting phase 24s to 25s, the motor driving phase 25s to 37.8s, the load driving phase 37.8s to 4.2s and the braking phase 4.2s to 4.3s.

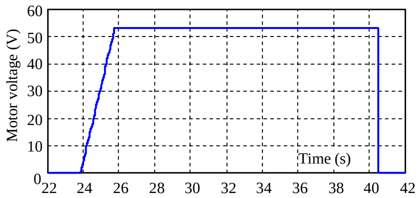


Figure 9: Open loop motor command voltage.

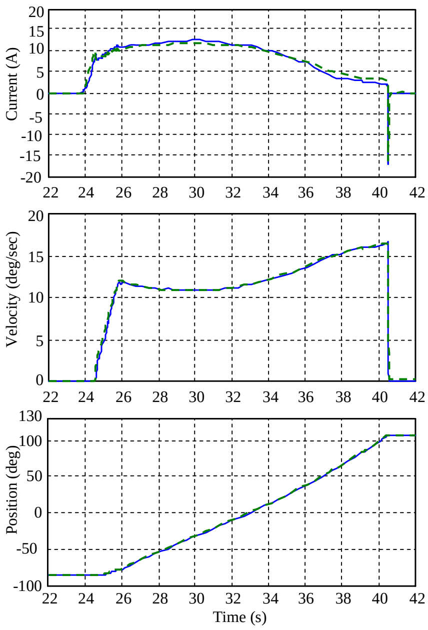


Figure 10: Direct motion outputs.

By comparing the obtained results, we notice that the differences are low for direct motion as well as for reverse motion. Therefore, these results prove that the used model is able to represent accurately the irreversibility property of the pivot drive chain.

6 CONCLUSIONS

In this paper, we presented a methodology in order to model the irreversibility characteristic in electromechanical drive chains. The proposed approach uses a macroscopic modeling of the gears, which are usually the origin of irreversibility in a drive chains. It consists of creating a state machine representing different functional states of the gears and attributing an efficiency coefficient to each specific state.

The validation of the proposed modeling was carried out on the Pivot axis of the LCA robot. The methodology has been tested in particular when the position trajectory leads to some transitions "motor driving to load driving" and the obtained results confirm the correctness of the used model.

The perspectives of this work concern two research orientations. The first one is the definition and the study of an automatic procedure to identify the efficiency coefficient for each state. The second one is the investigation of the trajectory planning and the control of robots with irreversible transmissions when considering state machines for gear's modeling.

REFERENCES

Abba G., Chaillet N. (1999) "*Robot dynamic modeling using using a power flow approach with application to biped locomotion*", Autonomous Robots 6, 1999, pp. 39–52.

Abba G., Sardain P.(2003), "*Modélisation des frottements dans les éléments de transmission d'un axe de robot en vue de son identification: Friction modelling of a robot transmission chain with identification in mind*", Mécanique & Industries, Volume 4, Issue 4, July-August, pp 391-396

Armstrong B. (1998), "*Friction: Experimental determination, modelling and compensation*", IEEE International Conference on Robotics and Automation, Philadelphia, PA, USA, April, vol. 3, pp. 1422–1427.

Dupont P.E.(1990) "*Friction modeling in dynamic robot simulation*", Robotics and Automation, 1990. Proceedings., IEEE International Conference, pp. 1370-1376 vol.2

Henriot G. (1991) "*Traité théorique et pratique des engrenages*", 5th ed. Dunod ed. vol. 1.

Khalil W., Dombre E. (1999) "*Modélisation identification et commande des robots*", 2nd ed. Hermes ed.

Pinard M. (2004), "*Commande électronique des moteurs électriques*", Paris Dunod ed. pp 157-163

ESTIMATION OF STATE AND PARAMETERS OF TRAFFIC SYSTEM

Pavla Pecherková, Jitka Homolová
Dept. of Adaptive Systems, Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic
Pod vodárenskou věží 4, 182 05 Prague 8, Czech Republic
nemcova@utia.cas.cz (P. Pecherková), *jkratoch@utia.cas.cz* (J. Homolová)

Jindřich Duník
Department of Cybernetics, Faculty of Applied Sciences, University of West Bohemia in Pilsen
Univerzitní 8, 306 14 Pilsen, Czech Republic
dunikj@kky.zcu.cz

Keywords: Traffic system, state space model, state estimation, identification, nonlinear filtering.

Abstract: This paper deals with the problem of traffic flow modelling and state estimation for historical urban areas. The most important properties of the traffic system are described. Then the model of the traffic system is presented. The weakness of the model is pointed out and subsequently rectified. Various estimation and identification techniques, used in the traffic problem, are introduced. The performance of various filters is validated, using the derived model and synthetic and real data coming from the center of Prague, with respect to filter accuracy and complexity.

1 INTRODUCTION

Intelligent traffic control is one possible way how to preserve or to improve capacity of current light controlled network. Generally, the problem can be solved by setting proper parameters of the signal lights. However, the suitable setting of these parameters is conditioned by exact knowledge of the current traffic situation at an intersection or micro-region[1].

Nowadays, when many intersection arms are equipped by detectors[2], the traffic situation can be sufficiently described by measurable intensity, occupancy, instant speed and hardly measurable queue length. Unfortunately, the knowledge of the queue length seems to be advantageous for a design of traffic control which can be based on the minimisation of the queue lengths (Kratochvílová and Nagy, 2004).

The key problem, either for estimation or control, is to specify the model of a micro-region. It is very interesting that the traffic situation can be described by a linear state space model (SSM) (Homolová and Nagy, 2005), where the directly immeasurable queue lengths are included in the state. Unfortunately, there are also some unknown parameters in the SSM, which cannot be determined from physical properties of the traffic situation and they have to be estimated as well.

Generally, there are two possibilities how to estimate the state and the parameters. The first possibility is based on an off-line identification of unknown parameters: prediction error methods (Ljung, 1999), instrumental variable methods (Söderström and Stoica, 2002), subspace identification methods (Viberg, 2002)) and subsequently on an on-line estimation of the state by the well-know Kalman Filter (KF) (Anderson and Moore, 1979). However, off-line identified time variant or invariant parameters represent the average values rather than the actual (true) parameters. The second possibility is based on the concurrent on-line estimation of the state and the parameters by suitable nonlinear estimation methods. There are two main groups of estimation methods for nonlinear systems, namely local and global methods (Sorenson, 1974). Although, the global methods are more sophisticated than local methods, they have significantly higher computation demands. Due to the computational efficiency, the stress will be mainly laid on the local methods, namely on the local derivative-free filtering methods (Nørgaard et al., 2000; Julier et al., 2000; Duník et al., 2005) and partially will be laid on a global method based on the Gaussian sums (Duník et al., 2005).

[1]One micro-region consists of several intersections with some detectors on the input and output roads. There must be at least one signal-controlled intersection.

[2]Detector is a inductive loop built in a cover of road, which is activated by a passing vehicle.

The aim of this paper is to apply and to compare the various estimation techniques in the area of the estimation of queue lengths and parameters of traffic system and to choose a suitable estimation technique with respect to the estimation performance and computational demands.

2 TRAFFIC MODEL

2.1 Traffic Model and its Parameters

This paper deals with the estimation of immeasurable queue length[3] on each lane[4] of controlled intersections in a micro-region. Lane can be equipped by one detector on the output and three detectors on the input: (i) detector on stop line, (ii) outlying detector, (iii) strategic detector. Ideally, each lane has all three types of the detectors but in real traffic system, the lane is usually equipped by one or two types of such detectors, due to the constrained conditions. The strategic detectors, which are most remote from a stop line, give the best information about the traffic flow at present.

The detector is able to measure following quantities:

- *Intensity* $I_{i,t}$ is the amount of passing unit vehicles on arm i per hour $[uv/h]$.
- *Occupancy* $O_{i,t}$ is the proportion of the period when the detector is occupied by vehicles [%].

Traffic flow can be influenced by signal lights setting. A signal scheme can be modified by split, cycle time and offset:

- *Cycle time* is time required for one complete sequence of signal phases $[s]$.
- *Split* z_t is proportional duration of the green light in a single cycle time [%].
- *Offset* is the difference between the start (or end) times of green periods at adjacent signals $[s]$.

The geometry of intersections and drivers demands determine other quantities which are needed for a construction of the traffic model. These quantities are valid for a long time period. They are:

- *Saturated flow* S is the maximal flow rate achieved by vehicles departing from the queue during the green period at cycle time $[uv/h]$.
- *Turning rate* $\alpha_{h,g}$ is the ratio of cars going from the h-th arm to the g-th arm [%].

[3]Queue length ξ_t is a number of vehicles waiting to proceed through an intersection (given in unit vehicles [uv] or meters [m]) per cycle time. It is supposed that 1 uv = 6 m.

[4]Each intersection arm consists of one or more traffic lanes.

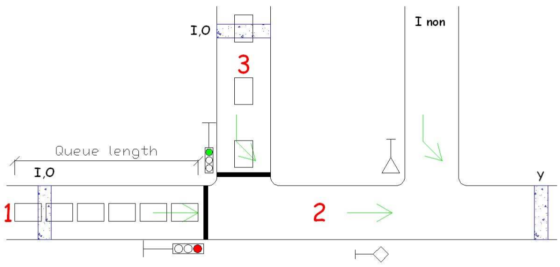


Figure 1: The micro-region: three-arm intersection with one unmeasured input.

The basic idea, which lies on the background of the model design, is the traffic flow conservation principle (Homolová and Nagy, 2005): "the new queue is equal to the remaining queue plus arrived cars minus departed cars".

The basic methodology of the traffic model design will be shown on a specific example. The micro-region consists of one three-arm controlled intersection with one unmeasured input, see Figure 1. Intersection is comprised of two one-way input arms (No.1 and 3) and one output arm (No. 2). The input arms are equipped by the strategic detectors and the output arm is equipped by the output detector. The unmeasured flow enters to the road No. 2 before the output detector. For the sake of the simplicity, the constant cycle time with two phases is considered.

In this case, the traffic system is described by the following model given by (1), (2), where $b_{i,t} = (1-\delta_{i,t}) \cdot I_{i,t} - \delta_{i,t} S_i$. Parameter $\delta_{i,t}$ is Kronecker function $(0\,,1)$, $\delta_{i,t} = 1$ if queue exist (on arm i at time t) and $\delta_{i,t} = 0$ in otherwise. Parameters $\kappa_{i,t}$, ϑ_t describe the relation between occupancy and queue length and parameter $\beta_{i,t}$ describes the relation between current and previous occupancy. The parameter $\lambda_{i,t}$ can be understood as a correction term to omit a zero occupancy. $I_{i,t}$ and $O_{i,t}$ are the input intensity and occupancy, respectively, measured by the input detectors. $Y_{i,t}$ is output intensity which is measured on the output detector. Mention that the subscript i stand for the number of intersection arm. The state and measurement noises are currently supposed to be Gaussian, i.e. $p(w_k) = \mathcal{N}\{w_k : 0, Q_k\}$ and $p(v_k) = \mathcal{N}\{v_k : 0, R_k\}$. The noise covariance matrices Q_k and R_k can be identified off-line by means of e.g. the prediction error method (Ljung, 1999) or the method based on the multi-step prediction (Šimandl and Duník, 2007). On-line noise covariance matrices estimation, so called adaptive filtering, has not been used due to the extensive computational demands.

Generally, traffic model can be described in matrix

$$\underbrace{\begin{bmatrix} \xi_{1,t+1} \\ \xi_{3,t+1} \\ O_{1,t+1} \\ O_{3,t+1} \end{bmatrix}}_{x_{t+1}} = \underbrace{\begin{bmatrix} \delta_{1,t} & 0 & 0 & 0 \\ 0 & \delta_{3,t} & 0 & 0 \\ \kappa_{1,t} & 0 & \beta_{1,t} & 0 \\ 0 & \kappa_{3,t} & \vartheta_t & \beta_{3,t} \end{bmatrix}}_{A_t} \cdot x_t + \underbrace{\begin{bmatrix} -b_{1,t} & 0 \\ 0 & -b_{3,t} \\ 0 & 0 \\ 0 & 0 \end{bmatrix}}_{B_t} \cdot \underbrace{\begin{bmatrix} z_{1,t} \\ z_{2,t} \end{bmatrix}}_{z_t} + \underbrace{\begin{bmatrix} I_{1,t} \\ I_{3,t} \\ \lambda_{1,t} \\ \lambda_{3,t} \end{bmatrix}}_{F_t} + w_t \tag{1}$$

$$\underbrace{\begin{bmatrix} Y_{2,t+1} \\ O_{1,t+1} \\ O_{3,t+1} \end{bmatrix}}_{y_{t+1}} = \underbrace{\begin{bmatrix} -\alpha_{1,2} & -\alpha_{3,2} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{C} \cdot x_{t+1} + \underbrace{\begin{bmatrix} \xi_{1,t} \\ \xi_{3,t} \\ 0 \\ 0 \end{bmatrix}}_{G_t} + v_{t+1} \tag{2}$$

form as follows:

$$x_{t+1} = A_t x_t + B_t z_t + F_t + w_t \tag{3}$$
$$y_{t+1} = C x_{t+1} + G_t + v_{t+1} \tag{4}$$

The last comment deals with the system initial condition. The starting time is chosen at early morning hours, when it can be supposed that there is no traffic in the micro-region and thus the system initial state is perfectly known and it is $x_0 = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^T$.

2.2 Nonlinearities in Traffic Model

The traffic model becomes nonlinear in two main cases. The first case is when the traffic system has one or more unmeasured inputs or outputs (and the particular intensities should be estimated). The second case is when the parameters $\kappa_{i,t}$, $\beta_{i,t}$, $\lambda_{i,t}$, and ϑ_t, which cannot be determined from the physical properties and from construction dispositions of the micro-region, are estimated as a part of the state.

To find an actual estimates of the parameters, it is necessary to estimate them on-line. One of the possibilities is to extend the system state with these parameters $\tilde{x}_t = [x_t^T, \kappa_{1,t}, \kappa_{3,t}, \beta_{1,t}, \beta_{3,t}, \lambda_{1,t}, \lambda_{3,t}, \vartheta_t]^T$ (Anderson and Moore, 1979). This extension inevitably also leads to a nonlinear SSM

$$\tilde{x}_{t+1} = f_t(\tilde{x}_t, z_t) + w_t \tag{5}$$
$$y_{t+1} = \tilde{C}\tilde{x}_{t+1} + G_t + v_{t+1} \tag{6}$$

and to an application of appropriate nonlinear estimation techniques. The variables with tildes stand for the variables which had to be modified due to the extension of the state.

It should be also mentioned that the concurrent estimation of the state and parameters is also advantageous for the unusual traffic situations, e.g. accidents, when the estimator adapts the model and the estimated results are significantly more exact towards the model with invariant parameters.

3 STATE ESTIMATION TECHNIQUES

This section is devoted to a brief introduction of possible estimation methods which can be used for the estimation of traffic system state. With respect to the nature of this problem only, a special part of the estimation problem will be considered, namely the filtering.

The aim of the filtering is to find a state estimate in the form of the probability density function of the state x_t at the time instant t conditioned by the measurements $y^t = [y_0, y_1, \ldots, y_t]$ up to the time instant t, i.e. the conditional pdf $p_{x_t|y^t}(x_t|y^t)$ is looked for. General solution to the filtering problem is given by the Bayesian Recursive Relations (BRRs) (Anderson and Moore, 1979).

The exact solution of the BRRs is possible only for a few special cases, e.g. for linear Gaussian system (with known parameters). In other cases, such as linear system with unknown parameters, nonlinear and/or non-Gaussian systems, it is necessary to apply some approximative method, either local or global.

The local methods are often based on approximation of the nonlinear functions in the state or measurement equation so that the technique of the Kalman Filter design can be used for the BRRs solution. This approach causes that all conditional probability density functions (pdfs) of the state estimate are given by the first two moments. This rough approximation of posterior estimates induces local validity of the state estimates and consequently impossibility to ensure the convergence of the local filter estimates. The resulting estimates of the local filters are suitable mainly for point estimates. On the other hand, the advantage of the local methods can be found in the simplicity of the BRRs solution. Generally, there are two main approaches in the local filter design. The first possibility is to approximate the nonlinear function in the model by means of the Taylor expansion first or second order, which leads e.g. to the Extended Kalman Filter, or by means of the Stirling's polynomial interpolation, which leads to the Divided Difference Filter first or second order, abbreviated as (DD1), (DD2) or together as (DDFs) (Nørgaard et al., 2000; Duník et al., 2005). The second possibility, often used in the local filter design, is based on the approximation of state estimates by a set of deterministically chosen points. This method is known as the Unscented Transformation and its application in the local filter design leads to e.g. the Unscented Kalman Filter (UKF) (Julier et al., 2000; Duník et al., 2005).

The global methods are rather based on approximation of the conditional pdf of the state estimate of some kind to accomplish better state estimates.

Due to the higher computational demands of the global methods, the main stress will be laid on the local methods especially on the derivative-free local methods, namely the DD1, the DD2, and the UKF. The derivative-free methods were chosen because of there is no need of computations of derivatives of nonlinear functions (Duník et al., 2005) which is tedious especially for high dimensional systems like traffic systems. However, some attention will be paid on the Gaussian sum approach as a representative of global methods. Moreover, the KF with off-line identification methods will be considered as well.

4 ANALYSIS OF MODEL

In the previous sections, the model design, estimation and identification techniques were discussed and it was also mentioned that the quality of the model affects the estimation performance of all filters. From the more detailed analysis of the traffic model, it is evident that the estimated state has a backward impact on the model through the parameter δ_t. That is the main weakness of the model because δ_t depends on the queue length which is estimated. In other words, δ_t can be understood as a parameter which switches between two models representing peak and off-peak hours. The problem arises in the situations when the traffic flow is in the transition from off-peak to peak hours. Then, δ_t can be switched from 0 to 1 although the real traffic flow still corresponds to value 0 and vice versa, due to non-exact state estimate.

There are two possibilities how to rectify this problem. The first one is based on the modification of the state equation(s) describing the evolution of the queue length (the first two equations in (1)). The discontinuous equation is approximated by the continuous approximation based on the hyperbolic tangent, as it is depicted in Figure 2, where the relation

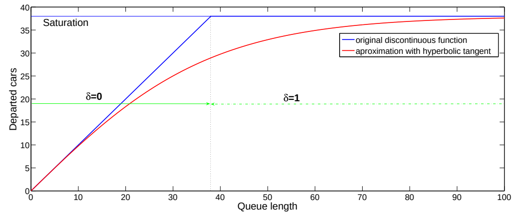


Figure 2: The approximation of the discontinuous function in the state equation.

between queue lengths and number of departed cars is shown. Then, the continuous approximation prevents from the bad switching of the models. Note that such approximation is done for all intersection arms. The second possibility is based on the Gaussian sum method and on the multi-model approach. It is still assumed that parameter δ_t belongs into the discrete set $\{0,1\}$ but at each time instant both values are used and the most probable value is looked for and then chosen. In case of more arms, all possible combinations of $\delta_{i,t}$ are tested and the most probable combination is chosen.

5 NUMERICAL ILLUSTRATIONS

In this section, the different micro-regions, either synthetic or real, will be described and the estimation task will be performed on each of them.

5.1 Synthetic Micro-regions

Micro-region with short queues: Let a micro-region consisting of one four-arm controlled intersection be considered, see Figure 3. All input roads are equipped by the strategic detectors. For estimation, the data from real traffic network supplemented with synthetic data was used. The queue lengths, supposed to be "true", and the missing output intensities were determined with simulation software AIMSUN[5].

The traffic model was built analogously to the model (1), (2). The original state has dimension $dim(x_t) = 8$ (queues and occupancies on four arms) and extended state has dimension $dim(\tilde{x}_t) = 24$ (original state and unknown parameters $\kappa_{i,t}$, $\beta_{i,t}$, $\lambda_{i,t}$, $\vartheta_{i,j,t}$, $i, j = 1, \ldots, 4$).

All local filters show very similar estimation performance in the traffic problem. The reason can be found in a absence from significant nonlinearities (Duník et al., 2005). Thus the results of the DD1 will be presented only.

[5]AIMSUN is a simulation software tool which is able to reproduce the traffic condition of any traffic network.

The local filter will be applied on three variants of model: (A) standard model illustrated by (1), (2), (B) model with continuous approximation of δ functions, and (C) multi-model approach. The variant (A) works with δ as Kronecker function. In the variant (B), the switching of parameter δ is replaced by the approximation with hyperbolic tangent. In the variant (C), the model includes Kronecker function δ as well but switching is replaced by multi-modal approach (by "brute" force) where the state of models with all combinations of δ functions are estimated.

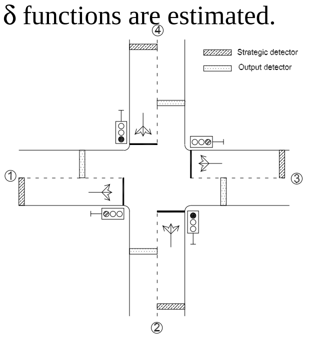


Figure 3: The micro-region: one four-arm controlled intersection.

Table 1 shows a comparison of all three variants with respect to the estimation performance for different types of traffic flows and to the computation load. The weak traffic flow is characterised by small intensities and on the other hand working days are rather characterised by high intensities. The estimation performance is measured by the Mean Square Error (MSE) of estimates queue length on one arm in $[uv^2]$. The average queue length is about 20 cars in all arms.

The best estimation performance, with respect to the MSE, shows the approach (C). On the other hand, the utilisation of the original model (A) leads to the least computational demand. With respect to Table 2, where the maximal differences between real and estimate queue are given, the best approach is multi-model. The same case is with the number of unsuitable values, which are defined as $\xi_{real} + 4 < \xi_{est}$.

For the needs of traffic control, the estimation method should be sufficiently exact (with small number of estimates which exceed allowable bound) and computational efficient. From the results, the best choice seems to be approach (B) and the DD1.

Table 1: Comparison of the different approaches with respect to the function δ (criterion no. 1).

	2 days (weekend) (1920 samples)		5 days (workweek) (4800 samples)	
	MSE	Time	MSE	Time
(A)	10.9	26 s	12.5	62 s
(B)	9.4	49 s	7.9	115 s
(C)	8.9	404s	7.1	858 s

Table 2: Comparison of the different approaches with respect to the function δ (criterion no. 2).

	Maximal difference	No. of unsuitable values (from 19200 data)	[%]
(A)	13.3	3284	17.0
(B)	15.7	1183	6.1
(C)	24.0	1086	5.7

Mention that the KF for micro-region with all measured arms provides little bit worse but comparable results with local filters and approach (A).

Micro-region with long queues: The typical micro-region has some arms unequipped by the detector. This situation together with a long queue lengths on the access roads will be illustrated in this part.

Let a micro-region consisting of one one-way three-arm controlled intersection and one unmeasured input given by (1), (2) be considered, see Figure 1. The two input roads have strategic detectors and one output road is equipped by an output detector. Moreover, the long queues on the access road will be considered to illustrate the situation with permanently engaged detectors which are not able to provide sufficient information about the current traffic flow.

For this simulation, the real data was used but the intensities were artificially increased. The "true" queue lengths were computed subsequently by means of simulation.

The four-dimensional state equation (1) includes eight a priory unknown parameters (κ_t, β_t, λ_t, ϑ_t for each measured input road). All these parameters and the immeasurable intensity can be estimated by the local filters. The KF is able to estimate the original state only and so for the application of the KF, the model was identified off-line by the prediction error method (Ljung, 1999) and the unmeasured intensity was considered as long time average value.

The results of the DD1 were compared with the KF results. The "true" and estimated queue lengths of both filters are depicted in Figure 4. It clearly shows the estimation improvement of the DD1, which is significantly better and perfectly matches the "true" queue on arm no. 3 contrary to the KF results.

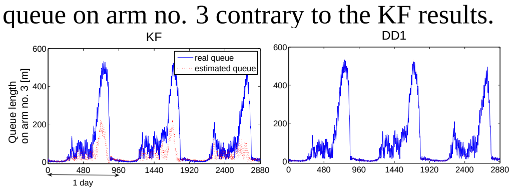


Figure 4: Comparison of the KF and local filter in the problem of queue estimation in the micro-region with unmeasured input intensity.

5.2 Real Micro-region

For the last experiment, the estimation of queue lengths was tested on data from the micro-region in

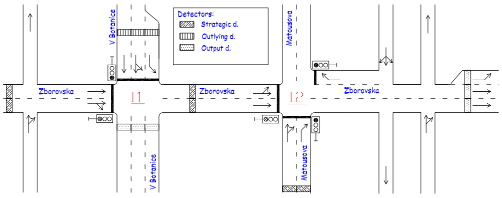


Figure 5: The micro-region: two four-arm controlled and three uncontrolled intersections.

Prague including two four-arm controlled intersections and three uncontrolled ones. The arms are one-way and they consist of several lanes, see Figure 5. Two input arms are equipped by strategic detectors and one input arm by outlying detector. Output detectors are in two arms.

For queue estimation, the state space model (3) and (4) without any approximation was used. The extended state is $dim(\tilde{x}_t) = 50$.

The input and output intensities, occupancies and green times were measured on the real traffic net during several months with sample period 90 sec. The "true" queue length was again determined with simulation software AIMSUN. Comparing of the estimated states and simulated ones shows that the estimation depends on the type of input detector. In lanes, which are equipped by the strategic detector, was usually $MSE \approx 17$, the error is about 10% with respect to the maximal queues. On the other hand, in lanes, equipped by the outlying detectors only, the good results was only in cases where the queue did not exceed the outlying detector. This is residual queue and for evaluation of traffic situation is not interesting.

The experiments show that the nonlinear estimation methods are a sufficient tool for estimation of the queue lengths, even in a real network. The dimension of the state, which is extended due to the estimation of parameters or intensities, increases the computation time, however, the computational demands remains still feasible (namely the DD1 and model variant (B)).

6 CONCLUSION

The problem of the queue length estimation, which is hardly measurable quantity, was considered in this paper. For the queue estimation, the mathematical model was presented, which describes the micro-region including its physical properties and taking into account behaviour of drivers. The disadvantage of the model was highlighted and two possible solutions of that were proposed. The theoretical results were illustrated by the numerical examples based on the synthetic or real data. It was shown that the Kalman Filter is suitable for situations where all quantities are measured. In other cases, it is advantageous to use a nonlinear filters for concurrent estimation of the state and parameters or possibly other unmeasurable quantities together with improved model.

ACKNOWLEDGEMENTS

The work was supported by the Ministry of Education, Youth and Sports of the Czech Republic, project No. 1M0572 and by the Ministry of Transport of the Czech Republic, project No. 1F43A/003/120.

REFERENCES

Anderson, B. D. O. and Moore, S. B. (1979). *Optimal Filtering*. Englewood Cliffs, New Jersey: Prentice Hall Ins.

Duník, J., Šimandl, M., Straka, O., and Král, L. (2005). Performance analysis of derivative-free filters. In *Proceedings of the 44th IEEE Conference on Decision and Control, and European Control Conference ECC'05*, pages 1941–1946, Seville, Spain. ISBN: 0-7803-9568-9, ISSN: 0191-2216.

Homolová, J. and Nagy, I. (2005). Traffic model of a microregion. In *Preprints of the 16th IFAC World Congress*, pages 1–6, Prague, Czech Republic.

Julier, S. J., Uhlmann, J. K., and Durrant-White, H. F. (2000). A new method for the nonlinear transformation of means and covariances in filters and estimators. *IEEE Transactions On AC*, 45(3):477–482.

Kratochvílová, J. and Nagy, I. (2004). Traffic control of microregion. In Andrýsek, J., Kárný, M., and Kracík, J., editors, *CMP'04: Multiple Participant Decision Making, Theory, algorithms, software and app.*, pages 161–171, Adelaide. Advanced Knowledge Int.

Ljung, L. (1999). *System identification: theory for the user*. UpperSaddle River, NJ: Prentice-Hall.

Nørgaard, M., Poulsen, N. K., and Ravn, O. (2000). New developments in state estimation for nonlinear systems. *Automatica*, 36(11):1627–1638.

Söderström, T. and Stoica, P. (2002). Instrumental variable methods for system identification. *Circuits, Systems, and Signal Processing*, 21(1):1–9.

Sorenson, H. W. (1974). On the development of practical nonlinear filters. *Inf. Sci.*, 7:230–270.

Viberg, M. (2002). Subspace-based state-space system identification. *Circuits, Systems, and Signal Processing*, 21(1):23–37.

Šimandl, M. and Duník, J. (2007). Multi-step prediction and its application for estimation of state and measurement noise covariance matrices. Technical report. University of West Bohemia in Pilsen, Department of Cybernetics.

BAYESIAN ADAPTIVE SAMPLING FOR BIOMASS ESTIMATION WITH QUANTIFIABLE UNCERTAINTY

Pinky Thakkar
Department of Computer Engineering, San Jose State University, San Jose, CA 95192, USA
pinkythakkar@gmail.com

Steven M. Crunk, Marian Hofer, Gabriel Cadden, Shikha Naik
Department of Mathematics, San Jose State University, San Jose, CA 95192, USA
crunk@math.sjsu.edu, marianhofer@gmail.com, gabec81@yahoo.com, naikshikha@hotmail.com

Kim T. Ninh
Department of Computer Engineering, San Jose State University, San Jose, CA 95192, USA
Tnk45@yahoo.com

Keywords: Adaptive Sampling, Bayesian Inference, BRDF, Maximum Entropy, Optimal Location Selection.

Abstract: Traditional methods of data collection are often expensive and time consuming. We propose a novel data collection technique, called Bayesian Adaptive Sampling (BAS), which enables us to capture maximum information from minimal sample size. In this technique, the information available at any given point is used to direct future data collection from locations that are likely to provide the most useful observations in terms of gaining the most accuracy in the estimation of quantities of interest. We apply this approach to the problem of estimating the amount of carbon sequestered by trees. Data may be collected by an autonomous helicopter with onboard instrumentation and computing capability, which after taking measurements, would then analyze the currently available data and determine the next best informative location at which a measurement should be taken. We quantify the errors in estimation, and work towards achieving maximal information from minimal sample sizes. We conclude by presenting experimental results that suggest our approach towards biomass estimation is more accurate and efficient as compared to random sampling.

1 INTRODUCTION

Bayesian Adaptive Sampling (BAS) is a methodology which allows a system to examine currently available data in order to determine new locations at which to take new readings. This procedure leads to the identification of locations where new observations are likely to yield the most information about a process, thus minimizing the required data that must be collected. As an example of the application of this methodology, we examine the question of standing woods in the United States.

In order to estimate the amount of carbon sequestered by trees in the United States, the amount of standing woods must be estimated with quantifiable uncertainty (Wheeler, 2006). Such estimates come from either satellite images or near ground measurements. The amounts of error in the estimates from these two approaches are currently unknown. To this end, an autonomous helicopter with differential GPS (Global Positioning System), LIDAR (Light Detection and Ranging), stereo imagers, and spectrometers has been developed as a testing platform for conducting further studies (Wheeler, 2006). These instruments are capable of measuring the reflectance data and the location of the Sun and helicopter in terms of the zenith and the azimuth angles (Figure 1). The objective is to develop a controlling software system for this robotic helicopter, which optimizes the required ground sampling.

The first simplistic data collection method is to conduct an exhaustive ground sampling i.e. to send the helicopter to every possible location. The second approach is to perform random sampling until the estimates have acceptable standard errors. Although random sampling presents a possibility that the helicopter will take samples from the locations that offer the greatest amount of information and therefore reduce the needed sample size, there is no guarantee that such a sample set will be chosen every time. The third and more efficient method is to take only a few samples from "key" locations that are expected to offer the greatest amount of information. The focus of this paper is to develop a methodology that will identify such key locations from which the helicopter should gather data.

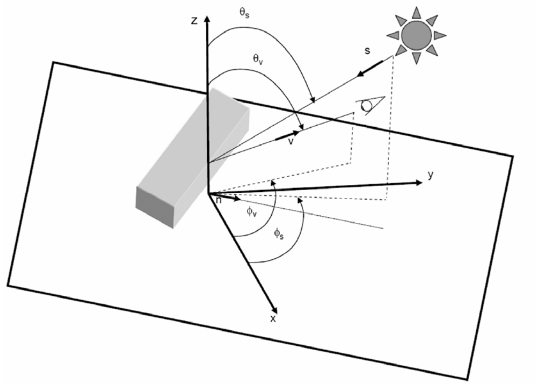


Figure 1: θ_S, ϕ_S are the zenith and the azimuth angles of the Sun, and θ_V, ϕ_V are the zenith and the azimuth angles of the view, respectively (Wheeler, 2006).

In the work described here, the key locations are identified using current and previously collected data. The software works in tandem with the sampling hardware to control the helicopter's position. Once a sample has been taken, the data are fed into the system, which then calculates the next best location to gather further data. Initially, the system assumes an empirical model for the ground being examined. With each addition of data from the instruments, the parameter estimates of the model are updated, and the BAS methodology is used to calculate the helicopter's next position. This process is repeated until the estimated uncertainties of the parameters are within a satisfactory range. This method allows the system to be adaptive during the sampling process and ensures adequate ground coverage.

The methodology employs a bi-directional reflectance distribution function (BRDF), in which the calculation of the amount of reflection is based on the observed reflectance values of the object, and the positions of the Sun and the viewer (Nicodemus, 1970). The advantage of using this function is that it enables the system to compensate for different positions of the Sun during sampling. Once the reflectance parameters are estimated, BAS uses the principle of maximum entropy to identify the next location where new observations are likely to yield the most information.

In summary, the BAS methodology allows the system to examine currently available data with regards to previously collected data in order to determine new locations at which to take new reflectance readings. This procedure leads to the identification of locations where new observations are likely to yield the most information.

2 RELATED WORK

Computing view points based on maximum entropy using prior information has been demonstrated by Arbel et al., 1970. They used this technique to create entropy maps for object recognition. Vazquez et al., 2001 also demonstrated a technique for computing good viewpoints; however their research was based on Information Theory. Whaite et al., 1994 developed an autonomous explorer that seeks out those locations that give maximum information without using a priori knowledge of the environment. Makay, 1992 used Shannon's entropy to obtain optimal sample points that would yield maximum information. The sample points are taken from the locations that have largest error bars on the interpolation function. In our work, the optimal locations that offer maximum amount of information are identified using the principle of maximum entropy, where the maximization is performed using techniques suggested by Sebastiani et al., 2000.

3 MODEL

The model for the data used in our framework is based on the semi-empirical MISR (multi-angle imaging spectrometer) BRDF Rahman model (Rahman et al., 1993):

$$r(\theta_s,\theta_v,\phi_s,\phi_v) = \rho\left[\cos(\theta_s)\cos(\theta_v)\{\cos(\theta_s)+\cos(\theta_v)\}\right]^{k-1} \cdot \exp\left(-b\cdot p(\Omega)\right)\cdot h(\theta_s,\theta_v,\phi_s,\phi_v) \quad (1)$$

where

$$h(\theta_s,\theta_v,\phi_s,\phi_v) = 1 + \frac{1-\rho}{1+G(\theta_s,\theta_v,\phi_s,\phi_v)} \quad (2)$$

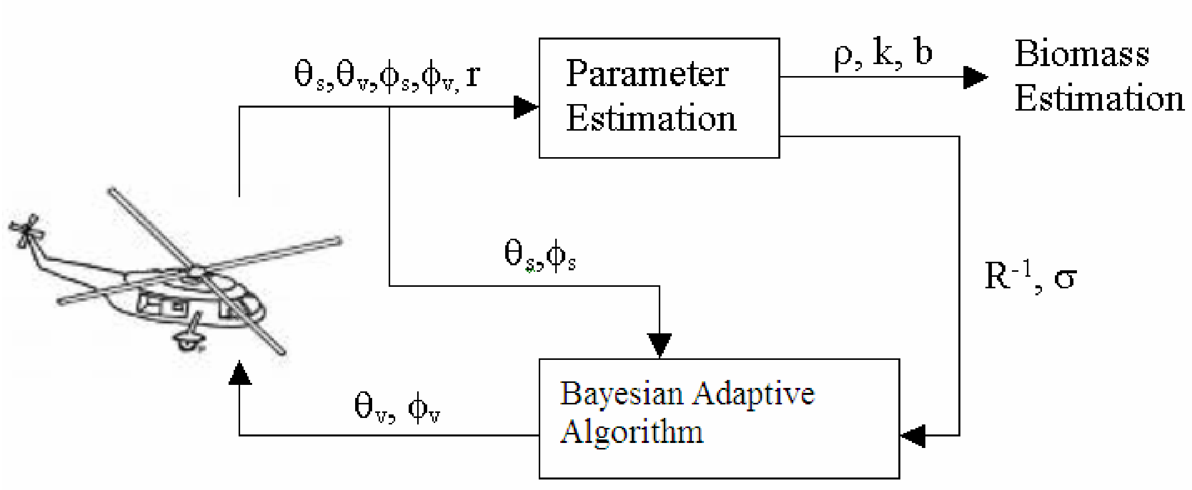


Figure 2: Overview of Bayesian Adaptive Sampling.

$$G(\theta_s,\theta_v,\phi_s,\phi_v) = \sqrt{\tan^2(\theta_s)+\tan^2(\theta_v)-2\tan(\theta_s)\tan(\theta_v)\cos(\phi_s-\phi_v)} \quad (3)$$

$$p(\Omega) = \cos(\theta_s)\cos(\theta_v)+\sin(\theta_s)\sin(\theta_v)\cos(\phi_s-\phi_v) \quad (4)$$

where $r(\theta_s,\theta_v,\phi_s,\phi_v)$ is the measured reflectance, ρ is the surface reflectance at zenith, k is the surface slope of reflectance, b is a constant associated with the hotspot, or "antisolar point" (the point of maximum reflectivity, which is the position where the sensor is in direct alignment between the Sun and the ground target), θ_s,ϕ_s are the zenith and the azimuth angles of the Sun, respectively (Figure 1), and θ_v,ϕ_v are the zenith and the azimuth angles of the view, respectively (Figure 1).

4 METHODOLOGY

Our framework consists of the following two steps:

1. Parameter Estimation: In this step, we estimate the values of the parameters (ρ, k and b), and their covariance matrix and standard errors, given the data collected to date of the amount of observed reflected light, and the zenith and azimuth angles of the Sun and the observer.
2. Bayesian Adaptive Sampling (Optimal Location Identification): In this step, we use the principle of maximum entropy to identify the key locations from which to collect the data.

Once the key location is identified, the helicopter goes to that location and the instruments on the helicopter measure the reflectance information. This information is then fed into the Parameter Estimation stage and the new values of the parameters (ρ, k and b) are calculated. This process is repeated until the standard errors of the parameters achieve some predefined small value, ensuring adequacy of the estimated parameters (Figure 2).

5 IMPLEMENTATION

5.1 Parameter Estimation

The input to this module is the observed reflectance value (r), zenith and azimuth angles of the Sun (θ_s,ϕ_s), and zenith and azimuth angles of the observer (θ_v,ϕ_v). The parameters (ρ, k and b) are estimated using the following iterated linear regression algorithm:

First, a near linear version of this model is accomplished by taking the natural logarithm of $r(\theta_s,\theta_v,\phi_s,\phi_v)$, which results in the following equation:

$$\ln r(\theta_s,\theta_v,\phi_s,\phi_v) = \ln \rho + (k-1)\cdot \ln\left[\cos(\theta_s)\cos(\theta_v)\{\cos(\theta_s)+\cos(\theta_v)\}\right] - b.p(\Omega) + \ln h(\theta_s,\theta_v,\phi_s,\phi_v) \quad (5)$$

Note that aside from the term $\ln h(\theta_s,\theta_v,\phi_s,\phi_v)$, which contains a nonlinear ρ, the function ln(r) is linear in all three parameters, $\ln(\rho)$, k, and b. "Linearization" of $\ln(h)$ is accomplished by using the value of ρ from the previous iteration, where at iteration *n* in the linear least-squares fit $h(\theta_s,\theta_v,\phi_s,\phi_v)$ is taken to be the constant

$$h^{(n)}(\theta_s,\theta_v,\phi_s,\phi_v) = 1 + \frac{1-\rho^{(n-1)}}{1+G(\theta_s,\theta_v,\phi_s,\phi_v)} \quad (6)$$

where $\rho^{(0)}$ is set equal to zero.

Second, regression is performed on our linearized model to calculate the estimates of the following quantities:

- ρ, k and b, the parameters
- R^{-1}, covariance matrix of the estimated parameters (ρ, k and b)
- σ, the standard deviation of the errors, (which are assumed to be independent identically distributed random variables from a normal distribution with zero mean)

Third, the current estimated value of ρ is then used in $h(\theta_s,\theta_v,\phi_s,\phi_v)$, and regression is again performed.

This procedure is repeated until the estimate of ρ converges.

5.2 Bayesian Adaptive Sampling

This module identifies the best informative location (θ_v,ϕ_v) to which to send the helicopter. We employ the principle of maximum entropy, in which the available information is analyzed in order to determine a unique epistemic probability distribution. The maximization is performed as per techniques suggested by Sebastiani et al., 2000, where in order to maximize the amount of information about the posterior parameters, we should maximize the entropy of the distribution function. Mathematically, maximizing the entropy is achieved by maximizing

$$\log[\left| (X'\Sigma^{-1}X+R) \right|] \quad (7)$$

where Σ is covariance matrix of the error terms, R^{-1} is covariance matrix of the estimates of the parameter ρ, k and b, and *X* is matrix of input variables where each row in *X* is associated with one observation

$$X = \begin{bmatrix} 1 & x_1 & x_2 \end{bmatrix} \quad (8)$$

where

$$x_1 = \ln\left[\cos(\theta_s)\cos(\theta_v)\{\cos(\theta_s)+\cos(\theta_v)\}\right]$$
$$x_2 = p(\Omega) = -\cos(\theta_s)\cos(\theta_v) + \sin(\theta_s)\sin(\theta_v)\cos(\phi_s-\phi_v)$$

Under the assumption that the errors are independent normally distributed random variables with mean zero and variance σ^2, (7) reduces to maximizing

$$\left| I + \left(1/\sigma^2\right)X'XR^{-1} \right|. \quad (9)$$

Note that σ^2 and R^{-1} are estimated in module 1 and are thus at this stage assumed to be known quantities. The matrix *X* contains both past observations, in which case all elements of each such row of *X* are known, and one or more new observations where the zenith and the azimuth angles of the Sun (θ_s,ϕ_s) are known, so the only remaining unknown quantities in (9) are the values of θ_v and ϕ_v (the zenith and azimuth angles of the helicopter viewpoint) in rows associated with new observations. Thus, the new location(s) to which the helicopter will be sent are the values of θ_v and ϕ_v in rows of *X* associated with new observations that maximize (9).

6 EXPERIMENT

We conduct two simulated experiments in which the estimates of the model parameters are calculated. In the first experiment, “Estimation Using Random Observations”, the data is collected by sending the helicopter to random locations. In the second experiment, “Estimation using BAS”, the data is collected using BAS.

The experiments are conducted under the following assumptions:

- The view zenith angle (θ_v) is between 0 and $\pi/2$, and the view azimuth angle (ϕ_v) is between 0 and 2π ($\approx$6.283185).
- The Sun moves 2π radians in a 24-hour period, i.e., at the rate of slightly less then 0.005 radians per minute.
- It takes about 2 minutes for the helicopter to move to a new location. Thus, the position of the Sun changes approximately 0.01 radians between measurements.

In our simulation, the true values of the parameters ρ, k and b are 0.1, 0.9, and -0.1, respectively. For the purpose of this paper, the observed values were simulated with added noise from the process with known parameters. This allows us to measure the efficacy of the algorithm in minimizing the standard errors of the parameter estimates, and also the estimates of the parameters. In actual practice, the parameters would be unknown, and we would have no way of knowing how close our estimates are to the truth, that is, if the estimates are as accurate as implied by the error bars.

6.1 Estimation using Random Observations

In this experiment, we send the helicopter to 20 random locations to collect data. Starting with the fifth observation, we use the regression-fitting algorithm on the collected input data set (the observed reflectance information, and the positions of the Sun and the helicopter), to estimate the values of the parameters ρ, k, b as well as their standard errors. Table 1 shows the results of this experiment

6.2 Estimation using BAS

In this experiment, the first five locations of the helicopter are chosen simultaneously using an uninformative prior distribution (i.e., as no estimate of R^{-1} has yet been formed; it is taken to be $\sigma^2 I$) and an X matrix with five rows in which the position of the Sun (θ_s, ϕ_s) is known and (9) is maximized over five pairs of helicopter viewpoints θ_v and ϕ_v.

Subsequently, we use BAS to calculate the next best informative location for the helicopter to move to in order to take a new reflectance observation., in which case the X matrix contains rows associated with previous observations, and (9) is maximized over a single new row of the X matrix in which the position of the Sun (θ_s, ϕ_s) is known and the only unknowns are a single pair of helicopter viewpoint values, θ_v and ϕ_v, in the last row of the X matrix.

Table 2 shows the results from this experiment. In both experiments, estimates of the parameters, along with their standard errors, cannot be formed until at least five observations have been taken.

7 RESULTS

In this section, we compare and analyze the results of our two experiments. The comparison results (Figure 3, Figure 4 and Figure 5) show that the estimates using the data from the "well chosen" locations using BAS are closer to the true values, $\rho = .1$, $k = 0.9$ and $b = -0.1$, than the estimates based on data from the randomly chosen locations. Also, the error bars using BAS are much shorter indicating higher confidence in the estimates of the parameters based on the "well chosen locations", i.e., the length of the error bar for the estimate calculated using data/observations from five well chosen locations is as short as the error bar based on data collected from 20 random locations.

Within each figure (Figure 3, Figure 4 and Figure 5), the horizontal axis indicates the number of observations between five and twenty that were used in forming the estimates. The vertical axis is on the scale of the parameter being estimated. Above each observation number, an "o" represents the estimate (using the data from the first observation through the observation number under consideration) of the parameter using the randomly chosen locations and the observations from those locations. The "x" represents the estimate of the parameter using observations taken at locations chosen through BAS.

The error bars are the standard errors of the estimated parameter based on these observations taken at "well chosen locations". The "bar" is the error bar, which extends one standard error above and below the parameter estimate. The horizontal line represents the true value of the parameter in our simulation.

Note that in Figure 4 and Figure 5, the error bars rarely overlap the true value of the parameter. This can be attributed to two factors. In large part, this is due to the fact that they are "error bars" with a length of one standard error beyond the point estimate. Traditional 95% statistical confidence intervals based on two standard errors would in virtually every case overlap the true values. Additionally, these are cumulative plots, in which the same data is used, adding observations to form the parameter estimates as one moves to the right in each figure. Thus the point estimates and error bars are dependent upon one another within a figure.

Finally, we see that the estimates using BAS (to select the points from which to take observation) are generally closer to the truth than when we use random points to take observations, and more importantly the standard errors associated with any given number of observations are much smaller.

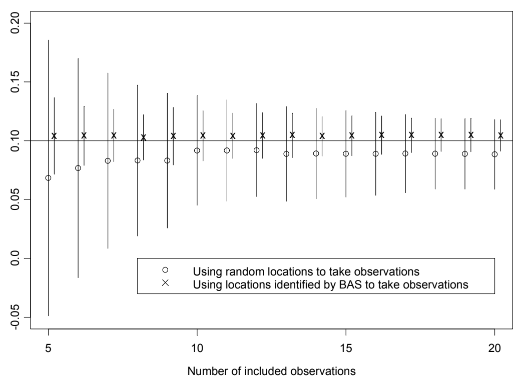


Figure 3: Estimates and error bars for ρ .

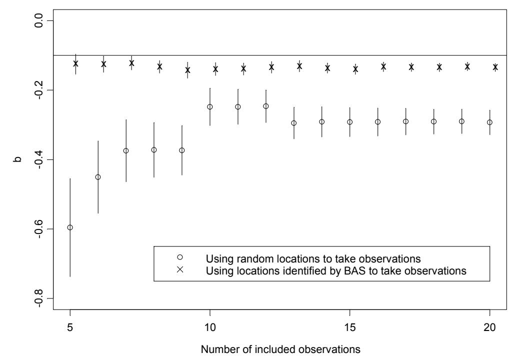


Figure 4: Estimates and error bars for b .

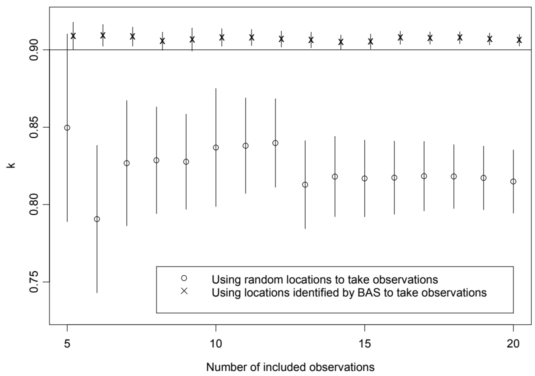


Figure 5: Estimates and error bars for k .

8 CONCLUSION

Our initial results have shown that BAS is highly efficient compared to random sampling. The rate at which the standard errors, or the error bars, are reduced is much quicker, and hence the significant amount of information is found more quickly compared to other traditional methods. We have also shown that this methodology performs well even in the absence of any preliminary data points. Further simulation has shown evidence that BAS can be three times as efficient as random sampling. This efficiency amounts to savings of time and money during actual data collection and analysis.

In addition to the application discussed in this paper, the theoretical framework presented here is generic and can be applied directly to other applications, such as, military, medical, computer vision, and robotics.

Our proposed framework is based on the

multivariate normal distribution. The immediate extensions of this framework will be:

a) To accommodate non-normal parameter estimate distributions. As part of our future study, we intend to employ sampling methodologies using Bayesian Estimation Methods for non-normal parameter estimate distributions. and

b) To use cost effectiveness as an additional variable. In this initial work, the focus was to identify the viewpoints that would give us the most information. However, it is not always feasible or efficient to send the helicopter to this next "best" location. As part of our future work, we intend to identify the next "best efficient" location for the helicopter from which it should collect data.

ACKNOWLEDGEMENTS

This work was done as part of the CAMCOS (Center for Applied Mathematics, Computation, and Statistics) student industrial research program in the Department of Mathematics at San Jose State University.

This work was supported in part by the NASA Ames Research Center under Grant NNA05CV42A. We are appreciative of the help of Kevin Wheeler, Kevin Knuth and Pat Castle, who while at Ames suggested these research ideas and provided background materials.

REFERENCES

Arbel, T., Ferrie, F., "Viewpoint selection by navigation through entropy maps," in Proc. 7th IEEE Int. Conf. on Computer Vision, Greece, 1999, pp. 248–254

MacKay, D., "A clustering technique for digital communications channel equalization using radial basis function networks," Neural Computation, vol.4, no.4, pp. 590–604, 1992.

Nicodemus, F. E., "Reflectance Nomenclature and Directional Reflectance and Emissivity," in Applied Optics, vol. 9, no. 6, June 1970, pp. 1474-1475.

Rahman, H. B., Pinty, B., Verstaete, M. M., "A coupled surface-atmosphere reflectance (CSAR) model. Part 1: Model description and inversion on synthetic data," Journal of Geophysical Research, vol.98, no.4, pp. 20779–20789, 1993.

Sebastiani P., Wynn, H. P., "Maximum entropy sampling and optimal Bayesian experimental design," Journal of Geophysical Research, 62, Part1, pp. 145-157, 2000.

Sebastiani, P., and Wynn, H. P. "Experimental Design to Maximize Information," Bayesian Inference and Maximum Entropy Methods in Science and Engineering: 20th Int. Workshop. AIP Conf. Proc., Gif sur Yvette, France, vol. 568, pp 192-203, 2000.

Vazquez, P. Feixas, P., Sbert, M., M., Heidrich, W., "Viewpoint selection using viewpoint entropy," in Proc. of Vision, Modeling, and Visualization, Germany, 2001, pp. 273–280.

Whaite, P., Ferrie, F. P., "Autonomous exploration: Driven by uncertainty," in Proc. of the Conf. On on Computer Vision and Pattern Recognition, California, 1994, pp. 339–346.

Wheeler, K. R., "Parameterization of the Bidirectional Surface Reflectance," NASA Ames Research Center, MS 259-1, unpublished.

Table 1: Observation and Estimates Using Random Sampling.

Observation Number	θ_v	ϕ_v	r	Estimate (se) of ρ	Estimate (se) of k	Estimate (se) of b
1	0.114	1.673	0.157552			
2	0.882	6.013	0.156616			
3	0.761	0.917	0.192889			
4	0.678	1.308	0.180404			
5	0.260	0.114	0.152558	0.0683 (0.1172)	0.8497 (0.0607)	-0.5958 (0.1413)
6	1.195	2.367	0.146659	0.0767 (0.0932)	0.7906 (0.0476)	-0.4506 (0.1040)
7	0.237	2.805	0.149475	0.0830 (0.0746)	0.8268 (0.0404)	-0.3745 (0.0893)
8	0.166	1.700	0.155497	0.0832 (0.0641)	0.8286 (0.0345)	-0.3722 (0.0788)
9	0.320	2.012	0.154191	0.0831 (0.0572)	0.8277 (0.0307)	-0.3735 (0.0713)
10	1.224	4.085	0.129133	0.0917 (0.0465)	0.8369 (0.0381)	-0.2483 (0.0539)
11	1.409	3.442	0.135005	0.0917 (0.0431)	0.8380 (0.0309)	-0.2481 (0.0503)
12	0.092	1.559	0.154096	0.0920 (0.0394)	0.8398 (0.0285)	-0.2462 (0.0471)
13	0.806	0.891	0.200401	0.0888 (0.0402)	0.8129 (0.0284)	-0.2952 (0.0453)
14	1.256	5.467	0.147654	0.0891 (0.0385)	0.8181 (0.0259)	-0.2914 (0.0433)
15	0.227	1.284	0.155373	0.0889 (0.0368)	0.8169 (0.0248)	-0.2919 (0.0418)
16	1.129	5.522	0.148721	0.0889 (0.0354)	0.8174 (0.0236)	-0.2918 (0.0402)
17	0.507	5.696	0.150381	0.0891 (0.0333)	0.8183 (0.0225)	-0.2904 (0.0380)
18	0.119	4.363	0.142232	0.0890 (0.0302)	0.8181 (0.0207)	-0.2908 (0.0357)
19	0.245	0.524	0.151915	0.0889 (0.0299)	0.8172 (0.0205)	-0.2901 (0.0355)
20	0.446	2.408	0.144471	0.0884 (0.0297)	0.8149 (0.0204)	-0.2930 (0.0354)

Table 2: Observations and Estimates using BAS.

Observation Number	θ_v	ϕ_v	r	Estimate (se) of ρ	Estimate (se) of k	Estimate (se) of b
1	0.460	0.795	0.172364			
2	0.470	0.805	0.177412			
3	1.561	3.957	0.161359			
4	1.561	0.825	0.183571			
5	1.265	3.977	0.129712	0.1041 (0.0325)	0.90904 (0.00879)	-0.1249 (0.0290)
6	0.514	0.845	0.173072	0.1042 (0.0252)	0.90927 (0.00700)	-0.1255 (0.0233)
7	1.561	3.400	0.160130	0.1045 (0.0223)	0.90857 (0.00615)	-0.1220 (0.0199)
8	1.172	4.007	0.130101	0.1029 (0.0192)	0.90547 (0.00577)	-0.1329 (0.0180)
9	0.723	0.875	0.189697	0.1039 (0.0244)	0.90663 (0.00748)	-0.1428 (0.0228)
10	1.561	0.885	0.192543	0.1042 (0.0213)	0.90801 (0.00569)	-0.1394 (0.0185)
11	0.527	0.895	0.172811	0.1042 (0.0193)	0.90796 (0.00523)	-0.1392 (0.0172)
12	1.561	4.047	0.164530	0.1044 (0.0193)	0.90696 (0.00519)	-0.1343 (0.0167)
13	1.561	4.057	0.164822	0.1046 (0.0190)	0.90636 (0.00505)	-0.1314 (0.0161)
14	1.137	4.067	0.131443	0.1038 (0.0169)	0.90483 (0.00471)	-0.1365 (0.0148)
15	0.713	0.935	0.183894	0.1042 (0.0169)	0.90538 (0.00480)	-0.1397 (0.0149)
16	1.561	0.945	0.192280	0.1048 (0.0163)	0.90777 (0.00427)	-0.1333 (0.0136)
17	1.187	4.097	0.134701	0.1047 (0.0146)	0.90757 (0.00399)	-0.1340 (0.0125)
18	0.655	0.965	0.176841	0.1048 (0.0140)	0.90779 (0.00385)	-0.1349 (0.0120)
19	1.561	4.117	0.168819	0.1049 (0.0142)	0.90694 (0.00388)	-0.1321 (0.0120)
20	1.148	4.127	0.132199	0.1045 (0.0132)	0.90617 (0.00373)	-0.1349 (0.0114)

DYNAMIC SENSOR NETWORKS: AN APPROACH TO OPTIMAL DYNAMIC FIELD COVERAGE

Simone Gabriele and Paolo Di Giamberardino
Dipartimento di Informatica e Sistemistica "Antonio Ruberti"
Università degli Studi di Roma "La Sapienza"
Via Eudossiana 18, Roma, Italia
gabriele@dis.uniroma1.it, paolo.digiamberardino@uniroma1.it

Keywords: Sensor network, dynamic configuration, optimal motion.

Abstract: In the paper a solution to the sensor network coverage problem is proposed, based on the usage of moving sensors that allow a larger fields coverage using a smaller number of devices. The problem than moves from the optimal allocation of fixed or almost fixed sensors to the determination of optimal trajectories for moving sensors. In the paper a suboptimal solution obtained from the sampled optimal problem is given. First, in order to put in evidence the formulation and the solution approach to the optimization problem, a single moving sensor has been addressed. Then, the results for multisensor systems are shown. Some simulation results are also reported to show the behavior of the sensors network.

1 INTRODUCTION

Distributed sensors systems and networks are growing relevance in the scientific and engineering community. Their introduction into several applications for monitoring or surveillance, like for example temperature, ground humidity and solar radiation in farms or parks, presence and distribution of people in critical structures, temperature for fire prevention (buildings as well as woods), and so on, together with the growth of decentralized control in large and complex structures (factories, refineries, energy production and distribution, etc) makes the interest of many researchers for these kind of problems growing and growing, as proved for example by (Akyildiz et al., 2002; Lewis, 2004).

The use of several sensors, suitably deployed, makes the range of measurements as wide as required. Then, one common features required by sensor networks is the full coverage of a given (large) area with the union of each single field of measurement. This problem has been usually faced studying optimal, suboptimal or heuristic solutions to the coverage problem in terms of *good* allocation of sensors in the area under measurement. In other terms, the problem usually has been posed answering the question ""which are the *best* places to put the N sensors?", where *best* is often considered with respect to energetic costs (for the deployment as well as for the communications) or number of sensors.

Such a problem has been well studied in a lot of works, such as (V. Isler and Daniilidis, 2004; Yung-Tsung Hou, 2006; Zou and Chakrabarty, 2004; Lin, 2005; Huang and Tseng, 2005; Meguerdichian et al., 2001).

In (Tan et al., 2004; Howard, 2002) the problem of self-deploying mobile sensors, able to configure according to the environment, is addressed and some solutions are proposed. In these kind of approaches a common fact is the use of a lot of quasi static sensor units to cover a given area.

An alternative idea is to use a reduced number of sensor units moving continuously; such an approach is the one followed by the authors in the present work. A result based on the solution of a suitable coordinated optimal control problem is presented in the sequel. The only limit of this approach is the impossibility of getting a continuous measure for a given point of the area under monitoring, allowing the user only to fix the maximum acceptable time between two consecutive measures of the same point. The problem is than to plan trajectories optimally in sense of area coverage. An optimal control formulation for this problem is proposed in (Wang, 2003). In (Tsai, 2004;

Cecil and Marthler, 2006) the same problem has been studied in the level set framework and some suboptimal solutions are proposed. An approach based on space decomposition and Voronoy graphs is proposed in (Acar et al., 2006).

In the present work we suggest a customizable optimal control framework that allow the study of a set of different case of the described problem.

The motion problem both for a single sensor and for a set of sensors, under kinematic and dynamic constraints on the motion, with the objective to maximize the area covered during the movement is formulated as an optimal control problem Since this problem, in the general case, cannot be solved analytically, a discretization of space and time is performed, so obtaining a discrete time optimal control problem tractable as a Non Linear Programming (NLP) one. Similar approach to optimal control problems was proposed for industrial manipulators control in (Bicchi and Tonietti, 2004) and for path planning in (Ma and Miller, 2006).

The paper is organized as follows. In Section 2 the mathematical model of the sensor(s) is given, together with the constraints to be satisfied. Model and constraints are then used to propose a formulation for the optimal control problem. In Section 3 the discrete problem, obtained by spatial and temporal discretization, is formulated in terms of a solvable NLP problem. Section 4 is devoted to the particularization of the problem for some cases, showing the respective simulation results. Some final comments in Section 5 end the paper.

2 PROBLEM FORMULATION

2.1 The Mathematical Model

A mobile sensor is modeled, from the dynamic point of view, as a material point of unitary mass, moving on a space $W \subset \mathbb{R}^2$, called the *workspace*, under the action of two independent control input forces named $u_1(t)$ and $u_2(t)$. Then, the position of the sensor in W at time t is described by its Cartesian coordinates $(x_1(t), x_2(t))$. The motion satisfies the well known equations:

$$\begin{aligned} \ddot{x}_1(t) &= u_1(t) \\ \ddot{x}_2(t) &= u_2(t) \end{aligned} \qquad (1)$$

The linearity of 1 allows one to write the dynamics in the form

$$\begin{aligned} \dot{z}(t) &= Az(t) + Bu(t) \\ y(t) &= Cz(t) \end{aligned} \qquad (2)$$

where

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad B = \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}$$

$$C = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

once the state vector $z(t) = \left(\dot{x}_1(t), x_1(t), \dot{x}_2(t), x_2(t) \right)^T$ and the output $y(t) = \left(x_1(t), x_2(t) \right)^T$ are defined. Clearly, $y(t)$ denotes the trajectory followed by the mobile sensor.

If the workspace M is supposed to be a rectangular subset of $\mathbb{R}^2$, the trajectory must satisfy the constraints

$$\begin{aligned} x_{1,min} \le x_1(t) \le x_{1,max} \\ x_{2,min} \le x_2(t) \le x_{2,max} \end{aligned}$$

Moreover, physical limits on the actuators (for the motion) and/or on the sensors (in terms of velocity in the measure acquisition) suggest the introduction of the following additional constraints

$$\begin{aligned} |\dot{x}_1(t)| &\le v_{max} \\ |\dot{x}_2(t)| &\le v_{max} \\ |u_1(t)| &\le u_{max} \\ |u_2(t)| &\le u_{max} \end{aligned}$$

In this work the hypothesis that the mobile sensor at time t can take measures within a circular area of radius ρ around its current position $y(t)$ is considered. Such an area under sensor *visibility* will be denoted as

$$M(t) = \sigma(y(t), \rho)$$

In other words, $M(t)$ denotes the area over which the sensor can take measures at time t.

2.2 The Mathematical Formulation of the Coverage Problem

According to what stated in subsection 2.1, given a time interval $\Theta = [0, t_f]$, the geometric expression of the area covered by the measures during Θ, say M_Θ, can be easily given by

$$M_\Theta = \bigcup_{t \in \Theta} M(t) = \bigcup_{t \in \Theta} \sigma(y(t), \rho) \qquad (3)$$

However, such a formulation is not easy to be used in an analytical optimal control problem formulation. Then, alternative expressions that gives in an analytic form how a given trajectory reflects on the space coverage for the sensor measure must be found.

The one used in this work is based on the distance $d(y(t),P)$ between the points $\{P|P \in W\}$ of the workspace and the trajectory.

Once the distance between a point P of the workspace and a given trajectory $y(t)$ is defined as

$$d(y(t),P) = \min_{t\in\Theta} ||y(t) - P|| \tag{4}$$

and making use of the function

$$\text{pos}(\xi) = \begin{cases} \xi & \text{if} \quad \xi > 0 \\ 0 & \text{if} \quad \xi \leq 0 \end{cases} \tag{5}$$

that fixes to zero any nonpositive value, the function

$$\hat{d}(y(t),P,\rho) = \text{pos}\left(d(y(t),P) - \rho\right) \geq 0$$

can be defined. Then, a measure of how the trajectory $y(t)$ produces a good coverage of the workspace can be given by

$$J(y(t)) = \int_{P\in W} \hat{d}(y(t),P,\rho) \tag{6}$$

Smaller is $J(y(t))$, better is the coverage. If $J(y(t)) = 0$ than $y(t)$ covers completely the workspace.

2.3 The Optimal Control Problem Formulation

Making use of the element introduced in previous subsections, the Optimal Control Problem can be formulated in order to find the best trajectory $y^*(t)$ that maximizes the area covered by sensor measurement during the time interval Θ, as defined in previous subsection, and satisfies the constraints. Then a constrained optimal control problem is obtained, whose form is

$$\begin{aligned} &\min J(\Lambda(u(t))) \\ &f(u(t)) = 0 \\ &g(u(t)) \leq 0 \end{aligned} \tag{7}$$

In (7), the cost functional $J(\cdot)$ is given by (from (6))

$$J(\Lambda(u(t))) = \int_{p\in W} \hat{d}(\Lambda(u(t)),p,\rho) \tag{8}$$

The optimal solution $u(t) = u^*(t)$ $(t \in \Theta)$is the control that produces the optimal trajectory $y^*(t) = \Lambda(u^*(t))$ $(t \in \Theta)$.

In general is not possible to solve analytically the optimal control problem defined in the precedent section, due the functional form of $J(\cdot)$ in (7). In next section a solvable discrete problem is defined.

3 DISCRETE TIME FORMULATION

In order to overcome the difficulty of solving a problem as (7) due to the complexity of the cost function $J(\cdot)$, a discretization is performed, both with respect to space W, and with respect to time in all the time dependent expressions.

The workspace is divided into square cells $c_{i,j}$ with resolution (size) l_{res}, and the trajectory is discretized with sample time T_s. The equations of the discrete time dynamics are:

$$\begin{aligned} &z((k+1)T_s) = A_d z(kT_s) + B_d u(kT_s) \\ &y(kT_s) = Cz(kT_s) \end{aligned} \tag{9}$$

where $A_d = e^{AT_s}$ and $B_d = \int_0^{T_s} e^{A\tau} B d\tau$

The state vector $z(t)$ at the generic time instant $t = NT_s$ depends on the initial state z_0 and on the controls from time $t = 0$ to time $t = (N-1)T_s$

$$z(NT_s) = A_d^N z_0 + \sum_{i=0}^{N-1} A_d^i B_d u((N-1)T_s - iT_s) \tag{10}$$

The following matrices are now defined:

$$Z_N = \begin{bmatrix} z(T_s) \\ \vdots \\ z(NT_s) \end{bmatrix}$$

$$Y_N = \begin{bmatrix} y(0) \\ \vdots \\ y(NT_s) \end{bmatrix} \qquad A_N = \begin{bmatrix} A_d \\ A_d^2 \\ \vdots \\ A_d^N \end{bmatrix}$$

$$B_N = \begin{bmatrix} B_d & 0 & \dots & 0 & 0 \\ A_d B_d & B_d & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ A_d^{N-1} B_d & A_d^{N-2} B_d & \dots & A_d B_d & B_d \end{bmatrix}$$

$$U_N = \begin{bmatrix} u(0) \\ \vdots \\ u((N-1)T_s) \end{bmatrix} \qquad U_{max} = \begin{bmatrix} u_{max} \\ \vdots \\ u_{max} \end{bmatrix}$$

$$Z_{max} = \begin{bmatrix} v_{max} \\ x_{1,max} \\ v_{max} \\ x_{2,max} \\ -- \\ \vdots \\ -- \\ v_{max} \\ x_{1,max} \\ v_{max} \\ x_{2,max} \end{bmatrix} \qquad Z_{min} = \begin{bmatrix} -v_{max} \\ x_{1,min} \\ -v_{max} \\ x_{2,min} \\ -- \\ \vdots \\ -- \\ -v_{max} \\ x_{1,min} \\ -v_{max} \\ x_{2,min} \end{bmatrix}$$

Making use of such matrices, the sequence of values for the sampled state vector $z(kT_s)$, with $0 \leq K \leq (N+1)$, can be expressed in the simple compact form

$$Z_N = A_N z_0 + B_N U_N \tag{11}$$

The cost function can then be written as:

$$J(Y_N) = \sum_{i=1}^{\nu_x} \sum_{j=1}^{\nu_y} \hat{d}(\Lambda(U_N), c_{i,j}, \rho) \tag{12}$$

where $\nu_x = \frac{(x_{max} - x_{min})}{l_{res}}$, $\nu_y = \frac{(y_{max} - y_{min})}{l_{res}}$ and $\Lambda(U_N) = Y_N$.

3.1 The Nonlinear Programming Problem Formulation

The problem of finding the maximum area coverage trajectory can now be written as a tractable discrete optimization problem with linear inequality and box constraints

$$\begin{aligned} &\min_{U_N} \sum_{i=1}^{\nu_x} \sum_{j=1}^{\nu_y} \hat{d}(\Lambda(U_N), c_{i,j}, \rho) \\ &A_{model} U_N \leq B_{model} \\ &-U_{max} \leq U_N \leq U_{max} \end{aligned} \tag{13}$$

where $A_{model} = \begin{bmatrix} B_N \\ -B_N \end{bmatrix}$

and $B_{model} = \begin{bmatrix} Z_{max} - A_N z_0 \\ -Z_{min} + A_N z_0 \end{bmatrix}$

Suboptimal solutions can be computed using numerical methods. In the simulations performed, the SQP (Sequential Quadratic Programming) method has been applied. The obtained model can be customized according to the specific task, as shown in the following section.

4 MODEL CUSTOMIZATION AND CASE SOLUTIONS

In this section some cases are faced in order to put in evidence the capabilities and he effectiveness of the proposed solution. The values of parameters used in all the simulations are:

$u_{max} = 0.5N$, $v_{max} = 1.5\frac{m}{sec}$, $x_{max} = y_{max} = 4m$,$x_{min} = y_{min} = -4m$, $T_s = 0.5sec$

4.1 The Case of a Single Sensor

Fixed initial state

The formulation adopted allows to find covering trajectories for a single sensor who start from a given initial state. In figure 1 the corresponding simulation results, for $z_0 = \begin{bmatrix} 0 & 0 & 0 & 0. \end{bmatrix}^T$ are depicted. With $t_f = 20sec$ the sensor covers the 70.9% of total area.

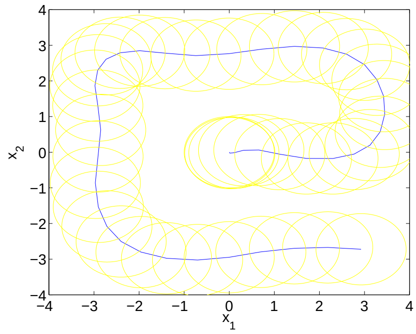


Figure 1: Suboptimal trajectory for one moving sensor with arbitrary starting point ($z = 0$).

Optimal initial state

The initial state z_0 can be included among the set of variables of the optimization problem. In fact, defining

$$V_N = \begin{bmatrix} z(0) \\ u(0) \\ \vdots \\ u((N-1)T_s)) \end{bmatrix}$$

$$H_N = \begin{bmatrix} A_d & B_d & \dots & 0 & 0 \\ A_d^2 & A_d B_d & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ A_d^N & A_d^{N-1} B_d & \dots & A_d B_d & B_d \end{bmatrix}$$

it is possible to write

$$Z_N = H_N V_N \tag{14}$$

The new optimization problem is then obtained setting

$$A_{model} = \begin{bmatrix} H_N \\ -H_N \end{bmatrix} \quad \text{and} \quad B_{model} = \begin{bmatrix} Z_{max} \\ -Z_{min} \end{bmatrix}$$

Leaving the initial condition free, better results are obtained since the initial state is also optimal, as it is shown in figure 2.

Here in the same time of the precedent simulation ($t_f = 20sec$), the sensor covers 73.49% of total area, versus the 70.9% of the fixed starting state case.

Periodic trajectory

Cyclic trajectories can be very useful in area monitoring or surveillance tasks because this choice, once

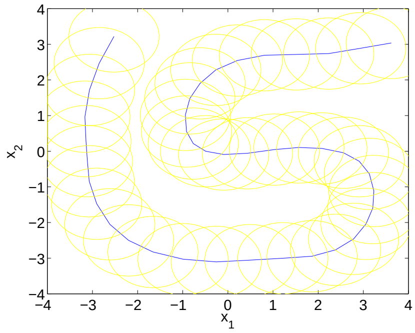


Figure 2: Suboptimal trajectory for one moving sensor with suboptimal starting point.

the maximum time NT_S between measures on the same point is fixed, allows to repeat the measure in the same point periodically.

According to the present formulation, the sampled dynamics over N sampled instants has a periodic behavior if and only if

$$z((N+1)T_s) = z(0) \tag{15}$$

Observing that the computation of the $(N+1)-th$ sampled values for the state gives

$$z((N+1)T_s) = [\begin{array}{cccc} A_d^{N+1} & \dots & A_d B_d & B_d \end{array}] V_N$$

while

$$z(0) = [\begin{array}{ccccc} I & 0 & \dots & 0 & 0 \end{array}] V_N$$

condition (15) can be rewritten as

$$[\begin{array}{cccc} A_d^{N+1} - I & \dots & A_d B_d & B_d \end{array}] V_N = 0 \tag{16}$$

Equation (16) must be added as a new constraint in the optimization problem in order to get periodic solutions.

The figure 3 shows the results obtained by simulations for this case.

With $t_f = 40sec$ the $98,17\%$ of the workspace area is covered.

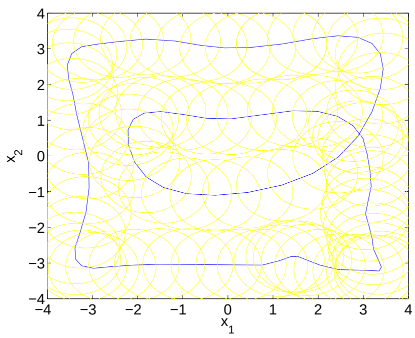


Figure 3: Sub-optimal trajectory for one moving sensor.

4.2 The Case of Multiple Sensors

The models shown above are very easily extended to the case under interest of area coverage with multiple moving sensors. The use of multiple sensors instead of one allows to reduce the time t_f within the same coverage or, equivalently, increase the coverage for the same t_f.

If n is the number of the moving sensors, the optimization problem can be formulated in the same way once the following matrices are defined and used instead of the corresponding ones:

$$U_N^n = \left[\begin{array}{c} U_{N,1} \\ \vdots \\ U_{N,n} \end{array}\right]$$

where $U_{N,i}$ stands for the control set (U_N) of the $i-th$ sensor.

$$A_{model}^n = \left(\begin{array}{cccc} A_{model,1} & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & A_{model,n} \end{array}\right)$$

$$B_{model}^n = \left(\begin{array}{c} B_{model,1} \\ \vdots \\ B_{model,n} \end{array}\right)$$

where $A_{model,i}$ and $B_{model,i}$ are the A_{model} and the B_{model} matrices of the $i-th$ single sensor model.

In figure 4 the result for the multi-sensor case with $n = 2$ is depicted. In time $t_f = 25sec$ the 99.86% of the workspace area is covered. The gain of time respect to the single sensor case is evident.

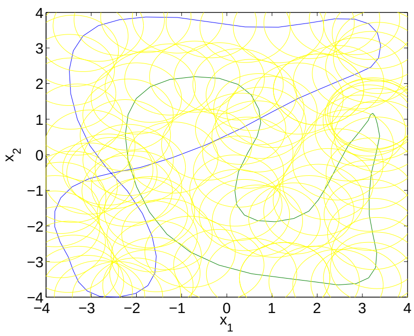


Figure 4: Sub-optimal cyclic trajectories for moving sensor 1 (blue) and moving sensor 2 (green), the yellow circles show the measures area.

5 CONCLUSIONS AND FUTURE WORKS

In the present paper a measurement system composed by several sensors moving within the area under measure has been considered. This system has been called *dynamic sensor network*. For this kind of system the formulation for an optimal solution to the area coverage problem has been provided. The complexity of the cost function makes very hard (actually impossible) the computation of the optimal solution. Then, in order to get a solution, a sampled model has been considered, bringing to a nonlinear programming problem that has been solved numerically. The results for a single sensor with different choices for initial conditions (freely given or optimal) and for behavior of the trajectory (non periodic or periodic) show the effectiveness of the proposed procedure. The case of a n-sensors systems has also been considered and, for $n = 2$ has been simulated in order to show the results when more sensors are present. The problem at present under investigation concerns the inclusion of non collision constraints, where non collisions are to be considered both between moving sensors and with fixed obstacles that can be present in the measurement area.

REFERENCES

Acar, E., Choset, H., and Lee, J. Y. (2006). Sensor-based coverage with extended range detectors. *Robotics, IEEE Transactions on [see also Robotics and Automation, IEEE Transactions on]*, 22(1):189–198.

Akyildiz, I., Su, W., Sankarasubramaniam, Y., and Cayirci, E. (2002). A survey on sensor networks. *Communications Magazine, IEEE*, 40(8):102–114.

Bicchi, A. and Tonietti, G. (2004). Fast and ”soft-arm” tactics [robot arm design]. *Robotics & Automation Magazine, IEEE*, 11(2):22–33.

Cecil and Marthler (2006). A variational approach to path planning in three dimensions using level set methods. *Journal of Computational Physics*, 221:179–197.

Howard, Mataric, S. (2002). An incremental self-deployment for mobile sensor networks. *Autonomus Robots*.

Huang, C.-F. and Tseng, Y.-C. (2005). The coverage problem in a wireless sensor network. *Mobile Networks and Applications*, 10:519–528.

Lewis, F. L. (2004). *Smart Environments: Technologies, Protocols and Applications*, chapter 2. D.J. Cook and S. K. Das.

Lin, F.Y.S. Chiu, P. (2005). A near-optimal sensor placement algorithm to achieve complete coverage-discrimination in sensor networks. *Communications Letters, IEEE*, 9:43–45.

Ma, C. and Miller, R. (2006). Milp optimal path planning for real-time applications. In *American Control Conference, 2006*, page 6pp.

Meguerdichian, S., Koushanfar, F., Potkonjak, M., and Srivastava, M. (2001). Coverage problems in wireless adhoc sensor networks. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1380–1387vol.3.

Tan, J., Lozano, O., Xi, N., and Sheng, W. (2004). Multiple vehicle systems for sensor network area coverage. In *Intelligent Control and Automation, 2004. WCICA 2004. Fifth World Congress on*, volume 5, pages 4666–4670Vol.5.

Tsai, Cheng, O. B.-S. (2004). Visibility and its dynamics in a pde based implicit framework. *Journal of Computational Physics*, 199:260–290.

V. Isler, S. K. and Daniilidis, K. (2004). Sampling based sensor-network deployment. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems IROS*.

Wang, P. K. C. (2003). Optimal path planing based on visibility. *Journal of Optimization Theory and Applications*, 117:157–181.

Yung-Tsung Hou, Tzu-Chen Lee, C.-M. C. B. J. (2006). Node placement for optimal coverage in sensor networks. In *Proceedings of IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*.

Zou, Y. and Chakrabarty, K. (2004). Uncertainty-aware and coverage-oriented deployment for sensor networks. *Journal of Parallel and Distributed Computing*, 64:788–798.

BEHAVIOR ANALYSIS OF PASSENGER'S POSTURE AND EVALUATION OF COMFORT CONCERNING OMNI-DIRECTIONAL DRIVING OF WHEELCHAIR

Yuta Sato, Yoshiyuki Noda, Takanori Miyoshi and Kazuhiko Terashima
Department of Production Systems Engineering, Toyohashi University of Technology
Hibarigaoka 1-1, Tempaku, Toyohashi, 441-8580, Japan
{*sato,noda,miyoshi,terasima*}*@syscon.pse.tut.ac.jp*

Keywords: Wheelchair, Omni-directional drive, Passenger's posture behavior, Comfort sensation, Acceleration sensor.

Abstract: The purpose of this study is to analyze the relationship between passenger's posture behavior and comfort while riding omni-directional wheelchair. First, an algorithm to transform the obtained data in the sensor coordinates using acceleration sensor into the vehicle coordinates by means of proposed correction algorithm. Its effectiveness is demonstrated by experiments. Second, analysis on the relationship between acceleration of wheelchair movement, passenger's posture behavior and comfort sensation in the riding motion to forward, backward and lateral direction is studied. Posture behavior of passenger's head and chest is measured by acceleration sensors, and comfort sensation of passenger is evaluated by applying the Semantic Differential (SD) method and a Paired Comparison Test. Finally, through a lot of experiment, influence factors concerning comfort while riding to wheelchair are discussed.

1 INTRODUCTION

In today's aged society, a wheelchair is the most common vehicle to assist elderly and handicapped people. Wheelchairs can provide many profits to users, such as extending mobility, broadening community and social activities, and enhancing quality of life (QOL) of the users. Therefore, the user of electric powered wheelchair generally increases, and the development of wheelchair which is able to drive comfortably is highly required.

Various factors are largely related to the riding comfort of electric powered wheelchairs, such as seat comfort, ambient noise, and stability. The passenger's posture swing of body and the driving acceleration and deceleration are generally the main factors which influence on comfort.

In the international standard of ISO-2631-1, the riding comfort of a transportation vehicle is evaluated by the magnitude of the acceleration weighted by oscillation frequency (ISO-2631-1, 1997). Train's driving was often evaluated by this standard method and improved by suppressing the acceleration of uncomfortable frequency (C.H.Lee et al., 2005).

Passenger's comfort while riding wheelchair is also improved by suppressing the vibration with discomfort frequencies. Maeda described a wheelchair with passenger has three resonant frequencies; the first resonant frequency is 5$\sim$7[Hz], the second is 8[Hz], and third is 13$\sim$15[Hz]. And he addressed that the main point for improving a wheelchair passenger's comfort was to reduce the seat vibration of wheelchair at around 8[Hz] (S.Maeda et al., 2003).

The result described above is concerned with the vibration while driving over long time. On the other hand, a wheelchair is not the only steady-state operation, but also the transient state such as starting and stopping. Additionally, the high drive acceleration or deceleration causes passenger's discomfort. Yamagishi improved comfort while riding a car by reduction of acceleration and jerk (derivative of acceleration) (Y.Yamagishi and H.Inooka, 2005). However, up to the present, passenger's posture behavior of body which causes discomfort during riding has not been studied to own knowledge.

In the author's laboratory, comfort driving for wheelchair has been one of the main research subjects. Omni-directional Wheelchair (OMW) which can drive towards omni-direction is developed, and has a power assist system for helping fragile or elderly attendants (K.Terashima et al., 2006).

Passenger's comfort has been improved by suppressing the both of OMW and organ's vibration of passengers. Passenger's vibration can be estimated by the proposed two-dimensional passenger model. To suppress its vibration, the control system with two notch filters has been given. According to the simulation results, passenger's vibration is suppressed almost completely by the proposed controller in the case of forward and backward (H.Kitagawa et al., 2002), (J.Urbano et al., 2005). However, in the case of omni-direction such as lateral movements, it is only verified by simulation, not experiments. Therefore, it must be verified by experiments with measuring passenger's vibration. If lateral motion gives large discomfort, OMW is not appropriate as wheelchair for human being. It is necessary to investigate the posture behavior and comfort for the movements to any direction, whether or not OMW can be applied as a vehicle to carry people.

In most study about comfort driving, passenger's body posture is moved. However, passenger's posture behaviors while riding the wheelchair are not measured explicitly in actual experiments. The authors predict that passenger's behavior while riding is fairly related with the passenger's discomfort sensation.

The purpose of this study is therefore to analyze the relationship between passenger's posture behavior and comfort while driving to the omni-direction. First, an algorithm to transform the obtained data in the sensor coordinates using acceleration sensor into the vehicle coordinates by the correction algorithm. Its effectiveness is demonstrated by experiments. Second, analysis on the relationship between acceleration of wheelchair movement, passenger's posture behavior and comfort sensation in the riding motion to the forward, backward and lateral direction is studied. Passenger's posture behavior is measured by acceleration sensors fixed at head and chest. Comfort sensation of passenger is evaluated by applying the Semantic Differential (SD) method. Thirdly, experimental analysis on the chest movement with comfort is done. Passenger's sensation is evaluated by a Paired Comparison Test.

Finally, through a lot of experiment, influence factors concerning comfort while riding to wheelchair are discussed.

2 EXPERIMENTAL SETUP

2.1 Experimental Wheelchair

To clearly analyze the relation between passenger's body behavior and comfort sensation while riding the wheelchair, high performance wheelchair "Emu-S" (Wakogiken Co., Ltd.) which can drive with high velocity and acceleration as shown in Figure 1 is used in experiments.

Figure 1: Wheelchair used in experiments.

This wheelchair was introduced for another's study, and also used for observing passenger's movement (S.Shimada et al., 2002). To observe the passenger's body behavior for omni-directional movement, wheelchair seat is set with 90 [deg] rotations as shown in Figure 2.

Figure 2: Seat allocation of a wheelchair in the case of observing a lateral motion.

Table 1 is a specification of this wheelchair. It has been made for a wheelchair football needed to move fast, and therefore it can drive with high velocity and high acceleration. The max velocity and acceleration are respectively 2.7[m/s] and 3.5[m/s^2]. This wheelchair's specification is enough from the viewpoint of practical wheelchair's use. However, it can largely induce passenger's posture movements and discomfort sensation while riding the wheelchair, and thus it is used to analyze in this experiments.

This wheelchair is driven by the reference signal of analog voltage -5 to +5[V]. And DSP is loaded for motor servo control and digital signal processing of brushless resolver signal.

2.2 Measurement of Passenger's Behavior

Acceleration sensor of ACA302 (Star Micronics Co., Ltd.) is used for measuring the passenger's behavior. This sensor can detect three-axes acceleration of X, Y and Z-axis, and the range of detection is $\pm$19.6[m/s^2]. Acceleration sensors are put at the passenger's head,

Table 1: Specification of wheelchair.

Size	800×630×900 [mm]
Weight	88 [kg]
Motor	AC Servo motor 232 [W] ×2
Rated torque	1.18 [N·m]
Max torque	4.5 [N·m]
Rated rotation	1880 [rpm]
Max rotation	3000 [rpm]
Location detect	Brushless resolver

chest and the wheelchair as shown in Figure 3 and Figure 4. Sensor signals can be amplified by circuit, because the signal voltage is very small ([mV] order). Acceleration data is obtained from AD board at the sampling times of 10[ms]. Reference of analog voltage for driving the wheelchair is provided from DA board.

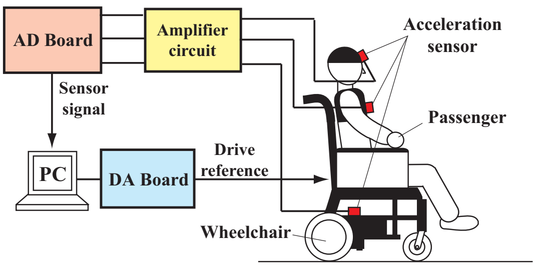


Figure 3: System for measuring passenger's behavior.

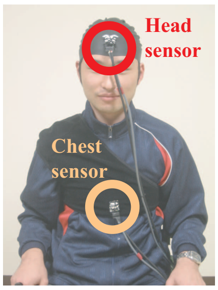


Figure 4: Acceleration sensor for measuring the motion of head and chest's point.

3 DATA CORRECTION OF ACCELERATION SENSOR

3.1 Correction Algorithm

It seems extremely difficult in the usual experiments that acceleration sensors must be horizontally placed at the exact accuracy. Therefore, the acceleration data obtained in the sensor coordinates by experiments should be converted in the drive coordinates by a correction algorithm. Acceleration sensor coordinates are x, y and z, and drive coordinates are u, v and w as shown in Figure 5. This study deals the drive toward one direction of forward, backward, rightward or leftward at once, and then the corrected acceleration must be appeared only in the drive direction acceleration of u-axis by the proposed algorithm.

With the proposed correction algorithm, acceleration sensor can be placed at any point of passenger's body part without considering sensor mount angle.

First, sensor coordinates are rotated through an angle ϕ around x-axis. Next, its coordinates are rotated through an angle θ around y-axis. With these rotation, z-axis is matched with w-axis. The coordinates after rotation are x', y' and z' as shown in Figure 6. Accelerations after the rotation represented as $a_{x'}$, $a_{y'}$ and $a_{z'}$, are calculated by Eq.(1), where a_x, a_y and a_z are the accelerations before the rotation. Notation of S and C denotes sinusoidal and cosine function respectively.

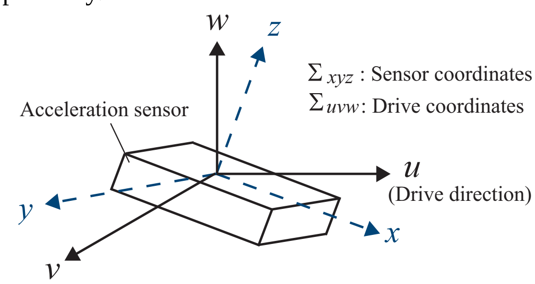


Figure 5: Sensor and drive coordinates.

$$\begin{pmatrix} a_{x'} \\ a_{y'} \\ a_{z'} \end{pmatrix} = \begin{pmatrix} C\theta & S\theta S\phi & -S\theta C\phi \\ 0 & C\phi & S\phi \\ S\theta & -C\theta S\phi & C\theta C\phi \end{pmatrix} \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} \quad (1)$$

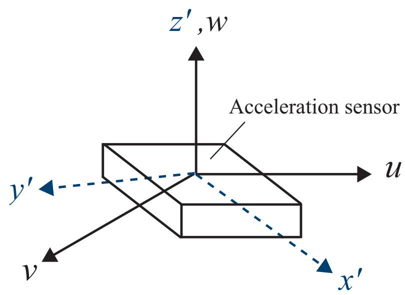


Figure 6: Coordinates after rotation through angles ϕ and θ.

The values of $a_{x'}$, $a_{y'}$ and $a_{z'}$ at initial state are described by Eq.(2). Acceleration is appeared only in z' axis, and gravity acceleration is g =9.8[m/s^2].

$$\begin{pmatrix} a_{x'} \\ a_{y'} \\ a_{z'} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ g \end{pmatrix} \quad (2)$$

By using the second row equation on Eq.(1), ϕ is calculated as Eq.(3), where $\bar{a_y}$ and $\bar{a_z}$ are the average of a_y and a_z at initial state, respectively.

$$\phi = \tan^{-1}\left(-\frac{\bar{a_y}}{\bar{a_z}}\right) \quad (3)$$

Then, θ is given in Eq.(4). ϕ is obtained by previous calculation, and $\bar{a_x}$, $\bar{a_y}$ and $\bar{a_z}$ are the average of acceleration at initial state.

$$\bar{a_x}\cos\theta + \bar{a_y}\sin\theta\sin\phi - \bar{a_z}\sin\theta\cos\phi = 0 \quad (4)$$

At the last, sensor coordinates are rotated through an angle ψ around z' axis. With this rotation, x' axis is matched with a drive direction of u axis. The relation between(a_u,a_v,a_w) and ($a_{x'}$,$a_{y'}$,$a_{z'}$) is expressed by the following equation.

$$\begin{pmatrix} a_u \\ a_v \\ a_w \end{pmatrix} = \begin{pmatrix} C\psi & S\psi & 0 \\ -S\psi & C\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} a_{x'} \\ a_{y'} \\ a_{z'} \end{pmatrix} \quad (5)$$

The rotation angle ψ can be defined as Eq.(6). The passenger and wheelchair are transferred towards one direction of forward, backward or lateral at once. Then, the acceleration a_v of v axis expressed by Eq.(7) which is the direction perpendicular to u-axis, is small. Therefore, ψ is chosen so as to minimize the following cost function J.

$$J = \min_{\psi} \int_0^T a_v(\psi,t)^2 dt \quad (6)$$

$$a_v(\psi,t) = -a_{x'}(t)\sin\psi + a_{y'}(t)\cos\psi \quad (7)$$

In off-line process, three rotation angles of ϕ, θ and ψ can be estimated automatically, and made the sensor coordinates transformed in order to correct the installation error.

3.2 Verification of Proposed Algorithm

Figure 7 indicates that acceleration sensor signals of a_x, a_y and a_z are converted to the data signal of a_u, a_v and a_w, and the corrected data has only in the drive direction of u-axis by the proposed algorithm described in the previous section. Through these results, the proposed algorithm can provide the exact value of acceleration using acceleration sensor.

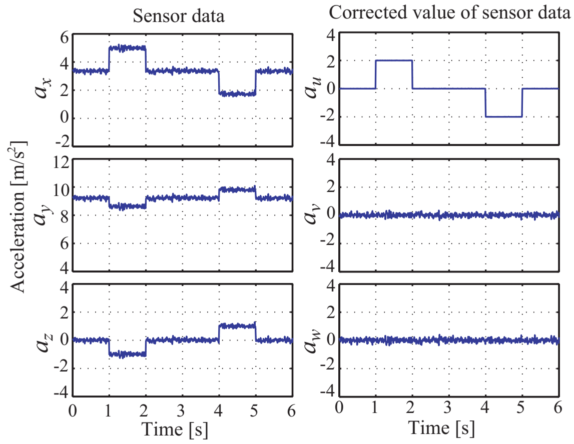


Figure 7: Result of the corrected values of sensor data by the proposed algorithm.

4 RELATIONSHIP BETWEEN PASSENGER'S POSTURE BEHAVIOR AND COMFORT

4.1 Passenger's Posture Behavior and Comfort in Omni-directional Drive

In order to discuss the relationship between passenger's posture behavior of body and comfort feeling, a lot of experiments are executed as follows.

4.1.1 Experimental Description

At experiments, the wheelchair is driven by three patterns with various acceleration of 0.5, 1.0 and 2.0[m/s^2]. These patterns are the trapezoidal velocity one with maximum velocity of 1[m/s] and movement distance of 2[m]. And it is respectively driven towards four-direction of forward, backward, rightward and leftward. The group of healthy and standard proportions comprised of 6 people (average: 170[cm], 60[kg]) is tested. All of them have never ridden on this wheelchair before.

Furthermore, the wheelchair is driven particularly in patterns with various acceleration of 1.0, 1.3, 1.7, 2.0 [m/s^2] with the same maximum velocity of 1[m/s] and distance of 2[m]. Drive directions are forward direction where 3 passengers are tested.

The passengers were given the information about the movement distance and the direction in experiments before start, and given a start sign when the wheelchair's movement starts. SD method is used for the investigation of passenger's comfort.

4.1.2 Experiment in Forward-backward Direction Drive

Passenger's posture acceleration of *u*-axis for the forward drive experiment with acceleration of 2[m/s^2] is shown in Figure 8. Numbers in the above side of Figure 8 corresponds to the passenger's posture shown in the bottom side.

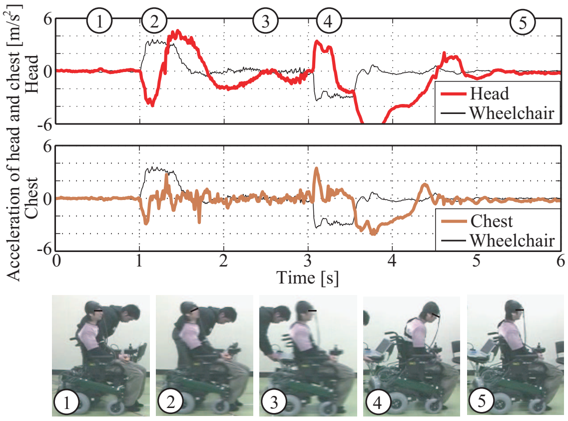


Figure 8: Acceleration of head and chest in the case of forward transfer with acceleration of 2.0 [m/s^2].

The head and chest acceleration can be obtained by subtracting the acceleration of the wheelchair from the measured acceleration of the head and chest.

When the forward transfer is started, a head moves in the opposite direction and a chest doesn't move due to the backrest as shown at the number 2 of Figure 8. During constant velocity drive, the head movement is suppressed by passenger's adjustment in number 3. The deceleration triggered a head swing. Then, a chest swing also appears, because there is not barrier of preventing a chest movement like the backrest in number 4. When a deceleration time is ended, the passenger intentionally suppresses the body movement, and there is no residual vibration at the end of driving in number 5.

The passenger's behavior in forward transfer of 0.5 and 1.0 [m/s^2] is shown in Figure 9. The head acceleration becomes bigger with increasing the drive acceleration. The average values of 6 passenger's maximum head acceleration in the driving acceleration of 0.5, 1.0 and 2.0 [m/s^2] are respectively 1.95 (standard variation σ=0.57), 2.84 (σ=0.96) and 6.64 (σ=1.41) [m/s^2]. The distinguished movement of chest doesn't appear at starting and stopping time in driving of 0.5 and 1.0 [m/s^2]. It seems that the chest movements appear between the drive acceleration of 1.0 and 2.0 [m/s^2].

During the backward drive, the passenger behaves in the same way as shown in Figure 10. The amplitude of head swing is similar to that of forward drive.

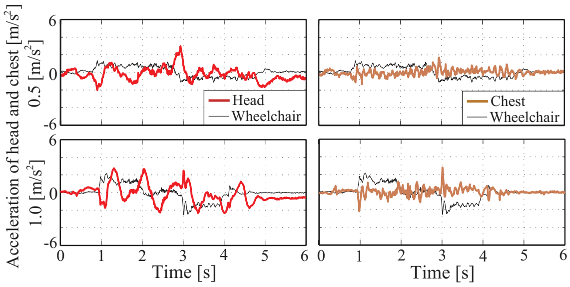


Figure 9: Acceleration of head and chest in the forward transfer of 0.5 and 1.0 [m/s^2].

The average of 6 passenger's maximum head accelerations in driving with 0.5, 1.0 and 2.0 [m/s^2] are respectively 2.29 (σ=0.59), 3.35 (σ=0.05) and 6.11 (σ=1.39) [m/s^2]. However, the chest swing is not appeared in the deceleration interval, and the swing is appeared in the acceleration interval, because of the backrest. The wheelchair's acceleration is opposite to that of forward, because the wheelchair is driven toward the reverse direction.

Questionnaires using SD method with forward and backward driving is shown in Figure 11 and 12 respectively. The right side in figure is a positive side, and left side is a negative side for all items of assessment. These results are total assessments about 6 passengers. Assessment of all items such as "Good Ride or Bad Ride", "Comfortable or Uncomfortable", "No-Swing or Swing", "Defensive or Aggressive" and "Bland or Pungent" are the significant difference with changing the drive acceleration by the analysis of variance was detected. Through this result, it is determined that the passenger's assessment becomes worse while increasing the drive acceleration during forward-backward direction drive.

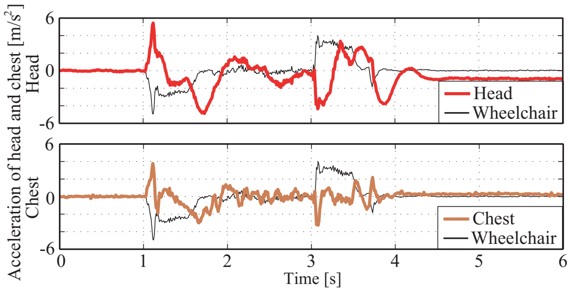


Figure 10: Acceleration of head and chest in the backward transfer of 2.0 [m/s^2].

4.1.3 Experiment in Lateral Direction Drive

Passenger's behavior in the rightward drive is shown in Figure 13 and Figure 14. When the wheelchair drives toward the right direction, the passenger's behavior is almost the same as that of forward or back-

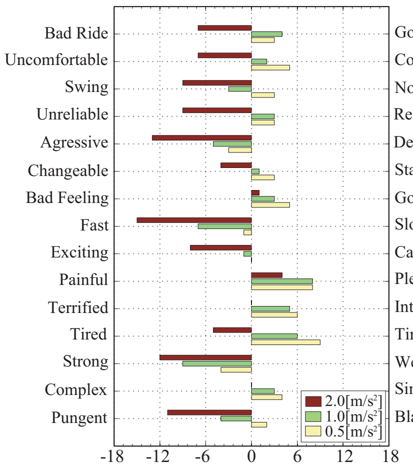


Figure 11: SD questionnaire result in the forward transfer.

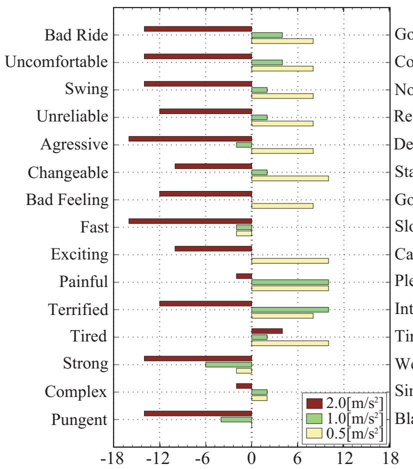


Figure 12: SD questionnaire result in the backward transfer.

ward drive. However, the head swing amplitude is bigger than that of forward drive. The average of maximum acceleration in 0.5, 1.0 and 2.0 [m/s^2] are respectively 4.4 (σ=1.18), 4.52 (σ=0.87) and 6.8 (σ=1.02) [m/s^2]. The distinguished chest swing only appears in driving of 2.0 [m/s^2] at the stopping and starting, and its amplitude is as the same level as the forward drive.

In the driving toward the left direction, the passenger behaves is similar to that of rightward drive as shown in Figure 15. The average of maximum acceleration in 0.5, 1.0 and 2.0 [m/s^2] are respectively 2.59 (σ=0.56), 3.59 (σ=0.37) and 7.85 (σ=1.46) [m/s^2]. The chest movements only appears in the driving of 2.0 [m/s^2].

Questionnaires using SD method in rightward and leftward driving is shown in Figure 16 and 17 respectively. The results show that the passenger's assessment becomes worse with increasing the drive acceleration during lateral direction drive. Furthermore, the lateral direction movements making the head swing bigger are more comfortable than the backward of the forward-backward direction movements. The backward driving is the most uncomfortable direction in four one. The leftward and rightward direction driving which often thought to be uncomfortable, are as comfortable as the case of the forward movements.

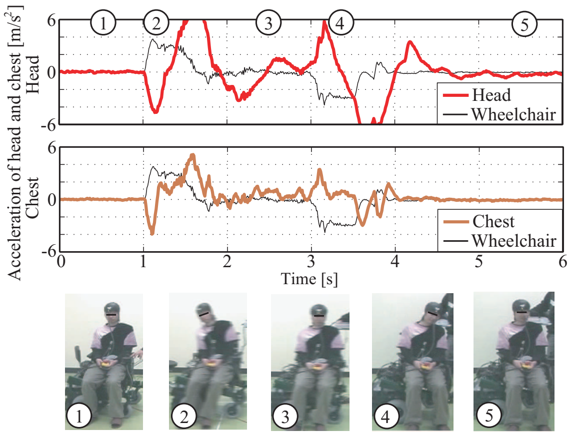


Figure 13: Acceleration of head and chest in the rightward transfer of 2.0 [m/s^2].

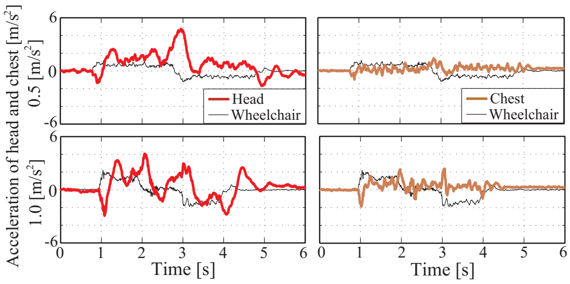


Figure 14: Acceleration of head and chest in the rightward transfer of 0.5 and 1.0 [m/s^2].

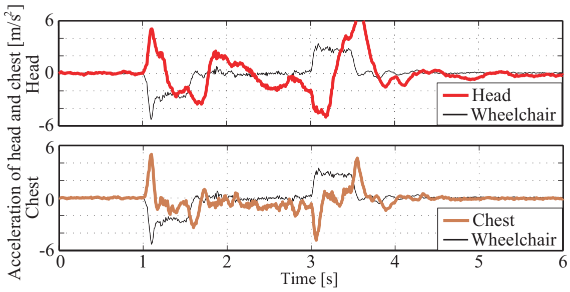


Figure 15: Acceleration of head and chest in leftward transfer of 2.0 [m/s^2].

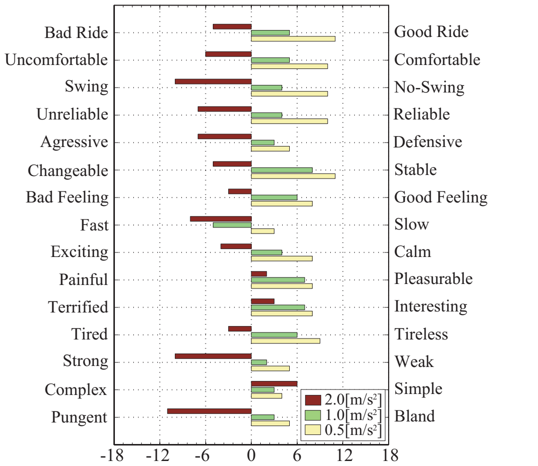


Figure 16: SD questionnaire result in the rightward transfer.

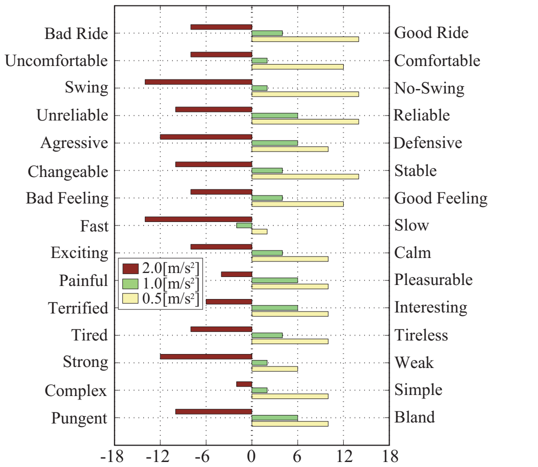


Figure 17: SD questionnaire result in the leftward transfer.

4.1.4 Experiment for Detecting Threshold Acceleration of the Chest Movements

The passenger's posture behavior at 1.0, 1.3, 1.7 and 2.0 [m/s^2] in the case of the forward drive are shown in Figure 18. Head and chest behavior at 1.0 and 1.3 [m/s^2] is alike as shown in Figure 18. The head movement becomes bigger while increasing the drive acceleration, and that is as the same as the previous experimental results. The distinguished chest movement appears in the case of 1.7 and 2.0 [m/s^2]. Threshold of the chest movement is estimated to be the value between 1.3 and 1.7 [m/s^2]. By the results using SD method, the passenger's comfort sensation becomes worse while increasing the drive acceleration. Then, the passenger says that they feel discomfort sensation when the chest is moved. The similar results were obtained in other passenger's experiments.

Through these results, it seems that the chest movements induce the passenger's uncomfortable, and uncomfortable feelings can be reduced by controlling of chest movements.

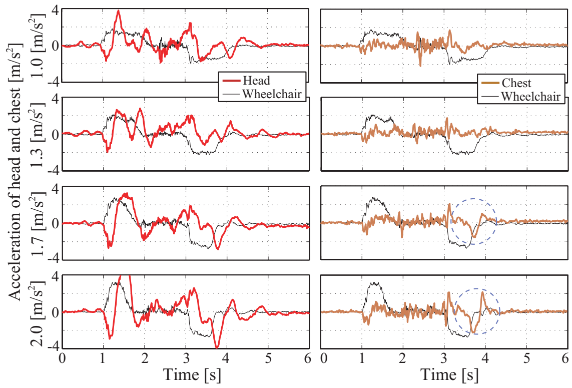


Figure 18: Acceleration of head and chest in case of forward transfer using various driving acceleration.

4.2 Comfort Sensation Focused on the Swing of Chest Part

Through the previous experiments, the passenger's comfort may be influenced by the chest movement. Therefore, the effect of chest movement on the comfort is investigated in details.

4.2.1 Experimental Description

In this section by using various patterns of the wheelchair is moved that acceleration and deceleration as shown in Table 2. The maximum velocity, distance and other condition are the same with the previous experiments.

Pattern I has a big deceleration such that induces the chest movement, because there is no backrest in the front part as shown in Figure 19. Pattern IV has a big acceleration such that induces the chest movement by the same reason. Drives by these two patterns are thought to be uncomfortable.

Table 2: Driving patterns.

[m/s^2]	Forward		Backward	
Driving pattern	I	II	III	IV
Acceleration	1.0	2.0	1.0	2.0
Deceleration	2.0	1.0	2.0	1.0

In the present experiments, to evaluate discomfort sensation more clearly, a Paired Comparison Test is used. After forward drive (or backward drive) two pattern of I and II (or III and IV), it was asked which pattern induces you discomfort feeling. 10 people with healthy and standard proportion were selected as subjects.

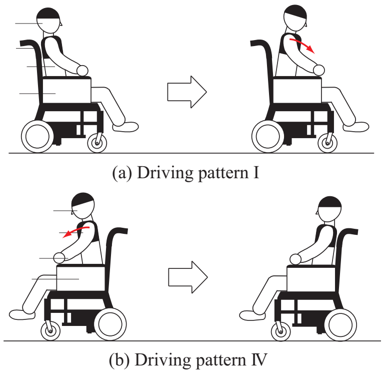


Figure 19: Drive patterns cause the chest swing.

4.2.2 Experimental Result and Discussion

Experimental results are shown in Figure 20. With respect to the pattern I and IV, the chest movement is observed at the anticipated point, because there is not backrest for the direction such that acceleration value is big. On the other hand, in the case of pattern II and III, the chest is not largely moved, because backrest exists for the direction in which the acceleration is big.

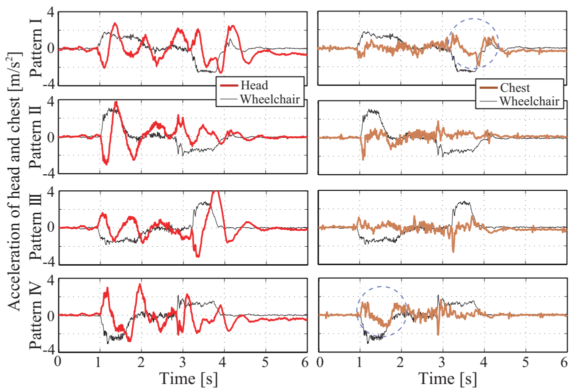


Figure 20: Acceleration of head and chest in the drive pattern I, II, III and IV.

According to the result of a paired comparison test as shown in Table 3, the discomfort sensations were almost same. The swing amplitude of I and IV is almost same. Here, that of II and III is almost same, where, the swing amplitude of I and IV is bigger than that of II and III. Therefore, the remarkable relationship between passenger's chest movement and discomfort could not be detected through this experiment conditions as seen from Table 2. However, comfort is almost same in each pattern. This may be the influence of the pressure from the backrest as a main factor.

Through these experiments, it becomes clearly that the passenger's body behavior is one of the factor

Table 3: Number of passenger with discomfort feeling. (Total number ; N=10)

	Forward		Backward	
Driving pattern	I	II	III	IV
Discomfort passenger	4	6	6	4

that affects passenger's behavior. However, passenger's comfort is not depending on only passenger's behavior, and it is possible that there is another factor. For example, velocity or distance of wheelchair driving, and the chest pressure with backrest. These analyses are the near future problems to be solved.

5 CONCLUSION

The results obtained in this paper were as follows.

1. The transformed value of acceleration into the movements coordinates of wheelchair at the various body parts from the sensor data of sensor coordinates could be calculated by the proposed algorithm, and its effectiveness was verified.
2. Amplitude of passenger's head swing became bigger while increasing the drive acceleration.
3. The chest movement was largely appeared from the certain value of the drive acceleration between 1.3 and 1.7 [m/s^2]
4. From SD questionnaires, high acceleration drive caused passenger's discomfort.
5. Ride comfort for the movements to vehicles, backward direction is the most uncomfortable in four kinds of movements, and forward, leftward and rightward directions are almost same level with respect to comfort. Further, we showed the possibility that OMW will be able to apply as the transfer wheelchair without particular discomfort in the same level with conventional wheelchair with the ability of only forward and backward movements.

In the future, to find another factor, experimental condition such as the driving pattern, the ambient environment, and the evaluating method for passenger's comfort should be studied. Further, motion of slant and rotation should be studied for investigating the comfort driving.

ACKNOWLEDGEMENTS

This study was partially supported by The 21st Century COE (Center of Excellence) Program "Intelligent

Human Sensing", and also The Hori Information Science Promotion Foundation. Authors should give the sincere thanks to both organizations.

REFERENCES

C.H.Lee, C.W.Kim, and M.Kawatani (2005). Dynamic response analysis of monorail bridges under moving trains and riding comfort of trains. In *Engineering Structures Vol.27 pp.1999-2013*.

H.Kitagawa, T.Beppu, T.Kobayashi, and K.Terashima (2002). Motion control of omni-directional wheelchairs considering patient comfort. In *Proceedings of the IFAC'02 World Congress pp. T-Tu-E20*.

ISO-2631-1 (1997). International organization for standardization. mechanical vibration and shock - evaluation of human exposure to whole-body vibration - part1 : General requirements.

J.Urbano, K.Terashima, and T.Miyoshi (2005). Velocity control of an omni-directional wheelchair considering user's comfort by suppresing vibration. In *Proceedings of the IEEE IROS 2005 pp.2385-2390*.

K.Terashima, J.Urbano, and H.Kitagawa (2006). Enhancement of maneuverability of a power assist omni-directional wheelchair by application of neuro-fuzzy control. In *Proceedings of the ICINCO 2006 pp.67-75*.

S.Maeda, M.Futatsuka, and J.Yonesaki (2003). Relationship between questionnaire survey results of vibration complaints of wheelchair users and vibration transmissibility of manual wheelchair. In *Health and Preventive Medicine Vol.8 pp.82-89*.

S.Shimada, K.Ishimura, and M.Wada (2002). Dynamic posture measurement and evaluation of electric-power wheelchair with human. In *Proceedings of the ROBOMEC '02 1P1-G02 (in Japanese)*.

Y.Yamagishi and H.Inooka (2005). Driver assistant system for improvement of ride quality. In *Collected papers of Society of Automotive Engineers of Japan Vol.36 No.1 pp.241-246 (in Japanese)*.

AN AUGMENTED STATE VECTOR APPROACH TO GPS-BASED LOCALIZATION

Francesco Capezio, Antonio Sgorbissa and Renato Zaccaria
Laboratorium D.I.S.T. – University of Genova
{Francesco.Capezio, Antonio.Sgorbissa, Renato.Zaccaria}@unige.it

Keywords: Mobile Robotics; Localization; State Estimation.

Abstract: The ANSER project (Airport Night Surveillance Expert Robot) is described, exploiting a mobile robot for autonomous surveillance in civilian airports and similar wide outdoor areas. The paper focuses on the localization subsystem of the patrolling robot, composed of a non-differential GPS unit and a laser rangefinder for map-based localization (inertial sensors are absent). Moreover, it shows that an augmented state vector approach and an Extended Kalman filter can be successfully employed to estimate the colored components in GPS noise, thus getting closer to the conditions for the EKF to be applicable.

1 INTRODUCTION

The work described in this paper is part of the ANSER[1] project, an ongoing project for autonomous surveillance in civilian airports and similar wide outdoor areas[2]. Within this framework, a system composed of two parts is foreseen: a mobile autonomous robot (also referred to as UGV – Unmanned Ground Vehicle), whose sensors and actuators have been especially crafted to successfully perform night patrols, and a fixed supervision station, which is under direct control of a human supervisor. The main surveillance task is to detect differences between perceived and expected environmental conditions; in particular to verify the state of doors and barriers, to verify the presence of allowed/non allowed persons in the current area, and to identify unexpected objects. In ANSER, this is done through the combination of a laser rangefinder and an on-board panning video camera, and it obviously requires a sufficient accuracy in self-localization to be able to recognize "what is normal" and "what is not" in a given area. A first system prototype is currently being tested at the Villanova d'Albenga Airport (Figure 6), where it is asked to patrol a wide outdoor area and the indoor Airport Terminal.

In the last years several autonomous surveillance systems based on a mobile platform have been presented.

A very interesting example in this sense is the MDARS project, a joint USA Army-Navy development effort (Heath-Pastore, *et al.*, 1999). The MDARS goal is to provide multiple mobile platforms that perform random patrols within assigned areas of warehouses and storage sites, both indoor and in semi-structured outdoor environments, such as storage yards, dock facilities, and airfields. MDARS-E apparently meets the requirements of the ANSER domain. However, it is immediate to notice that high performance are obtained by over equipping the system with a huge set of different sensorial devices and – consequently – providing adequate onboard computing power to process the huge amount of available data. For example, the localization and navigation subsystem of MDARS-E requires the joint use of a differential GPS, a fiber-optic gyro and the recognition of retroreflective landmarks via a laser-based proximity sensor.

In (Saptharishi, *et al.*, 2002) a network of mobile all-terrain vehicles and stationary sentries are exploited in an autonomous surveillance and reconnaissance system. The vehicles are equipped with video cameras, and are able to detect moving objects, classify them using a differential learning algorithm, and track their motion. Each robot relies for localization on a Differential GPS and an IMU (Inertial Measurement Unit); a PC/104 for the

[1] ANSER is an acronym for Airport Night Surveillance Expert Robot, and the Latin name for "goose" (referring to the Capitoline Geese which –according to tradition - neutralized a nighttime attack by the Gauls during the siege of Rome).
[2] Funded by the Parco Scientifico Tecnologico della Liguria (PSTL), www.pstliguria.it.

locomotion task, and three networked PCs for planning, perception and communication are required.

In (Vidal, *et al.*, 2002) a team of UAVs (Unmanned Arial Vehicle) and UGVs pursue a second team of evaders adopting a probabilistic game theory approach (pursuit-evasion is a classical problem that had been deeply investigated (Volkan, *et al.*, 2005)). Also in this case, the robots need enough computational power to manage a Differential GPS receiver, an IMU, video cameras and a color-tracking vision system.

In (Rybski,*et al.*, 2002) a multirobot surveillance system is presented. A group of miniature robots (Scouts) accomplishes simple surveillance task using an on-board video camera. Because of limitations on the space and power supply available on-board, Scouts rely on remote computers that manage all the resources, compute the decision processes, and finally provide them with action control commands.

As one could expect, autonomous navigation and self-localization capabilities are a fundamental prerequisites in all these systems. This is the reason why, starting from a minimal configuration in which self-localization relies on an Inertial Measurement Unit (IMU) and a Carrier Phase, Differential GPS receiver (CP-DGPS) – see for example (Panzieri, *et al.*, 2002; Schönberg, *et al.*, 1995; Dissanayake, *et al.* 2001; Farrell, *et al.*, 2000) -, a very common approach is to equip the mobile platform with a large set of sensors (video cameras, PIR sensor, RFID sensor, sonar, laser range finders etc.), thus consequently requiring a high computational power and complex data filtering techniques.

In partial contrast with this "over equipping" philosophy, the ANSER self-localization sub-system relies only on a standard (non-differential) GPS unit, and on a laser rangefinder. Unfortunately, GPS data are known to be affected by low-frequency errors that cannot be modeled as zero mean, Additive White Gaussian Noise (AWGN), thus making simple state estimation approaches (e.g., Kalman Filter) unfeasible (Sasiadek, and Wang, 2003). As a main contribution, this work proposes to estimate the low-frequency components of GPS noise through an augmented state vector approach, similar to (Farrell, *et al.*, 2000) (Martinelli, 2002). The paper shows that, by combining laser-based localization and GPS measurement, it is possible to estimate both the robot's position and the non-AWG components of GPS noise.

Section II briefly describes the localization techniques adopted; Section III theoretically investigates the properties of the approach, and carries out an observability analysis; Section IV presents experimental results obtained so far with a realistic simulator, and in a field set-up at the Villanova d'Albenga Airport. Conclusions follow.

2 GPS- AND LASER-BASED SELF-LOCALIZATION

2.1 Gps Based Localization

A single non-differential GPS receiver provides the mobile robot with absolute position measurements, that can be employed to correct the estimate provided by odometry. Unfortunately, the measurement process is corrupted by different error sources, which are consequence of the receiver and the satellites clock bias, the atmospheric delay, the multi-path effect, etc. (Farrell, *et al.*, 2000). The union of these errors is known as Common Mode Error, and it introduces into the GPS measure a greatly colored noise with a significant low-frequency component. Approximately, this can be modeled as a non-zero mean value in GPS errors that varies slowly in time (in the following, it will be referred to as a "bias" in GPS measurements).

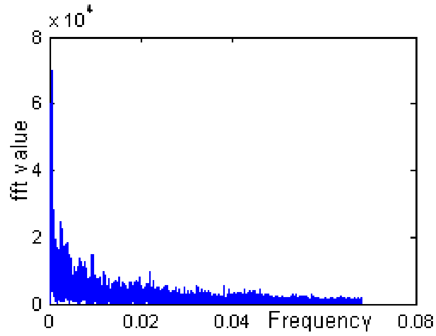


Figure 1: FFT of GPS latitude data.

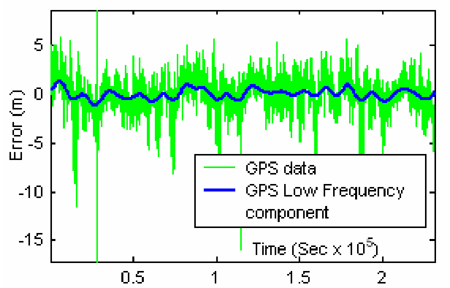


Figure 2: The estimated GPS bias.

The analysis of longitude and latitude data collected at a fixed location during 24 hours shows this effect: by considering the Fast Fourier Transform (FFT) of GPS longitude and latitude data (the latter is shown in Figure 1), low-frequency components can be noticed, corresponding to slow variation of the signal in time. By estimating this bias in GPS measurements, one can expect that – at least in theory – the precision of GPS data should improve, therefore making localization more accurate. The low-frequency component of latitude error is shown separately in Figure 2.

To estimate the bias, an augmented state vector **x** is defined; **x** comprises both (x, y, θ) components of the robot's position, and the (x_{GPS} , y_{GPS}) components of the low-frequency bias in GPS measurements. Notice that, by separating the colored components from Additive White Gaussian components of GPS noise, the system gets closer to the conditions for the Extended Kalman Filter to be applicable. When new measurements are available (i.e., both GPS data and the features detected by a laser rangefinders), the full state vector can be estimated through observations.

2.2 Laser Based Localization

When moving indoor, the robot is provided with an a-priori map of the environment; the laser-based localization subsystem simply updates the position by comparing this map with the features detected by the laser rangefinder.

In particular, (Capezio, *et al.*, 2006) describes in details how segment-like features are extracted from raw data and compared with the a-priori model: 1) line extraction produces a set of lines $\{ l_j \}$; 2) the Mahalanobis distance associated to each couple of line (l_j , m_i) is computed (where $\{ m_i \}$ is a set of oriented segment lines that define the a-priori map); 3) for each l_j, the line m_i for which such distance is minimum is selected and fed to the EKF.

When moving outdoor, lines in the a-priori map correspond to the external walls of buildings. Obviously, a smaller number of features is available outdoor, since the robot mostly traverses areas where no buildings are present at all (especially in the Airport scenario). However, when features are available, they are sufficient to estimate the full state vector, and – under some assumptions – the estimate stays valid even when the laser cannot provide any further information.

3 SYSTEM ARCHITECTURE

As anticipated, the proposed approach relies on the idea of "guessing" the bias that affects GPS measurements at a given time, by including it in the state to be estimated. The resulting augmented state vector is shown in Equation 1.

$$\mathbf{x}^T = \begin{bmatrix} x & y & \theta & x_{GPS} & y_{GPS} \end{bmatrix} \quad (1)$$

It includes the robot's position and orientation with respect to a fixed frame F_w, and the two components (with respect to the same frame) of the bias in GPS measurements.

After integrating the dynamic equations of the system through a standard Euler approximation with step size Δt=1, the system can be described with the following finite difference Equations:

$$\begin{cases} x_k = x_{k-1} + ds_{k-1} * \cos\theta_{k-1} \\ y_k = y_{k-1} + ds_{k-1} * \sin\theta_{k-1} \\ \theta_k = \theta_{k-1} + d\omega_{k-1} \\ x_{GPS,k} = x_{GPS,k-1} \\ y_{GPS,k} = y_{GPS,k-1} \end{cases} \quad with \quad \begin{matrix} ds = (dr + dl)/2 \\ d\omega = (dr - dl)/D \end{matrix} \quad (2)$$

The first three equations represent the discrete approximation of the robot's inverse kinematics, by assuming a unicycle differential drive model. As usual, *dr* and *dl* indicate the linear displacements of the right and left driving wheels, and *D* is the distance between them.

In the last two Equations, the dynamic of the bias $\mathbf{x}_{GPS}{}^T = \begin{bmatrix} x_{GPS} & y_{GPS} \end{bmatrix}$ is modeled. Notice that a constant dynamic is assumed for $\mathbf{x}_{GPS}{}^T$, since no cues are available to make more accurate hypotheses. This means that, when predicting the new state in the time-update phase of the EKF, $\mathbf{x}_{GPS}{}^T$ is left unchanged. However, the predicted value of $\mathbf{x}_{GPS}{}^T$ is updated whenever new measurement are available (i.e., in the correction phase of the EKF), thus finally producing an estimate that varies in time, and hopefully approximates the actual bias in GPS measurements. The approach seems reasonable whenever a component of the state vector changes slowly in time with respect to the remaining components, which is exactly the case.

When considering the remaining noise that affects **x**, the process can be described as governed by a non-linear stochastic difference Equation in the form

$$\mathbf{x}_k = \mathrm{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1}) \text{ with } \mathbf{x} \in \Re^5, \ \mathbf{w} = N(0,W) \quad (3)$$

where **w** represents the process noise, and **u** is the driving function; i.e., the 2-dimensional vector describing the current wheels displacements *dr* and *dl* ($\mathbf{u}^T = [dr \quad dl]$).

For what concerns errors in the process, they are currently modeled through a vector $\mathbf{w}^T = [wr \quad wl \quad wg]$. The first two element sums up to *dl* and *dr* (e.g., when the left encoder returns *dl*, the actual path traveled by the left wheel is *dl+wl*), whereas *wg* represents the error made in assuming that the bias has not changed since the last iteration of the Filter. By assuming that **w** has a zero-mean Gaussian distribution with covariance matrix **W**, systematic errors in odometry due to the approximate knowledge of the robot's geometric characteristics are not explicitly considered (in theory, geometric parameters should be included in the augmented state vector as well, as proposed in (Martinelli, 2002)).

Observations are provided both by the GPS and, when available, by the laser rangefinder. The measurements provided by the GPS are a non-linear function of the state:

$$\begin{cases} z_{long} = C_{LONG}(z_{lat}) * (x + x_{GPS}) \\ z_{lat} = C_{LAT}(z_{lat}) * (y + y_{GPS}) \end{cases} \tag{4}$$

Where $\mathbf{z}=(z_{long}, z_{lat})$ is a 2-dimensional vector representing the longitude and latitude of the mobile robot, by supposing the X-axis of F_w lying on the parallel passing through F_w's origin, and F_w's Y-axis lying on the meridian. The measurement model is not linear, mainly because the relationship between georeferenced data (i.e., latitude and longitude) and the estimated *x*- and *y*- coordinates varies with the latitude itself, as a consequence of the non planarity of the earth surface (as determined by C_{LONG} (z_{lat}) and C_{LAT} (z_{lat})).

For each line l_j observed by the laser rangefinder, the line m_i that best matches l_j can be expressed as:

$$\begin{cases} z_\rho = \dfrac{a_i \cdot x_k + b_i \cdot y_k + c_i}{\sqrt{a_i^2 + b_i^2}} \\ z_\alpha = \tan^{-1}\left(\dfrac{-a_i}{b_i}\right) - \theta_k \end{cases} \tag{5}$$

where a_i, b_i and c_i are the parameters characterizing the implicit equation of m_i; z_ρ and z_α are, respectively, the distance between the line and the robot, and the angle between the line and the robot's heading.

When putting together Equations 4 and 5, the measurement model results to be a non-linear function of the state:

$$\mathbf{z}_k = h(\mathbf{x}_k, \mathbf{v}_k) \qquad with \qquad \mathbf{v} = N(0, R) \tag{6}$$

Since non-AWG components of the GPS noise are estimated in the state vector, the remaining noise can be reasonably modeled with the vector **v**, a zero-mean AWG noise with covariance matrix **R.**

Equation 2 can be used to compute the a-priori state estimate at time *k*. Next, whenever new GPS or laser rangefinder data are available, they are fused with the a-priori estimate through an Extended Kalman Filter to produce a new estimate, thus reducing errors that are inherently present in odometry and providing a new estimate for the GPS bias.

Obviously, to evaluate the soundness of the previous assertion, it is necessary to perform an observability analysis of the system. The Kalman theorem requires to compute the observability matrix $Q = \left[H^T \mid \quad A^T H^T \mid \quad ... \qquad \mid (A^4)^T H^T\right]$, where *A* and *H* are the Jacobian matrices of the partial derivatives of *f* and *h* with respect to **x** (*Q*'s full expression is not shown for sake of brevity). The analysis shows that *Q* has full rank (and hence the state is fully observable) only when at least two observations l_j and l_m are available, corresponding to non-parallel lines m_i and m_n, together with a single GPS measurement.

Matrix *A* results to be:

$$A = \begin{bmatrix} 1 & 0 & -ds_{k-1} \cdot \sin(\theta_{k-1}) & 0 & 0 \\ 0 & 1 & ds_{k-1} \cdot \cos(\theta_{k-1}) & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{7}$$

whereas *H* results to be:

$$H = \begin{bmatrix} 1/C_{LONG} & 0 & 0 & 1/C_{LONG} & 0 \\ 0 & 1/C_{LAT} & 0 & 0 & 1/C_{LAT} \\ \dfrac{a_i}{\sqrt{a_i^2 + b_i^2}} & \dfrac{b_i}{\sqrt{a_i^2 + b_i^2}} & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ \dfrac{a_m}{\sqrt{a_m^2 + b_m^2}} & \dfrac{b_m}{\sqrt{a_m^2 + b_m^2}} & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix} \tag{8}$$

By inspecting matrix *H*, one could infer that the filter is updated only when a triplet of observations are available (i.e., two non-parallel lines and one

GPS measurement). However, this is assumed only to investigate the state's observability; during experiments, the laser rangefinder and the GPS returns observations asynchronously, and each observation is used to update the state as soon as available. The observability analysis demonstrates that, even if each measurement is able to correct the state only partially, the state is fully observable when more measurements are considered in cascade.

Unfortunately, in outdoor areas it often happens that the laser cannot detect any line mapped in the a-priori map: since the localization algorithm relies only on GPS data, the H matrix fed to the KF comprises only the first two rows in Equation 8. When this happens – as already stated - only a subspace of the state space results to be observable; by computing again the observability matrix Q, this yields the result in Equation 9.

When $ds_{k-1} \neq 0$, i.e. when the translational speed of the robot is not null (Capezio, *et al.*, 2005), the rank of Q is 3. The rank is not full since Q's first column (corresponding to the x-component of the state vector $\mathbf{x}$) equals the fourth column (corresponding to the x_{GPS}-component), and Q's second column (i.e., the y-component) equals the fifth column (i.e., the y_{GPS}-component). On the opposite, Q's third column (corresponding to the θ-component of $\mathbf{x}$) is linearly independent from the others.

Q's analysis confirms the intuition that – when no laser data are available – the subspace defined by $x+x_{GPS}$, $y+y_{GPS}$, and θ is fully observable: the robot's orientation is still corrected by GPS data (when $ds_{k-1} \neq 0$, see also (Capezio, *et al.*, 2005)), and the position has a permanent error that depends on the current estimate of the GPS bias.

$$Q = \begin{bmatrix} 1/C_{LONG} & 0 & 0 & 1/C_{LONG} & 0 \\ 0 & 1/C_{LAT} & 0 & 0 & 1/C_{LAT} \\ 1/C_{LONG} & 0 & -\frac{1}{C_{LONG}} \cdot ds_{k-1} \cdot \sin(\theta_{k-1}) & 1/C_{LONG} & 0 \\ 0 & 1/C_{LAT} & \frac{1}{C_{LAT}} \cdot ds_{k-1} \cdot \cos(\theta_{k-1}) & 0 & 1/C_{LAT} \\ 1/C_{LONG} & 0 & -\frac{2}{C_{LONG}} \cdot ds_{k-1} \cdot \sin(\theta_{k-1}) & 1/C_{LONG} & 0 \\ 0 & 1/C_{LAT} & \frac{2}{C_{LAT}} \cdot ds_{k-1} \cdot \cos(\theta_{k-1}) & 0 & 1/C_{LAT} \\ 1/C_{LONG} & 0 & -\frac{3}{C_{LONG}} \cdot ds_{k-1} \cdot \sin(\theta_{k-1}) & 1/C_{LONG} & 0 \\ 0 & 1/C_{LAT} & \frac{3}{C_{LAT}} \cdot ds_{k-1} \cdot \cos(\theta_{k-1}) & 0 & 1/C_{LAT} \\ 1/C_{LONG} & 0 & -\frac{4}{C_{LONG}} \cdot ds_{k-1} \cdot \sin(\theta_{k-1}) & 1/C_{LONG} & 0 \\ 0 & 1/C_{LAT} & \frac{4}{C_{LAT}} \cdot ds_{k-1} \cdot \cos(\theta_{k-1}) & 0 & 1/C_{LAT} \end{bmatrix} \quad (9)$$

In this case, the innovation due to GPS measurements is distributed by the Kalman gain onto x-, y-, x_{GPS}, and y_{GPS} components of the state according to the current value of the state covariance matrix $\mathbf{P}$. Since a constant dynamic is assumed for x_{GPS}, and y_{GPS}, corrections are adequately distributed onto the state components only if the actual bias in GPS changes slowly, and given that a new area where the laser rangefinder is able to guarantee full observablility will soon be available. This is reasonable when assuming a cyclic patrol path for autonomous surveillance, with periodic visits to outdoor areas that are mapped in the a-priori model and observable by the laser.

4 EXPERIMENTAL RESULTS

Many experiments in a realistic simulated environment and at the Albenga Airport (Figure 6) have been performed. Moreover, in order to test the system under different conditions, experiments are performed by varying the robot' speed. The GPS sensor is realistically simulated (data are taken from real GPS in a 24-hours interval), as well as errors in laser measurements and odometry.

In all the simulated tests, the robot is requested to move along a path that is identical to the patrol performed in Villanova d'Albenga Airport. Furthermore, the dimension and the position of the exterior walls of buildings considered for map-based localization in the simulated environment realistically emulate the Airport scenario (see Figure 3; walls are visible only in a very limited area of the Airport). The simulated robot travels for the whole day, performing a cyclic patrol about 500 meters long; next, the experiment is repeated by varying the navigation speed.

Tests have been performed in two modalities: A-tests correspond to localization with GPS bias estimation, and B-tests are performed without bias estimation.

Table 1: Errors statistics in B-tests.

Nav. Speed (m/s).	$\bar{e}_x$	σ_x	$\bar{e}_y$	σ_y	$\bar{e}_\vartheta$	σ_ϑ
0.4	1.50	1.01	1.18	0.55	0.067	0.047
0.9	1.36	0.97	1.75	1.22	0.048	0.039
1.4	1.47	1.14	1.29	0.72	0.043	0.035

Table 2: Errors statistics in A-test.

Nav. Speed (m/s).	$\bar{e}_x$	σ_x	$\bar{e}_y$	σ_y	$\bar{e}_\vartheta$	σ_ϑ
0.4	1.07	1.04	1.63	1.13	0.057	0.053
0.9	0.94	0.84	0.78	0.70	0.040	0.033
1.4	0.97	0.80	1.23	1.14	0.041	0.031

Tables 1 and 2 summarize results. In Table 1, results of the B-tests are shown. In this case the state vector includes only the position and the orientation of the robot. Table 2 shows the results of A-tests. In both Tables, $\bar{e}_x$ and $\bar{e}_y$ represent the average error between *x*- and *y*- components of **x** and the real robot's position (ground truth); $\bar{e}_\vartheta$ is the average error in the robot's heading; $\sigma_x, \sigma_y, \sigma_\vartheta$ are the corresponding standard deviations.

When computing the average position error as $\mathbf{x}_{err} = E(\sqrt{e_x^2 + e_y^2})$, this yields:

- B-tests: $\mathbf{x}_{err}$=2.12m, $\mathbf{x}_{err}$=1.92m, and $\mathbf{x}_{err}$=2.08m when the robot is moving at the speed of 0.4m/s, 0.9m/s, and 1.4m/s respectively

- A-tests: $\mathbf{x}_{err}$=1.52m,$\mathbf{x}_{err}$ =1.34m, and $\mathbf{x}_{err}$ =1.37m (for 0.4m/s, 0.9m/s, and 1.4m/s).

By analyzing the data, some considerations can be made:

- $\mathbf{x}_{err}$ is always below 2.5m, with and without the GPS bias Estimation.

- In the best case (1.4m/sec navigation speed), the bias estimation reduces $\mathbf{x}_{err}$ of about 34%.

- The improvement due to GPS bias estimation seems to increase when the navigation speed increase; this could be consequence of the fact that the robot returns quicker in areas where laser-based localization is possible.

- Standard deviations are always comparable with averages; i.e., the error oscillates significantly around its mean value.

- Errors in the θ-component are bounded, and always below 10 degrees.

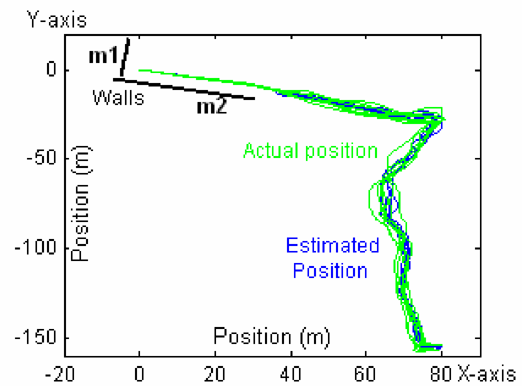


Figure 3: Test performed at 0.9 m/sec.

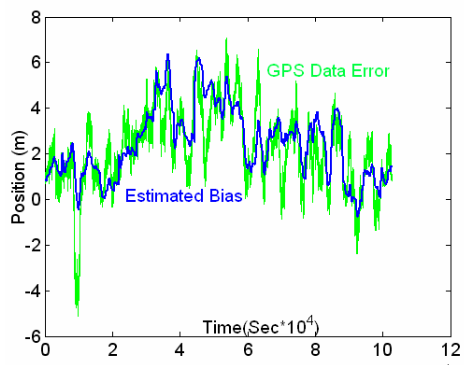


Figure 4: GPS bias estimation.

Figure 3 shows a plot of the robot's trajectory during an A-test 3 hours long (moving at 0.9 m/sec); Figure 4 shows the estimated longitude bias during such test

Real world experiments have been carried out as well with the ANSER robot in the Albenga Airport. During the test, the robot is manually driven at 1.0m/s along a pre-established cyclic path that is about 500 meters long (walls are similar to Figure 3). Different A- and B-tests are been performed (each lasting about 3 hours), by memorizing the robot's estimated position in a finite number of selected places along the path.

Figure 5 shows the estimated robot's position in 8 different places along the real path (the Figure can be superimposed onto Figure 3 to infer where features for laser-based localization are visible).

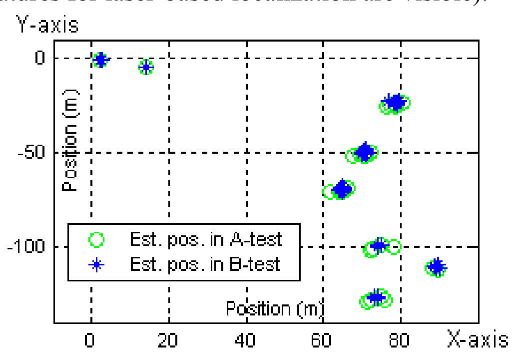


Figure 5: Estimated position in a real scenario.

In the real scenario, A-tests exhibit a smaller improvement in performance with respect to simulation. This is probably due to the fact that the state is fully observable (and hence the GPS bias can be correctly estimated) only when laser data are available. However, this happens in the vicinity of buildings (e.g., walls in Figure 3 correspond to a hangar); unfortunately, near a building the GPS signal is less precise, since the GPS satellites are occluded by the building itself.

Figure 6: The robot ANSER at Albenga Airport.

5 CONCLUSIONS

The paper describes the localization subsystem of a mobile robot that has been designed for night patrols and surveillance tasks within a civilian airport. The localization subsystem is a small – but fundamental component – of the whole project (ANSER – Airport Night Surveillance Expert Robot). Instead of equipping the robot with a huge amount of expensive sensors (and the computing power that is adequate to deal with them), a simple approach is chosen that relies exclusively on a non-differential GPS unit and a laser rangefinder (i.e., inertial sensors are absent). Laser measurements are exploited only in some areas of the outdoor patrol path of the robot, i.e. where it is possible detect line features and match them against an a-priori model of the environment. Along the rest of the path, the robot relies on GPS-based localization. An Extended Kalman Filter algorithm is employed to estimate an augmented state vector comprising the robot position and orientation, together with the low frequency components (bias) of the GPS error.

A formal model of the whole localization subsystem is given, including an analysis of the system's observability. The experiments performed in a realistic simulated environment and at Villanova d'Albenga Airport have confirmed the expectations, showing that the approach reasonably improves the localization accuracy of the system. Obviously, the accuracy achieved is not sufficient for fine motion in cluttered areas; however, for surveillance applications in which the robot has to reach an area of interest and to further investigate on the basis of local sensor feedback, it seems appropriate.

REFERENCES

Capezio, F., Sgorbissa, A., and Zaccaria (2005), R. GPS-based Localizzation for UGV Performing Surveillance Patrols in Wide Outdoor Areas, In *proc. of Int. Conf. on Field and Service Robotics*,

Capezio, F., Mastrogiovanni, F., Sgorbissa, A., and Zaccaria, R. (2006), Fast Position Tracking of an Autonomous Vehicle in Cluttered and Dynamic Indoor Environments, In *proc. of 8th International IFAC Symposium on Robot Control*

Dissanayake**,** G., Sukkarieh, S., Nebot, E. and Durrant-Whyte, H. (2001), The aiding of a low-cost strapdown inertial measurement unit using vehicle model constraints for land vehicle applications, In *IEEE Trans. on Robotics and Automation*, vol. 17, Issue 5, pp. 731 – 747,

Farrell, J. A., Givargis, T. and Barth, M. (2000), Real-time differential carrier phase GPS-aided INS, In *IEEE Trans. Contr. Syst. Technol.*, vol. 8, pp. 709-721.

Heath-Pastore, T., H.R. Everett, and K. Bonner (1999), Mobile Robots for Outdoor Security Applications, In *American Nuclear Society 8th International Topical Meeting on Robotics and Remote Systems*, Pittsburgh.

Martinelli, A. (2002) The Odometry Error of a Mobile Robot with a Synchronous Drive System. In *IEEE Trans. on Robotics and Automation*, Vol 18, no. 3, pp 399-405.

Panzieri, S., Pascucci, F., and Ulivi, G. (2002), An Outdoor Navigation System Using GPS and Inertial Platform, I*EEE Trans. on mechatronics*, vol. 7, no 2

Rybski, P. E., Stoeter, S. A., Gini, M., Hougen, D. F. and Papanikolopoulos, N P. (2002), Performance of a Distributed Robotic System Using Shared Communications Channels, In *IEEE Trans on Robotics and Automation*, vol. 18, no.5

Saptharishi M., Oliver, C. S., Diehl, C. P., Bhat, K. S., Dolan, J. M., Trebi-Ollennu, A. and Khosla, P. K. (2002), Distributed Surveillance and Reconnaissance Using Multiple Autonomous ATVs: CyberScout, In *IEEE Trans. on Robotics and Automation,* vol. 18, no.5

Sasiadek, J. Z. and Wang, Q. (2003), Low cost automation using INS/GPS data fusion for accurate positioning, In *Robotica,* Vol. 21, pp. 255-260. Cambridge University Press.

Schönberg, T., Ojala, M., Suomela, J., Torpo, A., and Halme, A (1995), Positioning an autonomous off-road vehicle by using fused DGPS and inertial navigation, In *2nd IFAC Conference on Intelligent Autonomous Vehicles*, Espoo, pp. 226-231

Vidal, R.,, Shakernia, O., Kim, H. j., Shim, D. H. and, Sastry, S. (2002), Probabilistic Pursuit–Evasion Games: Theory, Implementation, and Experimental Evaluation, In *IEEE Trans. on Robotics and Automation*, vol. 18, no. 5,

Volkan, I., Sampath, K. and Sanjeev K. (2005), Randomized Pursuit-Evasion in a Polygonal Environment, In *IEEE Trans. on robotics*,vol.21, no. 5

HOMOGRAPHY-BASED MOBILE ROBOT MODELING FOR DIGITAL CONTROL IMPLEMENTATION

Andrea Usai and Paolo Di Giamberardino
Department of Computer and System Sciences "Antonio Ruberti"
University of Rome "La Sapienza"
via Eudossiana 18, Rome, Italy
{*usai,digiamberardino*}*@dis.uniroma1.it*

Keywords: Nonholonomic mobile robot, multirate digital control, visual servoing.

Abstract: The paper addresses the development of a kinematic model for a system composed by a nonholonomic mobile robot and a camera used as feedback sensor to close the control loop. It is shown that the proposed homography-based model takes a particular form that brings to a finite sampled equivalent model. The design of a multirate digital control is then descibed and discussed, showing that exact solutions are obtained. Simulation results are also reported to put in evidence the effectiveness of the proposed approach.

1 INTRODUCTION

In several applications a digital implementation of a controller is required even if the plant dynamics is a continuous time one.

The classical approach, especially for nonlinear systems, makes use of a preliminary design of a continuous time control law that is then implemented in a digital way by means of a digital device (computer or microcontroller, for example). In this approach the digital control law is computed as an approximation of the continuous time one. This kind of approach is usually denoted as *indirect digital control*.

Such an approach can lead to poor controller performance due to the system approximation performed: the choice of the sampling time affects the capacity of the digital controller, that generate a piecewise constant signal, to have a behavior close enough to the nominal continuous time controller that assures the desired performances with a continuous time output signal.

All the works done in the field of visual servoing, deal with continuous time systems (see, for example, in the case of mobile robots: (Chen et al., 2006; Lopez-Nicolas et al., 2006; Mariottini et al., 2006)). Image acquisition, elaboration and the nonlinear control law computation are very time consuming tasks. So it is not uncommon that the system is controlled, in a indirect digital control context, with a sampling rate of $0.5Hz$. Clearly this kind of choices strongly affects the system performances, since the use of a slow rate in the controller implies that only a slow evolution of the control system can be required to assure satisfactory behaviors.

To face the drawbacks of poor system approximation, the trivial solution is slowing down the system dynamics slowing down the controls variation rate. This is not always possible, but when the kinematic model is considered, the system dynamics depends only on the velocity imposed by the controller (there is no *drift* in the model) and so it is quite easy to limit the effect of a poor system approximation.

In general, taking into account, also in the design phase, the discrete time nature of the control system can lead to better closed loop system performances. This means that the control law designed *directly* in the discrete time domain can better address the discrete time evolution of the controlled system. This is well understood in the linear case and it is also true for a quite large class of nonlinear systems, the ones that admits a finite or an exact sampled representation (Monaco and Normand-Cyrot, 2001). In fact, a necessary step in the design of a digital controller is to deal with a discrete time plant; if the plant is described by means of a continuus time dynamics, a discretization of the plant is required. The possibility of an exact discretization, better if finite, of the continuous time model clearly affects the overall performances of the control system since in this case no approximations are performed in the conversion from continuous time to discrete time.

2 THE CAMERA-CYCLE MODEL

In this section the kinematic model of the system composed of a mobile robot (an unicycle) and a camera, used as a feedback sensor to close the control loop, is presented.

To derive the mobile robot state, the relationship involving the image space projections of points that lie on the floor plane, taken from two different camera poses, are used. Such a relationship is called *homography*. A complete presentation of such projective relations and their properties is shown in (R. Hartley, 2003).

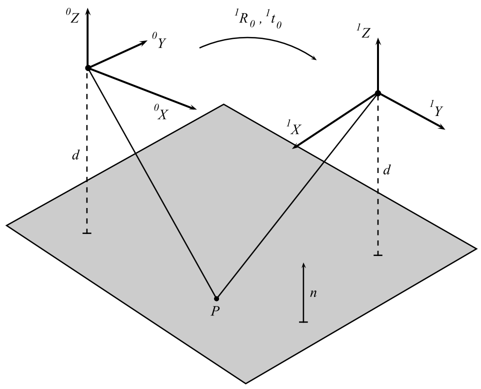


Figure 1: 2-view geometry induced by the mobile robot.

2.1 The Geometric Model

With reference to Figure 1, the relationship between the coordinates of the point P in the two frames is

$$\begin{bmatrix} {}^1P \\ 1 \end{bmatrix} = \begin{bmatrix} {}^1R_0 & {}^1t_0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} {}^0P \\ 1 \end{bmatrix} \tag{1}$$

It is an affine relation that becomes a linear one in the homogeneous coordinate system. If the point P belongs to a plane in the 3D space with normal versor n and distance from the origin d, it holds that

$$\begin{bmatrix} n^T & d \end{bmatrix} \begin{bmatrix} {}^0P \\ 1 \end{bmatrix} = 0 \quad \Rightarrow \quad -\frac{n^{T\,0}P}{d} = 1 \tag{2}$$

note that $d > 0$, since the interest is in the planes observed by a camera and so they don't pass trough the optical center (that is the camera coordinate system origin).

Combining the Equation 1 and the right term of 2, the following relation holds

$$^1P = \left({}^1R_0 - \frac{1}{d}\,{}^1t_0 n^T \right) {}^0P = H\,{}^0P \tag{3}$$

The two frame systems in Figure 1 represent the robot frame after a certain movement on a planar floor. Choosing the unicycle like model for the mobile robot, the matrix H become

$$H = \begin{bmatrix} \cos\theta & \sin\theta & \frac{1}{d}\left(X\cos\theta + Y\sin\theta\right) \\ -\sin\theta & \cos\theta & \frac{1}{d}\left(-X\sin\theta + Y\cos\theta\right) \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

is the homography induced by the floor plane during a robot movement between two allowable poses. Note that $[X, Y, \theta]^T$ is the state vector of the mobile robot with reference to the first coordinate system.

2.2 The Kinematic Model

Taking four entries of matrix H such that

$$\begin{aligned} h_1 &= \cos\theta \\ h_2 &= \sin\theta \\ h_3 &= x\cos\theta + y\sin\theta \\ h_4 &= -x\sin\theta + y\cos\theta \end{aligned} \tag{5}$$

and noting that, for the sake of simplicity, the distance d has been chosen equal to one, since it is just a scale factor that can be taken into account in the sequel, the kinematic unicycle model is

$$\begin{aligned} \dot{X} &= v\cos\theta \\ \dot{Y} &= v\sin\theta \\ \dot{\theta} &= \omega \end{aligned} \tag{6}$$

where v and ω are, respectively, the linear and angular velocity control of the unicycle.

Differentiating the system in Equation 5 with respect to the time and combining it with the system in Equation 6, one obtains

$$\begin{aligned} \dot{h}_1 &= -h_2\omega \\ \dot{h}_2 &= h_1\omega \\ \dot{h}_3 &= h_4\omega + v/d = h_4\omega + {}_d v \\ \dot{h}_4 &= -h_3\omega \end{aligned} \tag{7}$$

that is the kinematic model of the homography induced by the mobile robot movement.

3 FROM CONTINUOUS TO DISCRETE TIME MODEL

In the first part of this section, it will be presented how to derive a the discrete time system model from the continuous time one. Afterwards, a control (and planning) strategy for the discrete time system is introduced and applied. Simulations will be presented to prove the presented strategy effectiveness.

3.1 The General Case

Suppose a system such that

$$\dot{x} = f(x) + \sum_{i=1}^{m} u_i g_i(x) \tag{8}$$

with $f, g_1, \ldots, g_m : M \rightarrow \mathbb{R}^n$, analytical vector fields.

To derive a discrete time system from the previous one, suppose to keep constant the controls $u_1, \ldots, u_m$, by means of a zero order holder, for $t \in [kT, (k+1)T)$ and $k \in \mathbb{N}$. Suppose that the system output is sampled (and acquired) every T seconds, too. The whole system composed by a z.o.h, the system and the sampler is equivalent to a discrete time system.

Following (Monaco and Normand-Cyrot, 1985; Monaco and Normand-Cyrot, 2001), it is possible to characterize the discrete time system derived by a continuous time nonlinear system. Sampling the system in Equation 8 with a sampling time T, the discrete time dynamics becomes

$$\begin{aligned} x(k+1) &= x(k) + T\left(f + \sum_{i=1}^{m} u_i(k) g_i\right)\Bigg|_{x(k)} + \\ &\quad + \frac{T^2}{2}\left(L_{f+\sum_{i=1}^{m} u_i[k] g_i}\right)^2 (Id)\Bigg|_{x(k)} + \ldots \\ &= F^T(x(k), u(k)) \end{aligned} \tag{9}$$

where $L_{(\cdot)}$ denotes the Lie derivative. It is possible to see that, this series is locally convergent choosing an appropriate T. See (Monaco and Normand-Cyrot, 1985) for details.

The problem here is the analytical expression of $F^T(x(k), u(k))$. It can happen that it is not possible to compute it. Otherwise, if from the series of Equation 9 it is possible to derive an analytical expression for its limit function, the system of Equation 8 is said to be *exactly discretizable* and its limit function is called an *exact sampled representation* of 8. If, better, the series results to be finite, in the sense that all the terms from a certain index on goes to zero, a *finite sampled representation* is obtained.

Finite sampled representations are transformed, under coordinates changes, into exact sampled ones ((Monaco and Normand-Cyrot, 2001)). As obvious, finite discretizability is not a coordinate free property while exact discretizability is. Note that the existence of an exact sampled representation corresponds to analytical integrability.

The possibility of getting an exact sampled representation is related to the existence of a state feedback which allow that ((Di Giamberardino et al., 1996b)).

A nonholonomic system as the one of Equation 8, can be transformed into a chained form system by means of a coordinate change.

This leads to a useful property for discretization pointed out in (Monaco and Normand-Cyrot, 1992). In fact it can be seen that a quite large class of nonholonomic systems admit exact sampled models (polynomial state equations). Among them, one finds the chained form systems which can be associated to many mechanical systems by means of state feedbacks and coordinates changes.

3.2 The Camera-Cycle Case

Suppose the controlled inputs are piecewise constant, such that

$$\begin{cases} v(t) = v_k \\ \omega(t) = \omega_k \end{cases} \quad t \in [kT, (k+1)T) \tag{10}$$

where $k = 0, 1, ..$ and T is the sampling period.

Since the controls are constant, it is possible to integrate the system in Equation 7 in a linear fashion. It yields to

$$\begin{aligned} \begin{bmatrix} h_1(k+1) \\ h_2(k+1) \end{bmatrix} &= A(\omega_k) \begin{bmatrix} h_1(k) \\ h_2(k) \end{bmatrix} \\ \begin{bmatrix} h_3(k+1) \\ h_4(k+1) \end{bmatrix} &= A(\omega_k)^T \begin{bmatrix} h_3(k) \\ h_4(k) \end{bmatrix} + B(\omega_k)_d v_k \end{aligned} \tag{11}$$

where

$$\begin{aligned} A(\omega_k) &= \begin{bmatrix} \cos\omega_k T & -\sin\omega_k T \\ \sin\omega_k T & \cos\omega_k T \end{bmatrix} \\ B(\omega_k) &= \begin{bmatrix} \sin(\omega_k T)/\omega_k \\ (\cos(\omega_k T) - 1)/\omega_k \end{bmatrix} \end{aligned} \tag{12}$$

If one considers the angular velocity input as a time varying parameter, the system in Equation 11 become a linear time varying system. Such a property allows an easy way to compute the evolution of the system. Precisely, its evolution becomes

$$\begin{bmatrix} h_1(k) \\ h_2(k) \end{bmatrix} = \prod_{i=0}^{k-1} A(\omega_{k-i-1}) \begin{bmatrix} h_1(0) \\ h_2(0) \end{bmatrix}$$

$$\begin{bmatrix} h_3(k) \\ h_4(k) \end{bmatrix} = \prod_{i=0}^{k-1} A(\omega_{k-i-1})^T \begin{bmatrix} h_3(0) \\ h_4(0) \end{bmatrix} +$$

$$+\sum_{j=0}^{k-2} \left(\prod_{i=j+1}^{k-1} A(\omega_{k-i})^T \right) B(\omega_j)\, {}_d v_j + B(\omega_{k-1})\, {}_d v_{k-1} \tag{13}$$

where the sequences $\{v_k\}$ and $\{\omega_k\}$ are the control inputs. This structure will be useful in the sequel for the control law computation.

4 CONTROLLING A DISCRETE TIME NONHOLONOMIC SYSTEM

Interestingly, difficult continuous control problems may benefit of a preliminary sampling procedure of the dynamics, so approaching the problem in the discrete time domain instead of in the continuous one.

Starting from a discrete time system representation, it is possible to compute a control strategy that solves steering problems of nonholonomic systems. In (Monaco and Normand-Cyrot, 1992) it has been proposed to use a *multirate digital control* for solving nonholonomic control problems, and in several works its effectiveness has been shown (for example (Chelouah et al., 1993; Di Giamberardino et al., 1996a; Di Giamberardino et al., 1996b; Di Giamberardino, 2001)).

4.1 Camera-cycle Multirate Control

The system under study is the one in Equation 11 and the form of its state evolution in Equation 13.

The problem to face is to steer the system from the initial state $h_0 = [1,0,0,0]^T$ (obviously corresponds to the origin of the configuration space of the unicycle) to a desired state ${}^d h$, using a multirate controller.

If r is the number of sampling periods chosen, setting the angular velocity constant over all the motion, one gets for thestate evolution

$$\begin{aligned} \begin{bmatrix} h_1(r) \\ h_2(r) \end{bmatrix} &= A^r(\bar{\omega}) \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ \begin{bmatrix} h_3(r) \\ h_4(r) \end{bmatrix} &= A^{r-1}(\bar{\omega}) B(\bar{\omega})\, {}_d v_0 + \\ &+ A^{r-2}(\bar{\omega}) B(\bar{\omega})\, {}_d v_1 + \ldots + B(\bar{\omega})\, {}_d v_{r-1} \end{aligned} \tag{14}$$

At this point, given a desired state, one just need to compute the controls. The angular velocity $\bar{\omega}$ is firstly calculated such that:

$$\begin{bmatrix} {}^d h_1 \\ {}^d h_2 \end{bmatrix} = A^r(\bar{\omega}) \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{15}$$

Once $\bar{\omega}$ is chosen, the linear velocity values ${}_d v_0, \ldots, {}_d v_{r-1}$ can be calculated solving the linear system

$$\begin{bmatrix} {}^d h_3 \\ {}^d h_4 \end{bmatrix} = R \begin{bmatrix} {}_d v_0 \\ {}_d v_1 \\ \ldots \\ {}_d v_{r-1} \end{bmatrix} \tag{16}$$

$$R = \begin{bmatrix} A^{k-1}(\bar{\omega}) B(\bar{\omega}) & \ldots & B(\bar{\omega}) \end{bmatrix} \tag{17}$$

which is easily derived from the second two equations of 14.

Note that, for steering from the initial state to any other state configuration, at least $r = 2$ steps are needed, except for the configuration that present the same orientation of the initial one. More precisely, it can be seen that if this occurs, the angular velocity $\bar{\omega}$ is equal to zero or Π/T and the matrix R in Equation 16 become singular. Exactly that matrix shows the reachability space of the discrete time system: if $\bar{\omega} \neq \{0, \Pi/T\}$ then the vector B is rotated r-times and the whole configuration space is spanned.

Furthermore, if 2 is the minimum multirate order to guarantee the reachability of any point of the configuration, one can choose a multirate order such that $r > 2$ and the further degrees of freedom in the controls can be used to accomplish the task obtaining a smoother trajectory or avoiding some obstacles, for instance. Note that it can be achieved solving a quadratic programming problem as

$$\min_{V \in \mathbb{V}} \frac{1}{2} V^T \Sigma V + \Gamma V \tag{18}$$

where Σ and Γ are two weighting matrixes, such that the robot reaches the desired pose, granting some optimal objectives. Other constraints can be easily added to take account of further mobile robot movements requirements.

4.2 Closing the Loop with the Planning Strategy

Let $[{}^d\theta, {}^dh_3, {}^dh_4]^T$ be the desired system state and mark the actual one with the subscript k . Note that, in the control law development, the orientation of the mobile robot is used, instead of the first two components of the system of Equation 11.

Summarize the algorithm steps as

0. Set $r_k = r$
1. Choose $\bar{\omega} = \left({}^d\theta - \theta_k\right) / r_k T$
2. Compute the control sequence V as $\min_V V^T V$ such that $RV = \begin{bmatrix} {}^dh_3 \\ {}^dh_4 \end{bmatrix}$ with the same notation of Equations 16 and 18
3. If $r_k > 2$ then $r_k = r_k - 1$ and go to step 1. Otherwise, the algorithm stops.

The choice of the cost function shown leads to a planned path length minimization. If the orientation error of the point 1 is equal to zero, it needs to be perturbed in order to guarantee some solution admissibility to the programming problem of point 2.

Furthermore, since the kinematic controlled model derives directly from an homography, it is possible use the homographies compositional property to easily update the desired pose, from the actual one, at every control computation step. Exactly, since

$$ {}^dH_0 = {}^dH_{r-1}{}^{r-1}H_{r-2} \ldots {}^kH_{k-1} \ldots {}^1H_0 \qquad (19) $$

it is possible to easily update the desired pose as needed for the close loop control strategy.

A Simulated path is presented in Figures 2: the ideal simulated steer execution is perturbed by the presence of some additive noise in the controls. This simulate the effect of some non ideal controller behavior (wheel slipping, actuators dynamics, ...). The constrained quadratic problems involved in the controls computation are solved using an implementation of the algorithm presented in (Coleman and Li, 1996)

5 CONCLUSION

In this paper, a kinematic model for a system composed by a mobile robot and a camera, has been presented. Since such a model is exactly discretizable, it has been possile to propose a multirate digital control strategy able to steer the system to a desired pose in an exact way. The effectiveness of the control scheme adopted has been verified by simulations.

REFERENCES

Chelouah, A., Di Giamberardino, P., Monaco, S., and Normand-Cyrot, D. (1993). Digital control of nonholonomic systems two case studies. In *Decision and Control, 1993., Proceedings of the 32nd IEEE Conference on*, pages 2664–2669vol.3.

Chen, J., Dixon, W., Dawson, M., and McIntyre, M. (2006). Homography-based visual servo tracking control of a wheeled mobile robot. *Robotics, IEEE Transactions on [see also Robotics and Automation, IEEE Transactions on]*, 22(2):406–415.

Coleman, T. and Li, Y. (1996). A reflective newton method for minimizing a quadratic function subject to bounds on some of the variables. *SIAM Journal on Optimization*, 6(4):1040–1058.

Di Giamberardino, P. (2001). Control of nonlinear driftless dynamics: continuous solutions from discrete time design. In *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on*, volume 2, pages 1731–1736vol.2.

Di Giamberardino, P., Grassini, F., Monaco, S., and Normand-Cyrot, D. (1996a). Piecewise continuous control for a car-like robot: implementation and experimental results. In *Decision and Control, 1996., Proceedings of the 35th IEEE*, volume 3, pages 3564–3569vol.3.

Di Giamberardino, P., Monaco, S., and Normand-Cyrot, D. (1996b). Digital control through finite feedback discretizability. In *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, volume 4, pages 3141–3146vol.4.

Lopez-Nicolas, G., Sagues, C., Guerrero, J., Kragic, D., and Jensfelt, P. (2006). Nonholonomic epipolar visual servoing. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 2378–2384.

Mariottini, G., Prattichizzo, D., and Oriolo, G. (2006). Image-based visual servoing for nonholonomic mobile robots with central catadioptric camera. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 538–544.

Monaco, S. and Normand-Cyrot, D. (1985). On the sampling of a linear analytic control system. In *Decision and Control, 1985., Proceedings of the 24th IEEE Conference on*, pages pp. 1457–1461.

Monaco, S. and Normand-Cyrot, D. (1992). An introduction to motion planning under multirate digital control. In *Decision and Control, 1992., Proceedings of the 31st IEEE Conference on*, pages 1780–1785vol.2.

Monaco, S. and Normand-Cyrot, D. (2001). Issues on nonlinear digital control. *European Journal of Control*, 7(2-3).

R. Hartley, A. Z. (2003). *Multiple View Geometry in Computer Vision*. Number ISBN: 0-521-54051-8. Cambridge University Press.

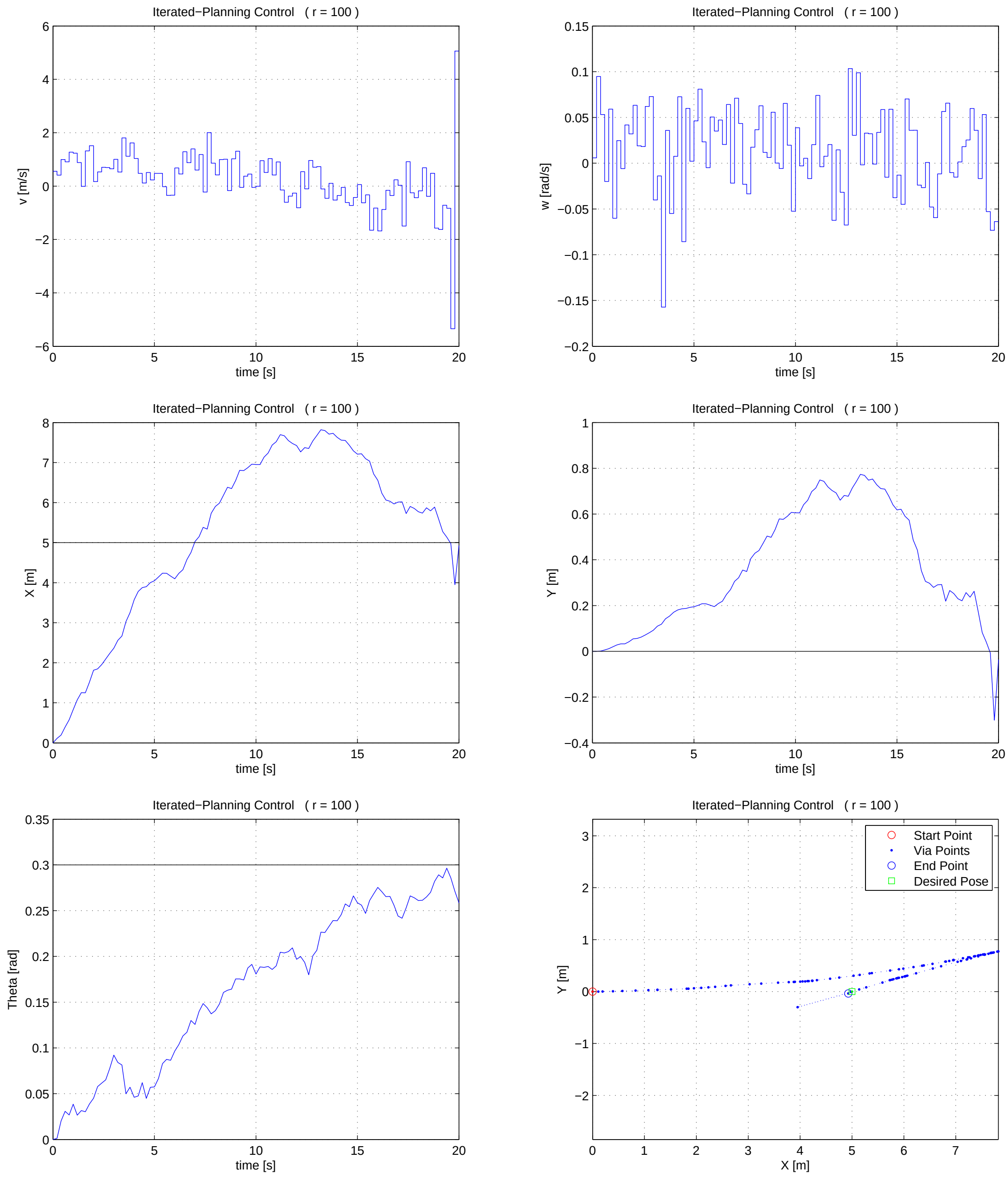


Figure 2: Multirate control simulation ($d = 1m$). Additive random noise on controls (gaussian with std.dev. 0.5 and 0.05, for v and ω respectively).

BEHAVIOR ACTIVITY TRACE METHOD
Application to Dead Locks Detection in a Mobile Robot Navigation

Krzysztof Skrzypczyk
Department of Automatic Control, Silesian University of Technology, Akademicka 16, 44-100 Gliwice, Poland
krzysztof.skrzypczyk@polsl.pl

Keywords: Mobile robot, behavior–based robotics, data representation, navigation.

Abstract: In the paper a novel approach to representation of a history in a mobile robot navigation is presented. The main assumptions and key definitions of the proposed approach are discussed in this paper. An application of the method to detection a dead end situations that may occur during the work of reactive navigation systems is presented. The potential field method is used to create an algorithm that gets the robot out of the dead-lock. Simulations that show the effectiveness of the proposed method are also presented.

1 INTRODUCTION

An autonomous mobile robot is a machine that can operate in an environment model of which is unknown apriori and can react dynamical changes of this environment (Cox and Wilfong, 1990). Inaccurate sensors, world unpredictability and imperfect control often cause the failure of traditional, planner based approaches to a mobile robot control system design (Cox and Wilfong, 1990). Therefore more efficient and faster methods of a mobile robot collision free movement control have been developed. One of them is a purely reactive architecture introduced in (Braitenberg, 1984, Brooks 1991) which implements a control strategy as a collection of stimulus-reaction pairs. The system consists of a collection of purely reactive rules that contain minimal internal state. These systems use no internal models, perform no search and merely lookup and take appropriate action for each set of sensor readings. A Braitenberg algorithm (Braitenberg, 1984) could be a good example of the reactive architecture. Behavior based control system architecture (Arkin, 1998, Brooks, 1991, Mataric , 1992, Michaud and Mataric, 1998) embody some of the properties of reactive systems and may contain reactive components. However the primary feature of behavior based systems is their distributed nature. They consist of a collection of parallel executing behaviors devoid of centralized reasoning module. The behaviors are more powerful than purely reactive rules because they may use different forms of internal representation and perform computations on them in order to decide what action to take. One of the key issues that appears while designing behavior based systems is just a representation of the knowledge about an environment the robot is dedicated to work in (Michaud and Mataric, 1998). Since these systems are intended to work with the low cost, inaccurate sensors, the problem of building a model of an workspace is emerging. Here in this paper the method of knowledge representation based on so called behavior activity trace is presented. An idea of the proposed approach is to store in a time ordered way the knowledge of events that happened during the robot work. What is crucial for this method is the fact that these events are marked and recognized by behavior characteristic sequences. While collecting a knowledge of characteristic events a sort of event-map is built. The advantage of the proposed method is that this form of representation does not consume much memory resources. Another one is the computations using this sort of map can be performed in an efficient way. In the paper an example application of this method is presented. A module of behavior based system designed for detection of emergency situations during the work of the system is described. Simulation experiments proved the proposed approach to be effective.

2 THE CONTROL SYSTEM

A design of the control system used in this work is based on the behavior-based idea of control (Arkin, 1998, Brooks, 1991). The system is composed of behaviors that process the state and sensory information into proper set-points for motion controller - linear and angular velocity. The coordination of behavior activities is made by fixed priority arbiter. A general diagram of the controller is presented in fig.1. Its easy to distinguish five main modules of the controller:

- Behavior definition module;
- Arbitration module;
- Control computation module;
- Task execution watcher module;
- Dead lock detector module;

Each behavior can be perceived as a schema of reaction to a given stimulus that comes from an environment and it is represented by current sensory information and state of the robot itself. In our system there are eight behaviors implemented. First four of them *(avoid left, avoid right, avoid front, speed up*) are responsible for avoiding collisions with objects located correspondingly on the left, right, frontal and back side of the robot platform. Fifth behavior (*goal tracking*) minimizes the distance between the robot and the target. Behavior *stop* simply stops the robot in case a collision is detected or the target is reached. Sixth behavior called *stroll* makes the robot goes straight in case when no objects are detected.

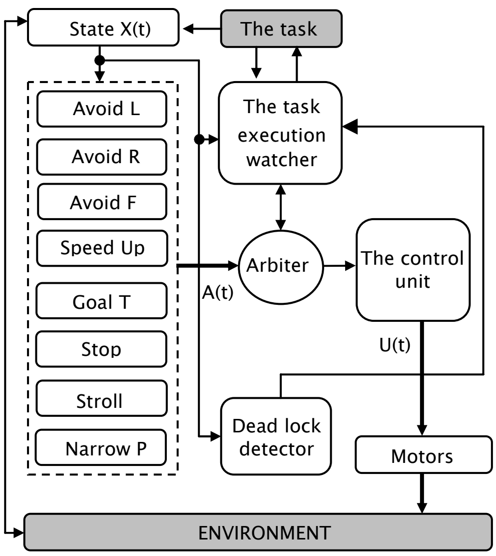


Figure 1: The control system architecture.

And the last behavior – *narrow passage* stabilizes robot movement preventing oscillations during going through narrow passages. Each behavior is designed as a function which maps a part of input data X into activation level of the given behavior a_i, and it is defined by the s-class function.

Describing details of implementation of particular behaviors is out the scope of the paper. But for understanding concepts that are presented in the next sections it is reasonable to show more details of an implementation of the behavior *goal tracking*. This behavior will be used further to generate action of escape from the dead lock situation. The work of this behavior consist in monitoring the error $\Delta\Theta$ between the current heading of the robot $\Delta\Theta_r$ and the desired heading $\Delta\Theta_d$. The way of calculating the value of the last one is presented in the section 5. In each moment of time this behavior is checking the error $\Delta\Theta$. A value of the error is determined from:

$$\Delta\Theta = \begin{cases} \left|\Theta_d - \Theta_r\right| & \text{for } \left|\Theta_d - \Theta_r\right| \le \pi \\ 2\pi - \left|\Theta_d - \Theta_r\right| & \text{for } \left|\Theta_d - \Theta_r\right| > \pi \end{cases} \tag{1}$$

The heading $\Delta\Theta_d$ is a set point generated by the control system. If $\Delta\Theta$ it is greater than some threshold value ε then the output signal of the behavior is rising rapidly according to:

$$f_{gt}(\Delta\Theta) = \frac{1}{1 + e^{-\alpha_{gt}(\Delta\Theta - \varepsilon)}} \tag{2}$$

Given behavior generates a control optimal from the perspective of its own "point of view". Therefore the method of coordination has to be used to obtain final control of the robot optimal from the perspective of the task executed. In our case we use the method of priority arbitration, which select this k-th behavior which satisfy the following:

$$k = \max_{i=1,2,\ldots,B=7}(a_i q_i) \tag{3}$$

where q_i denotes the priority fixed to the i-th behavior. The activation of the selected k-th behavior constitute the basis for computation of robot control. Both angular and linear velocities are defined by heuristic functions of the activation level of the selected behavior:

$$u = [\omega, v] = f_k(a_k) \tag{4}$$

The next module called task execution watcher, is designed as a finite state automaton the role of which is to supervise the process of the task execution. The automaton is determined by four states presented in fig.2:

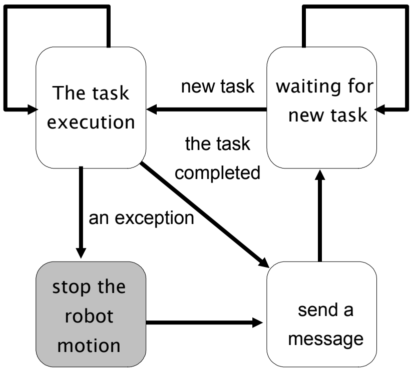


Figure 2: The finite state automaton of the task execution watcher module.

The module starts its work in a state of waiting for a new task to do. If the new task is sent to the module it will switch itself to the state of execution of the task - the robot moves in collision free way toward the target. If task is completed the module will send a message to the global coordinator and switch to the first state. If any exception happens during task execution (collision detection for instance) the robot will stop, send appropriate message to the global coordinator, and switch to the first state.
The aim of the above is only to sketch the main principles and ideas of construction of the behavior based motion controller module. Detailed description exceeds the scope of the work and is not its main subject. For more details please refer to (Skrzypczyk, 2005).

3 BEHAVIOR ACTIVITY TRACE

The problem of local minima as well as dead lock situations are the weak points of reactive systems. Moreover, the fact that in a given moment the robot is provided only with current information from sensors makes the problem of detection of dead locks hard to solve. There are many reasons of dead end situations occurrence. One of them (and probably the most common one) is a structure of a workspace the robot operates inside of which. Reactive systems are usually designed to work with an inaccurate sensory systems. The information provided by this kind of sensor is not sufficient for construction of precise maps that would be useful in environmental structure analysis. Therefore another methods of data representation and processing should be applied. Here in this paper we propose to use a method of behavior activity trace. The key issue of the proposed approach is that the system does not store and analyze the information abut the shape of the environment but it utilizes information about events. Since the events are caused by a configuration of the environmental objects information about the structure of the workspace is obtained in an indirect way. The discussion of the proposed method we start with defining the notion of the activity trace itself. While the control system is working, in each discrete moment of time behaviors are activated. Next in the arbitration process the most appropriate one is selected. Since the activity of behavior is strictly related to the configuration of the workspace, the place the robot was located when given behavior was activated can be perceived as a part of a symbolic map. These places are called further characteristic points.

Definition 1: The characteristic point CP_k we call a point in a cartesian space of coordinates (x_{kCP} , y_{kCP}) defined by a location of the center of a mass of the robot (x_r, y_r) recorded in a moment when the *k*-th characteristic event occurred.

For the purpose of this work three characteristic events were defined.

Event 1
This event is determined by a moment of beginning of the process of navigation. The result of detection of this event distinguishes the characteristic point CP1.

Event 2
This event is set up when the behavior g*oal tracking* was selected to control the robot. It is related to the situation when robot is far from any obstacle and starts the tracking of the goal. The characteristic point CP2 is related to this event.

Event 3
Third event is defined by an occurrence of a situation when the behavior *avoid front* was selected to control the robot and the condition $\Delta\Theta < \varepsilon$ is satisfied at the same time. Such a condition denotes that the behavior *goal tracking* is slightly activated or is not activated at all. The conditions above can be interpreted as a detection of the obstacle on the course of the vehicle straight toward the target. Occurrence of such an event defines the characteristic point CP_3.
Now the notion of the activity trace can be introduced.

Definition 2: The activity trace isthe time ordered sequence of characteristic points:

$$T = \left\{ CP_k^0, CP_k^1, ..., CP_k^{n-1}, CP_k^n \right\} \quad k = 1,2,3 \tag{5}$$

As can be seen the activity trace is the record of the past activity of the robot by means of characteristic events. The events have been recorded since a moment of the beginning of the navigation process $t=0$ till the present time $t = n\Delta t$.

4 DEAD LOCK DETECTION

The activity trace concept discussed in the previous section was applied as a base of dead lock detection module. On the base of multiple experiments with the controller a few observations have been made. It was stated that a situation when the robot is not able to reach the target is mainly caused by a configuration of environmental objects that form u-shaped lay-by located on the course of the vehicle. The undesirable behavior of the robot manifest in repeating a sequence of actions what push the robot into the dead end. Such a situation can be easily detected using behavior activity trace concept. It was observed that dead end situation described above corresponds to an occurrence of three element chunk of the activity trace *CP2, CP3, CP2*. The illustration of this fact is presented in fig.3.

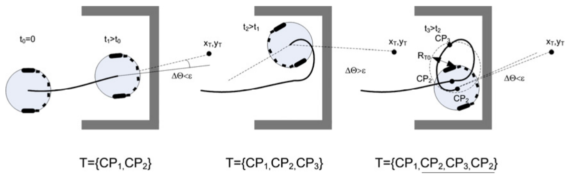


Figure 3: Dead lock detection based on the activity trace concept.

Detection of this three element sequence in the activity trace may show that the system got stuck in a dead lock. Additionally a mutual location of the characteristic points is checked. If all of them are inside of a circle of a radius r_T and a center in the gravity center of these points that means the navigation algorithm failed. Detected dead-lock location is recorded in a buffer and denoted by coordinates $(x_{d,i}, y_{d,i}) \quad i = 1,2,...,N_d$, where N_d denotes the number of all detected dead locks. In such a case the recovery algorithm should be turned on. Although the method is very simple multiple experiments proved its efficiency.

5 RECOVERY ALGORITHM

The method described in the previous section allows to detect the dead end situation the navigation algorithm stuck in. Next step of the control system synthesis is to design a recovery algorithm that is able to get the primary algorithm out of the dead lock. In order to construct the recovery algorithm we utilize the concept of potential field method (Khatib, 1986). According to this concept the workspace of the robot is filled with artificial potential field inside of which the robot is attracted to its target position and repulsed away from the obstacles. The robot navigates in direction of the resulting virtual force vector. In order to apply this idea to get the robot out of the trap each detected dead-lock is considered as a source of repulsive force that has an effect on the robot. So the value of the repulsive force that k-th dead lock acts on the robot is determined from

$$\begin{cases} \left|\mathbf{F_{r,k}}\right| = k_r \left(\dfrac{1}{L_k} - \dfrac{1}{L_0} \right)^2 & \text{for } L_k < L_0 \\ \left|\mathbf{F_{r,k}}\right| = 0 & \text{otherwise} \end{cases} \tag{6}$$

where L_0 is a limit distance of influence of virtual forces. The L_k in (6) denotes a distance between k-th dead lock and the robot:

$$L_k = \sqrt{(x_{d,k} - x_r)^2 + (y_{d,k} - y_r)^2} \tag{7}$$

The repulsive force is a vector sum of forces generated by all dead locks:

$$\mathbf{F_r} = \sum_{k=1}^{N_d} \mathbf{F_{r,k}} \tag{8}$$

The value of the attractive force is determined from the following formula:

$$\left|\mathbf{F}_{a,i}\right| = k_a \frac{1}{L_T^2} \tag{9}$$

Where L_T in (9) is a distance between the robot and the target. Coefficients k_r, k_a determine share of each force-component in the resultant force F and they are adjusted experimentally.Finally the resulting force that acts on the robot is determined as a vector sum:

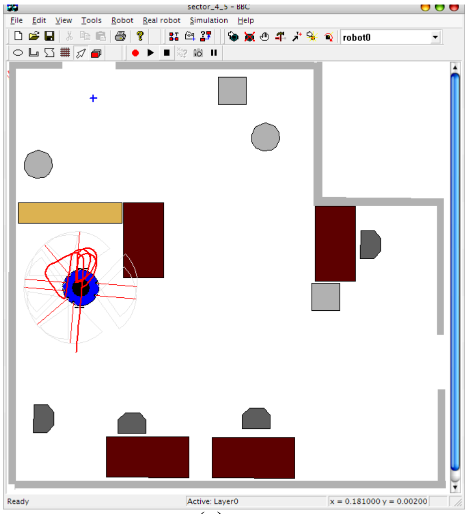
$$\mathbf{F} = \mathbf{F_r} + \mathbf{F_a} \tag{10}$$

the argument of the vector **F** determines a direction the robot is demanded to go. Therefore the value of the $\Delta\Theta_d$ from (1) is calculated as:

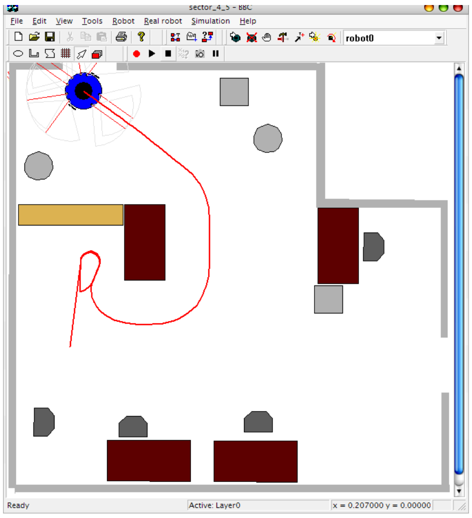
$$\Delta\Theta_d = \arg(\mathbf{F}) \tag{11}$$

6 SIMULATION

In order to verify the presented method it was implemented in the M.A.S.S. simulation environment. The workspace structure as well as the navigation target was set in the way that create dead lock situations. Figure 4 present the result of the simulation of the system without the dead lock detector and recovery algorithm. It is easy to see that the robot suck in the dead lock inside of an u-shaped obstacle.



(a)



(b)

Figure 4: The result of the simulation with dead lock (a) and the recovery algorithm action (b).

In figure 4a the situation when the robot stuck in a dead lock is presented. In figure 4b the result of a work of the dead lock detector and recovery algorithm is shown. The algorithm got the robot out of the dead lock.

7 CONCLUSION

In the paper a novel approach to representing a history in a mobile robot navigation was presented. The method was applied to detect a dead lock situations that may occur during the work of reactive navigation systems. The potential field method was used to create an algorithm that gets the robot out of the dead-lock. Multiple simulations proved an efficiency of this method. There are ongoing works focused on implementation of the method and application to a real robot control.

ACKNOWLEDGEMENTS

This work have been supported by MNI grant no 3T11A 029 28 in the year 2007.

REFERENCES

Althaus P., Christensen H.I., 2003. Behaviour coordination in structured environments. Advanced Robotics, 17(7).

Arkin R. C., 1998. *Behavior-Based Robotics*, MIT Press, Cambridge, MA.

Bicho E., Schoner G., 1997. The dynamic approach to autonomous aobotics demonstrated on a low-level vehicle platform. Robotics and Autonomous Systems, 21(1).

Braitenberg V., 1984. Vehicles: Experiments in synthetic psychology, MIT Press.

Brooks R.A., 1991. Inteligence without representation.
Artificial Inteligence, (47).

Cox I.J., Wilfong G.T., 1990. *Autonomous Robot Vehicles*, Springer-Verlag, New York.

Khatib O.:, 1986. Real-time obstacle avoidance for manipulators and mobile robots, The International Journal of Robotics Research 5(1).

Mataric M.J., 1992. Integration of representation into goal-driven behavior-based robots. IEEE Transactions on Robotics and Automation, 8(3).

Michaud M. , Mataric M. J., 1998. Learning from history for behavior-based mobile robots in nonstationary Conditions. Special issue on Learning in Autonomous robots, Machine Learning, 31(1-3), and Autonomous Robots, 5(3-4).

Skrzypczyk K., 2005. Hybrid control system of a mobile robot. PHD dissertation, Gliwice, Poland, 2005

SELECTIVE IMAGE DIFFUSION FOR ORIENTED PATTERN EXTRACTION

A. Histace
ETIS UMR CNRS 8051, ENSEA-UCP, 6 avenue du Ponceau, 95014 Cergy, France
aymeric.histace@ensea.fr

V. Courboulay, M. Ménard
L3i, University of La Rochelle, Pole Sciences et Technologie, 17000 La Rochelle, France
vincent.courboulay@univ-lr.fr, michel.menard@univ-lr.fr

Keywords: Image Diffusion, Extreme Physical Information, Oriented Pattern Extraction, Selectivity.

Abstract: Anisotropic regularization PDE's (Partial Differential Equation) raised a strong interest in the field of image processing. The benefit of PDE-based regularization methods lies in the ability to smooth data in a nonlinear way, allowing the preservation of important image features (contours, corners or other discontinuities). In this article, a selective diffusion approach based on the framework of Extreme Physical Information theory is presented. It is shown that this particular framework leads to a particular regularization PDE which makes it possible integration of prior knowledge within diffusion scheme. As a proof a feasibility, results of oriented pattern extractions are presented on ad hoc images. This approach may find applicability in vision in robotics.

1 INTRODUCTION

Since the pioneering work of Perona-Malik (Perona and Malik, 1990), anisotropic regularization PDE's raised a strong interest in the field of image processing. The benefit of PDE-based regularization methods lies in the ability to smooth data in a nonlinear way, allowing the preservation of important image features (contours, corners or other discontinuities). Thus, many regularization schemes have been presented so far in the literature, particularly for the problem of scalar image restoration (Perona and Malik, 1990; Alvarez et al., 1992; Catté et al., 1992; Geman and Reynolds, 1992; Nitzberg and Shiota, 1992; Whitaker and Pizer, 1993; Weickert, 1995; Deriche and Faugeras, 1996; Weickert, 1998; Terebes et al., 2002; Tschumperle and Deriche, 2002; Tschumperle and Deriche, 2005). In (Deriche and Faugeras, 1996) Deriche and *al.* propose a unique PDE to express the whole principle.

If we denote $\psi(\mathbf{r},0) : \mathbb{R}^2 \times \mathbb{R}^+ \to \mathbb{R}$ the intensity function of an image, to regularize the considered image is equivalent to a minimization problem of a particular PDE which can be seen as the superposition of two monodimensionnal heat equations, respectively oriented in the orthogonal direction of the gradient and in the tangential direction (Eq. (1 and Fig. 1) :

$$\frac{\partial\psi}{\partial t} = \frac{\phi'(\|\nabla\psi\|)}{\|\nabla\psi\|}\psi_{\xi\xi} + \phi''(\|\nabla\psi\|)\psi_{\eta\eta} \qquad , \qquad (1)$$

where $\eta = \nabla\psi/\|\nabla\psi\|$ and $\xi\perp\eta$ and $\phi : \mathbb{R} \to \mathbb{R}$ is a decreasing function.

This PDE is characterized by an anisotropic diffusive effect in the privileged directions ξ and η allowing a denoising of scalar image.

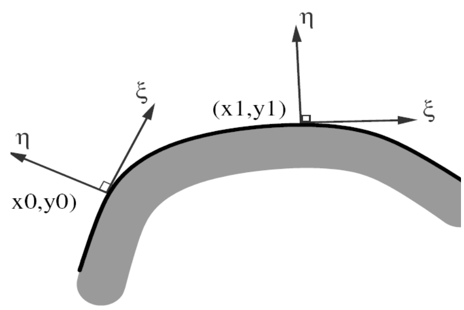


Figure 1: An image contour and its moving vector basis (ξ,η). Taken from (Tschumperle and Deriche, 2002).

The major limitations of this diffusion process is its high dependance to the intrinsic quality of the original image and the impossibility to integrate prior information on the pattern to be restored if it can be characterized by particular data (orientation for example). Moreover, no characterization of the uncertainty/inaccuracy compromise can be made on the studied pixel, since the scale parameter is not directly integrated in the minimisation problem in which relies the common diffusion equations (Nordstrom, 1990).

In this article we propose an original PDE directly integrating the scale parameter and allowing the taking into account of *a priori* knowledge on pattern to restore. We propose more particularly, to derive this PDE, to use a recent theory known as Extreme Physical Information (EPI) recently developed by Frieden (Frieden, 1998) and applied to image processing by Courboulay and *al.* (Courboulay et al., 2002).

The second section of this article is dealing with the presentation of EPI and with the obtaining of the particular PDE. The third one presents a direct application to the presented diffusion process which may find applicability in robotics and automation. Last part is dedicated to discussion.

2 EPI AND IMAGE DIFFUSION

2.1 EPI

Developed by Frieden, the principle of Extreme Physical Information (EPI) is aimed at defining a new theory of measurement. The key element of this new theory is that it takes into account the effect of an observer on a measurement scenario. As stated by Frieden (Frieden, 1996; Frieden, 1998), ”EPI is an observer-based theory of physics”. By observing, the observer is both a collector of data and an interference that affects the physical phenomenon which produces the data. Although the EPI principle brings new concepts, it still has to rest on the definition of information. Fisher information was chosen for its ability to effectively represent the quality of a measurement. Fisher information measure was introduced by Fisher in 1922 (Fisher, 1922) in the context of statistical estimation. In the last ten years, a growing interest for this information measure has arisen in theoretical physics. In his recent book (Frieden, 1998), Frieden has characterized Fisher information measure as a versatile tool to describe the evolution laws of physical systems; one of his major results is that the classical evolution equations as the Shrodinger wave equation, the Klein-Gordon equation, the Helmotz wave equation, or the diffusion equation, can be derived from the minimization of Fisher information measure under proper constraint.

Practically speaking, EPI principle can be seen as an optimization of the information transfer from the system under measurement to the observer, each one being characterized by a Fisher Information measure denoted respectively I and J. The first one is representative of the quality of the estimation of the data, and the second one allows to take into account the effect of the subjectivity of the observer on the measure. The existence of this transfer leads to create fluctuations on the acquired data compared to the real ones. In fact, this information channel leads to the loss of accuracy on the measure whereas the certainty is increased.

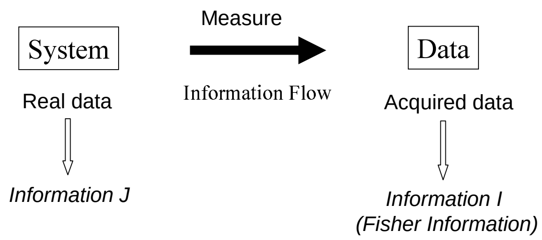


Figure 2: Fisher Information.

The goal of EPI is then to extremize the difference $I-J$ (*i.e.* the uncertainty/inaccurracy compromise) denoted K, called Physical Information of the system, in order to optimized the information flow.

2.2 Application to Image Diffusion

Application to image diffusion can be illustrated by Fig. (3).

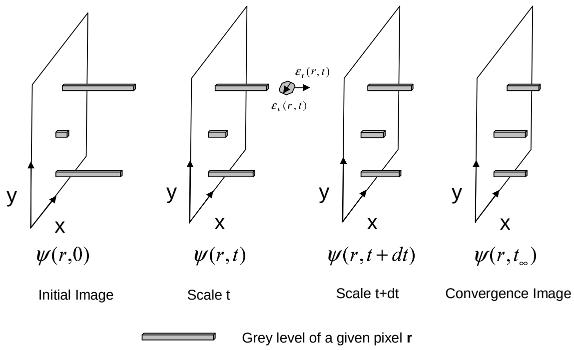


Figure 3: Uncertainty/inaccuracy compromise and isotropic image diffusion. When parameter $t \longrightarrow \infty$, luminance of all pixels of the corresponding image is the same and equal to the spatial average of the initial image.

As far as isotropic image diffusion is concerned, the uncertainty deals with the fluctuations of the grey level of a given pixel compared with its real value, whereas the inaccuracy deals with the fluctuations of the spatial localisation of a given pixel compared with the real one. The two different errors ($\varepsilon_r(t)$ and $\varepsilon_v(t)$) of Fig. (3) which are introduced all along the diffusion process are characterized by a measure of Fisher information. Intrinsic Fisher information J will be an integration of the diffusion constrained we impose on the processing.

Then, we can apply EPI to image diffusion process by considering an image as a measure of characteristics (as luminance, brightness, contrast) of a particular scene, and diffusion as the observer of this measure at a given scale. Extreme Physical Information K is then defined as follows (Frieden, 1998):

$$K(\psi) = \int\int d\Omega dt \times \left[(\nabla - \mathbf{A})(\nabla - \mathbf{A})\psi^2 + (\frac{\partial\psi}{\partial t})^2 - \psi^2 \right], \quad (2)$$

where $\psi(\mathbf{r}, 0) : \mathbb{R}^2 \times \mathbb{R}^+ \to \mathbb{R}$ is the luminance function of the original image and $\mathbf{A}$ a potential vector representing the parameterizable constrain integrated within diffusion process.

Extremizing K by Lagrangian approach leads to a particular diffusion equation given by :

$$\frac{\partial\psi}{\partial t} = \frac{1}{2}(\nabla - \mathbf{A}).(\nabla - \mathbf{A})\psi \quad . \quad (3)$$

As a consequence, thanks to the possible parameterization of $\mathbf{A}$, it is possible to take into account particular characterized pattern to preserve from the diffusion process.

2.3 About A

The $\mathbf{A}$ potential allows to control the diffusion process and introduce some prior constrains during image evolution. For instance, if no constrain are to be taken into account, we set $\mathbf{A}$ as vector null and (Eq. 3) becomes :

$$\frac{\partial\psi}{\partial t} = \nabla.\nabla\psi = \triangle\psi \quad . \quad (4)$$

which is the well known heat equation characterized by an isotropic smoothing of the data processed.

In order to enlarge the possibility given by Eq. (3), the choice we make for $\mathbf{A}$ is based on the fact that Eq. (3) allows a weighting of the diffusion process with the difference of orientation between the local calculated gradient and $\mathbf{A}$. More precisely, to explain the way $\mathbf{A}$ is practically implemented, let consider Fig. 4.

The expression of the local gradient $\nabla\psi$ in terms of θ” is, considering Fig. 4 :

$$\nabla\psi = \begin{pmatrix} \|\nabla\psi\| \cos\theta'' \\ \|\nabla\psi\| \sin\theta'' \end{pmatrix}, \quad (5)$$

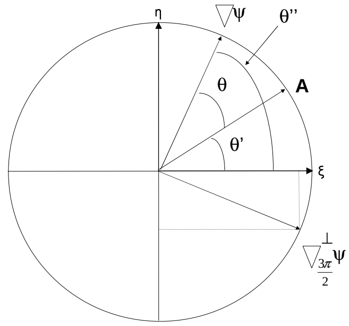


Figure 4: Local geometrical implementation of $\mathbf{A}$ in terms of the local gradient $\nabla\psi$.

and an expression of $\mathbf{A}$ in terms of θ’ is :

$$\mathbf{A} = \begin{pmatrix} \|\nabla\psi\| \cos\theta' \\ \|\nabla\psi\| \sin\theta' \end{pmatrix}. \quad (6)$$

Norm of $\mathbf{A}$ is imposed in order to make it possible the comparison with the gradient. To this point, the most interesting expression of $\mathbf{A}$ would be the one in terms of θ, which represents the difference angle between $\mathbf{A}$ and the local gradient. If we made so, using trigonometrical properties and noticing that $\theta = |\theta'' - \theta'|$, we obtain a new expression for $\mathbf{A}$:

$$\mathbf{A} = \begin{pmatrix} \|\nabla\psi\|(\cos\theta''\cos\theta + \sin\theta''\sin\theta) \\ \|\nabla\psi\|(\sin\theta''\cos\theta - \cos\theta''\sin\theta) \end{pmatrix}. \quad (7)$$

Eq. (7) could be simplified by integrating the vectorial expression of the local gradient (Eq. (5)) :

$$\mathbf{A} = \nabla\psi.\cos\theta + \nabla^{\perp}_{\frac{3\pi}{2}}\psi.\sin\theta\,. \quad (8)$$

From Eq. (8), we could then derive a general expression for $\mathbf{A}$ considering it as a vectorial operator :

$$\mathbf{A} = \nabla.\cos\theta + \nabla^{\perp}_{\frac{3\pi}{2}}.\sin\theta\,, \quad (9)$$

with θ the relative angle between $\mathbf{A}$ et $\nabla\psi$ for a given pixel and $\nabla^{\perp}$ the local vector orthogonal to ∇ (Fig. 4). This expression only represents the way it is possible to reexpress $\mathbf{A}$ by an orthogonal projection in the local base. Considering it, Eq. (3) becomes :

$$\frac{\partial\psi}{\partial t} = \frac{\partial^2\psi}{\partial\eta^2}.(1 - \cos\theta) + \frac{\partial^2\psi}{\partial\xi^2}.(1 - \cos\theta) \quad . \quad (10)$$

One can notice on Eq. (10) that when angle $\theta = 0$ (*i.e.* $\mathbf{A}$ and $\nabla\psi$ are colinear), the studied pixel will not be diffused for $\frac{\partial \psi}{\partial t} = 0$. On the contrary, a non-zero value of θ will lead to a weighted diffusion of the considered neighborhood of the pixel (Eq. (10)).

As a consequence, by imposing local θ values, it is possible to preserve particular patterns from the diffusive effect within the processed image.

3 APPLICATION TO ORIENTED PATTERN EXTRACTION

In this section, we present results obtained on simple images in order to show the restoration and denoising potential of the method.

For practical numerical implementation, the process of Eq. (10) is discretized with a time step τ. The images $\psi(t_n)$ are calculated, with Eq. (10), at discrete instant $t_n = n\tau$ with n the number of iterations in the process.

Let first consider an image showing vertical, horizontal, and 45°-oriented dark stripes on a uniform background (Fig. 5).

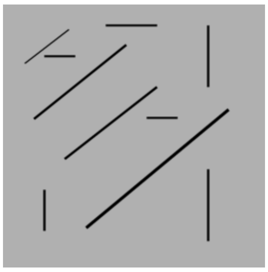


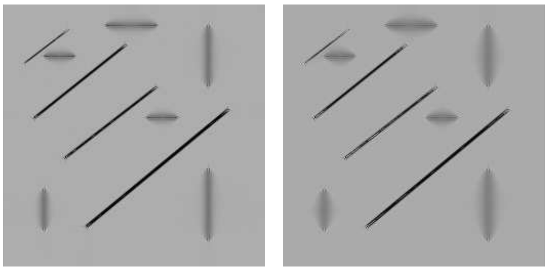
Figure 5: Image 1: Dark stripes with various orientations on a uniform background.

Considering Eq. (10), by imposing two possible orientations for **A** (135°, 325°) which corresponds to the gradient orientations of the diagonal stripes, one could expect to preserve them from isotropic diffusion. Diffusion results are presented Fig. 6.

As one was expected it, the vertical and horizontal dark stripes in diffused images tend to disappear whereas the diagonal stripes are preserved all along the diffusion process.

Let now consider a noisy simple grid diagonally oriented corrupted by a Gaussian noise of standard deviation set to 0.3.

If we apply the same diffusion process of Eq. (10) to this noisy simple grid imposing this time four possible orientations for **A** corresponding to the four possible gradient orientations of the grid, it is then possible to show the denoising effect of the diffusion process (Fig. 8).



(a) (b)

Figure 6: Diffusion of ”Image 1” (Fig. 5) for (a) n=100 and (b) n=200. **A.** is chosen in order to preserve only diagonal stripes from isotropic diffusion process. Time step τ is fixed to 0.2.

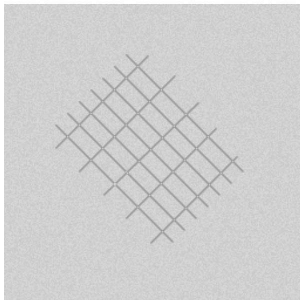
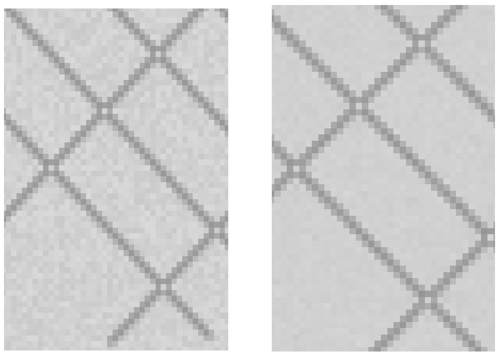


Figure 7: Image 2: Noisy diagonally oriented grid (Gaussian noise). PSNR (calculated with the non corrupted version of the grid as reference) is equal to 68 dB.



(a) (b)

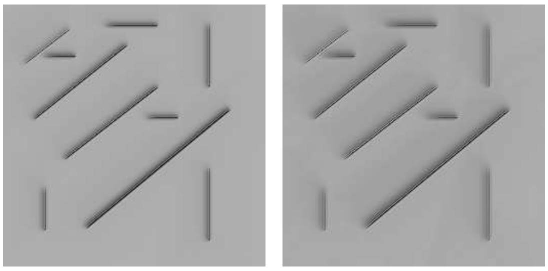
Figure 8: Diffusion of ”Image 2” (a) (Fig. 7) for (b) n=50. As one can notice, the grid itself is preserved from the diffusive effect of Eq. (3) whereas noise is iteration after iteration removed. Time step τ is fixed to 0.2.

As intended, the grid itself is not diffused at all and the increase of the Peak Signal to Noise Ratio (PSNR) from 68 dB to 84 dB, shows that the added Gaussian noise is removed iteration after iteration.

4 DISCUSSION

In this article an original diffusion method, based on the use of a particular PDE (Eq. (3)) derived from EPI theory, has been presented. It has been shown that the integration of the potential vector **A** within the formulation of this PDE makes it possible the integration within the diffusion scheme of particular constrains. This has been assimilated to integration of selectiveness within classical isotropic diffusion process. Examples on ad hoc images have been presented to show the potential of the presented method in the areas of denoising and extraction of oriented patterns.

Applications presented can be discussed, for frequential filterings or Gabor-filters convolution can lead to similar results. Considering that, it is necessary to keep in mind that processed image have been chosen in an ad hoc way to show the potential of the method. Nevertheless, one major difference must be noticed. Let consider again Fig. 5. If **A** is chosen in order to preserve only one direction of the diagonal stripes, implementation of Eq. (3) leads to result presented Fig. 9.



(a) (b)

Figure 9: Diffusion of ”Image 1” (Fig. 5) for (a) n=20 and (b) n=50. As one can notice, Eq. (9) makes it possible to only preserve one gradient direction of the diagonal stripes. Time step τ is fixed to 0.2.

That kind of results can not be obtained by classical methods and enlarge the possible applications of Eq. (3).

As a conlusion, an alternative method for oriented pattern extraction has been presented in this article. It has been demonstrated, as a proof a feasibility, on ad hoc images that the developed approach may find applicability in robotics and visions as far extraction of oriented pattern is still an open problem. Industrial control quality check can also be an other area of applications.

REFERENCES

Alvarez, L., Guichard, F., Lions, P., and Morel, J. (1992). Image selective smoothing and edge detection by nonlinear diffusion (ii). *Arch. Rationnal Mech. Anal.*, 29(3):845–866.

Catté, F., Coll, T., Lions, P., and Morel, J. (1992). Image selective smoothing and edge detection by nonlinear diffusion. *SIAM Journal of Applied Mathematics*, 29(1):182–193.

Courboulay, V., Ménard, M., Eboueya, M., and Courtellemont, P. (2002). Une nouvelle approche du filtrage linéaire optimal dans le cadre de l'information physique extreme. In *RFIA 2002*, pages 897–905.

Deriche, R. and Faugeras, O. (1996). Les edp en traitements des images et visions par ordinateur. *Traitement du Signal*, 13(6):551–578.

Fisher, R. (1922). *Philosophical Transactions of the Royal Society of London*, 222:309.

Frieden, B. (1996). Fisher information as a measure of time. *Astrophysics and Space Sciences*, 244:387–391.

Frieden, B. (1998). *Physics from Fisher Information*. Cambridge University Press.

Geman, S. and Reynolds, G. (1992). Constrained restoration and the recovery of discontinuities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(3):367–383.

Nitzberg, M. and Shiota, T. (1992). Nonlinear image filtering with edge and corner enhancement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(8):826–833.

Nordstrom, N. (1990). Biased anisotropic diffusion-a unified regularization and diffusion approach to edge detection. *Image and Vision Computing*, 8(4):318–327.

Perona, P. and Malik, J. (1990). Scale-space and edge detection using anistropic diffusion. *IEEE Transcations on Pattern Analysis and Machine Intelligence*, 12(7):629–639.

Terebes, R., Lavialle, O., Baylou, P., and Borda, M. (2002). Mixed anisotropic diffusion. In *Proceedings of the 16th International Conference on Pattern Recognition*, volume 3, pages 1051–4651.

Tschumperle, D. and Deriche, R. (2002). Diffusion pdes on vector-valued images. *Signal Processing Magazine, IEEE*, 19:16–25.

Tschumperle, D. and Deriche, R. (2005). Vector-valued image regularization with pde's: A common framework for different applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:506–517.

Weickert, J. (1995). Multiscale texture enhancement. In *Computer Analysis of Images and Patterns*, pages 230–237.

Weickert, J. (1998). *Anisotropic Diffusion in image processing*. Teubner-Verlag, Stuttgart.

Whitaker, R. and Pizer, S. (1993). A multi-scale approach to nonuniform diffusion. *CVGIP:Image Understanding*, 57(1):99–110.

EXTENSION OF THE GENERALIZED IMAGE RECTIFICATION
Catching the Infinity Cases

Klaus Häming and Gabriele Peters
LSVII, University of Dortmund, Otto-Hahn-Str. 16, D-44227 Dortmund, Germany
haeming@ls7.cs.uni-dortmund.de, peters@ls7.cs.uni-dortmund.de

Keywords: Stereo vision, epipolar geometry, rectification, image processing, object acquisition.

Abstract: This paper addresses the topic of image rectification, a widely used technique in 3D-reconstruction and stereo vision. The most popular algorithm uses a projective transformation to map the epipoles of the images to infinity. This algorithm fails whenever an epipole lies inside an image. To overcome this drawback, a rectification scheme known as polar rectification can be used. This, however, fails whenever an epipole lies at infinity.
For autonomous systems exploring their environment, it can happen that successive camera positions constitute cases where we have an image pair with one epipole at infinity and the other inside an image. So neither of the previous algorithms can be applied directly. We present an extension to the polar rectification scheme. This extension allows the rectification of image pairs whose epipoles lie even at such difficult positions. Additionally, we discuss the necessary computation of the orientation of the epipolar geometry in terms of the fundamental matrix directly, avoiding the computation of a line homography as in the original polar rectification process.

1 INTRODUCTION

A common problem in all stereo vision tasks is the correspondence problem. To simplify this search for image structures representing the same world structure, images are usually rectified. The result is a pair of images where corresponding points lie on the same horizontal line, this way limiting the search region.

How this process is carried out in a given case depends on the camera geometry. The epipoles are points in the image plane induced by this geometry. These points are the images of the camera centers, i.e., the first epipole is the projection of the second camera center onto the fist image plane and the second epipole is the same for the first camera center. In the original, non-rectified images, the corresponding lines mentioned above all meet at the epipoles. So, epipoles and their position play an important role in image rectification.

In most stereo vision tasks the camera centers have a distance of a few centimeters and the cameras are pointing to a common point in front of them. In those cases the epipoles neither lie inside an image nor at infinity. This makes things easy and the traditional way of rectification through image homography as described in popular textbooks such as (Hartley and Zisserman, 2004) might be applied.

As our interest lies in object recognition and object learning, including autonomous movements of a camera guiding system (Peters, 2006), successive camera positions are more or less unpredictable. This means those special cases mentioned above can even occur in combination, as shown in Figure 1.

For epipoles inside the image boundaries, the approach of polar rectification exists (Pollefeys et al., 1999). Inside an image, the epipole divides each epipolar line into two half-lines, thus limiting the search region not only to epipolar lines, but to epipolar *half*-lines. To exploit this advantage, the epipolar geometry has to be oriented. In Pollefeys' approach, the orientation is carried out using a line homography computed from the fundamental matrix or from the camera projection matrices. Our approach uses the fundamental matrix directly. This is described in section 2.

The process of resampling the images to produce

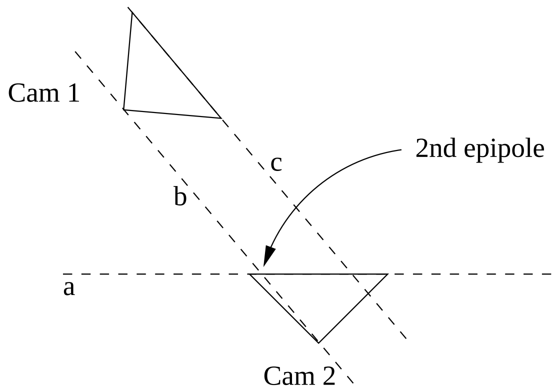


Figure 1: A difficult camera constellation for image rectification. Line b connects both camera centers. The second epipole lies at the position, where this line meets the image plane a of "Cam 2". In this case, this is inside the second image. On the other hand, the first epipole lies at the position where b meets c, the image plane of "Cam 1". This is, however, at infinity.

a rectified image pair is the topic of Section 3. First, in Section 3.1, the procedure for epipoles at a finite position will be reviewed shortly, then, in Section 3.2, our extension for the case of infinity is presented.

As two rectification techniques already exist—on the one hand polar rectification, which can handle epipoles inside the image and on the other hand rectification through image homography, which easily handles epipoles at infinity—one can argue it may be sufficient to switch to the appropriate one for the given case. This will, unfortunately, not cover the case in which one epipole lies inside and the other at infinity. This case can easily occur as already shown in Figure 1. To solve this problem, our extension of the polar rectification method will now be discussed.

2 ORIENTATION

The polar rectification samples the epipolar lines one by one and puts them one after another into a new image. If the epipole lies inside the image, it will divide the epipolar lines into two half-lines. That means, we do not only have to match the correct epipolar lines to each other, but the correct half of them. Otherwise we have to search the whole epipolar line where half the effort would suffice. Now the question is how to tell which halfs belong to each other. To solve this, we first have to orient the epipolar geometry. Unlike Pollefeys in (Pollefeys et al., 1999) we do not compute the line homography but use the fundamental matrix directly.

The fundamental matrix, denoted by F, describes the relationship between two images and their cameras known as the epipolar geometry. Let x' be a point in the first view, then

$$l'' = Fx' \tag{1}$$

is the the line in the second view on which any corresponding point of x' must lie. Similarly, there exists a relation

$$l' = F^T x'' \tag{2}$$

between a point x'' in the second view and a line l' in the first view. These lines are 3-vectors of the coefficients of the equation of an implicit line in two dimensions:

$$l'_0 * x + l'_1 * y + l'_2 = 0 \tag{3}$$

Therefore each line divides the plane into a positive and a negative region. This can be used to orient the epipolar geometry.

Usually "orient the epipolar geometry" means to ensure that every world point seen by one of the cameras lies *in front* of this very camera. The usefulness of oriented epipolar geometry for computer vision was first described by Stéphane Laveau in (Laveau and Faugeras, 1996).

However, what we like to know is on which side of their epipolar lines with respect to the epipoles lie two corresponding points.

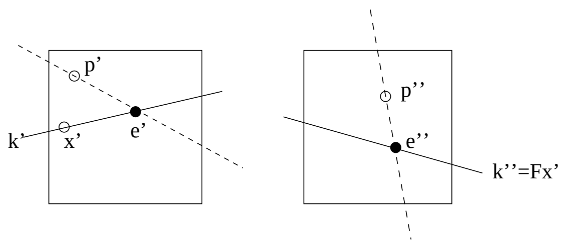


Figure 2: A visualization of the orientation process. The symbols correspond to those in Equation 4 and 5. Essentially, after orientation, the sign of the fundamental matrix F is modified in such a way that the regions in which p' and p'' lie have the same sign with respect to the lines k' and k''.

To answer our question, we use equations 1 and 2 and a pair of points initially known to constitute a correspondence. As in Figure 2, the point pair is denoted by p' and p'' for the point in the first and second view, respectively. We use an auxiliary line

$$k' = x' \times e' \tag{4}$$

where x' is an arbitrary point (with the exception that it must not lie on the line $p' \times e'$) and e' is the first epipole. Then we choose the sign s in such a way that

$$\mathtt{sign}(k'p') = s \cdot \mathtt{sign}((Fx')p'') \tag{5}$$

where Fx' is k'', the line in the second image corresponding to k'. Having s computed, we can orient our fundamental matrix

$$F^o = sF \tag{6}$$

Having the oriented fundamental matrix F^o, the correct half-line is easily determined. Suppose there is a point q' in the first image and the appropriate line in the second image is $t'' = Fq'$. For each candidate q'' on the correct half-line of t''

$$\mathtt{sign}((e' \times p') \cdot q') = \mathtt{sign}((F^o p') \cdot q'') \tag{7}$$

must hold true. Thus, we have limited the search region from epipolar lines to epipolar half-lines.

When using polar rectification, make sure all quantities used in equation 7 retain their sign throughout the process. Otherwise the result will be disappointing even if F is oriented first.

3 RECTIFICATION

In this section, we will first review the re-sampling process via polar rectification as described by Pollefeys. Then, we will introduce our proposed extension for epipoles at infinity.

3.1 Resampling the Image

The main idea is to sample the image one epipolar line after another. This leads to the question about an appropriate step size between successive epipolar lines. The main criterion is to avoid a too coarse sampling, meaning a loss of information contained in the image.

Figure 3 visualizes how to determine a good distance between successive epipolar lines. It shows two congruent right-angled triangles defined by the point triplets abc and $a'b'c'$. Both have the point $b = b'$ in common. Now, the goal is to ensure that the distance $|c'a'|$ is at least one pixel.

Exploiting the observed congruency, the distance $|c'b'|$ along the image border can be computed as the ratio

$$|c'b'| = \frac{|bc|}{|ac|} \tag{8}$$

The same is done in the second image. The resulting point is transferred into the first image by applying equation 2 and intersecting the resulting line with the image border. Point b'' denotes the result. If $|c'b'| < |c'b''|$, the step size of the first image is used, otherwise the step size of the second image is used.

Obviously, if one of the epipoles approaches infinity, this procedure will fail because in equation 8 both numerator and denominator also approach infinity.

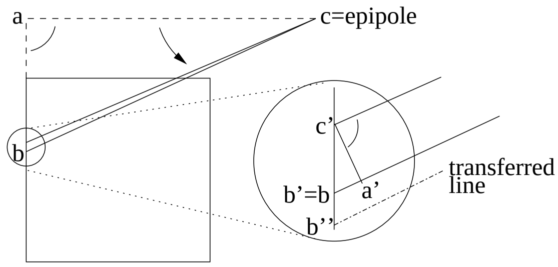


Figure 3: Resampling of an image using polar rectification. The two congruent right-angled triangles abc and $a'b'c'$ are shown. The line computed from the step size in the second image is depicted as "transferred line". Its intersection with the image border is b''.

3.2 Resampling an Image with its Epipole at Infinity

When the epipole approaches infinity, the left side of equation 8 approximates one, which is our step size for this case. The third co-ordinate of the epipole equals or gets close to zero and the first two coordinates form a 2D-vector pointing to the epipole. This allows us to compute the perpendicular direction and sample the image along this direction in parallel lines using unit steps. This is shown in Figure 4. Of course, this unit step again is compared to the step size of the other image to avoid loss of information.

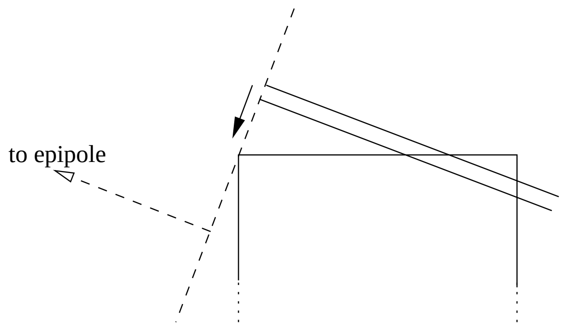


Figure 4: If the epipole lies at infinity, the sampling can easily be done scanning the image line by line perpendicular to the direction of the epipole. The line distances are at most one pixel.

To decide whether an epipole lies at infinity or not, a threshold on the distance of the epipole to the image jumps out as the natural parameter of choice. This distance threshold is now denoted by d, the epipole by e. Figure 5 shows how to compute a satisfying value of d. The width of the image is w, the height is h.

To simplify, we assume $e = (e_0, 0, 1)^T$, meaning $f = (0, 1, 0)^T$ is an epipolar line. Thus, we can com-

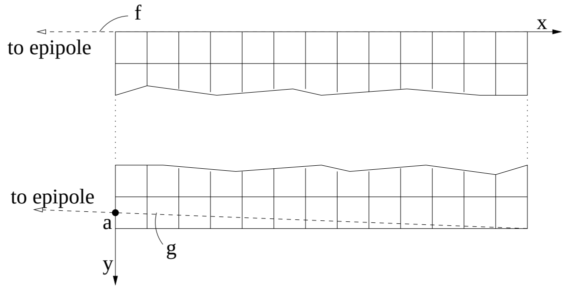


Figure 5: Finding a distance threshold d. A usable value is computed as the horizontal distance of the epipole to the image. The epipole in turn lies at the intersection of the epipolar lines g and h.

pute $|e_0|$ as an appropriate value for d.

In our simplified consideration, we look at the epipolar lines f and g, the latter intersecting the lower right corner. The epipole may be assumed to lie near infinity whenever these epipolar lines run parallel "enough". g also intersects the left image border in a point denoted by a. The distance of point a to the lower left image corner can be used to compute the value of e_0.

Choosing a distance of $\frac{1}{2}$ pixel from a to the lower left corner, we get $a = (0, h - \frac{1}{2}, 1)^T$. Because the left image border is $(1,0,0)^T$ and $g = (e \times (w,h,1)^T)$, a can also be computed as $a = (1,0,0)^T \times (e \times (w,h,1)^T)$. This yields $e_0 = w - 2wh$.

For an image with $w = h = 1000$, the epipole has an x-coordinate of -1 999 000. For such an image a distance threshold of more than 2 000 000 would therefore be sufficient.

Once d is computed (or chosen), let $\varepsilon = |\frac{1}{d}|$. Then, a usable rule to decide when to switch to sampling with parallel lines looks like:

$$\begin{array}{lcl} \text{epipole} & & (|e_2| < \varepsilon) \texttt{ OR} \\ \text{is at} & \Leftrightarrow & (|e_0| > 0 \texttt{ AND } (|\frac{e_2}{e_0}| < \varepsilon)) \texttt{ OR} \\ \text{infinity} & & (|e_1| > 0 \texttt{ AND } (|\frac{e_2}{e_1}| < \varepsilon)) \end{array} \quad (9)$$

It is advantageous to first compute the point pairs on the image borders where in one of the images a corner is met during rectification. Between two consecutive pairs, the whole process is merely a simple loop of repeatedly determining the optimum step size and sampling the images.

4 RESULTS

We examine our method proposed in section 3 with two stereo pair examples shown in Figures 6 and 7[1].

[1]They show the freely available VRML model "Al" from different camera positions.

For both examples the rectified images are shown below the original images. It can easily be recognized that corresponding features now lie on the same line which is, after all, what rectification is all about.

The initial point correspondence needed to orient the epipolar geometry was not obtained through feature matching, but by intersecting the known viewing pyramids of the cameras and choosing the closest point in a decent distance to both camera centers.

The first example covers the case of one epipole outside the image at a finite position and the second epipole lying at infinity (Figure 6). In the second example we examine the case of the first epipole lying inside the image and the second one at infinity (Figure 7). To show the effect of rotated cameras, which is likely to occur in an autonomous system, the second camera of the latter example was rotated 45 degrees. The figures are arranged to have the same pixel size in all for sub-images.

Owing to the calculation of the optimum step size during sampling, a stretching effect is noticeable. For example, in Figure 6, sub-image a) is sampled along the left image border in steps less than one pixel to ensure the distance $|c'a'|$ of Figure 4 being one pixel. To match this step size, the right figure was in turn sampled in steps less than one pixel, thus stretched.

Figure 7 shows the same stretching effect for similar reasons. Additionally, the rectified sub-image d) takes a diamond form. This happens because the second camera was rotated 45 degrees along the camera direction. So, the sampling occurred along a diagonal line. The rectified sub-image c) of Figure 7 shows what happens if an epipole lies inside an image. The half-lines are sampled in such a way that their beginning, which is the epipole, always is placed on the left side of the rectified image.

Summarized, rectified images such as those shown in figures 6 and 7 (parts c) and d)) provide the possibility of a fast calculation of correspondences between their source images (such as those shown in parts a) and b)). The former method of polar rectification would fail to produce these rectified images in both examples as one epipole of the source images is positioned at infinity. Using an image homography would succeed in the first example but fail in the second one, because parts of the rectified image would be mapped to infinity, as one of the epipoles lies inside the image.

5 CONCLUSION

An extension of the polar rectification process was presented, covering the special cases where the

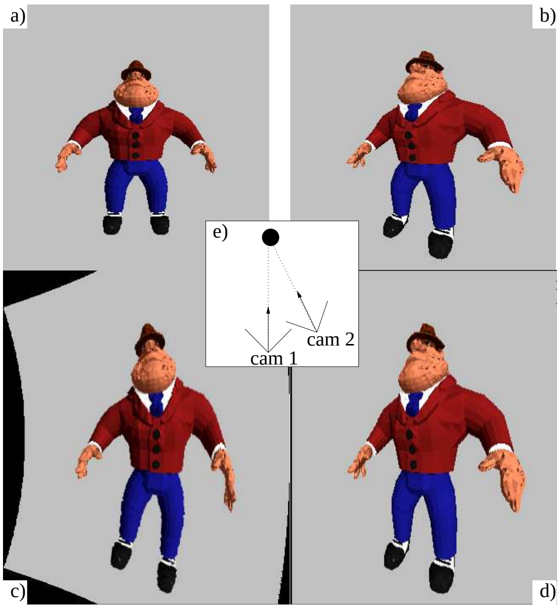


Figure 6: One epipole at infinity. a) First image recorded with camera 1 as shown in e). Its epipole lies outside the image at a finite position. b) Second image recorded with camera 2 as shown in e). Its epipole lies at infinity. c) and d) are rectified versions of images a) and b), respectively, which have been generated with the proposed method.

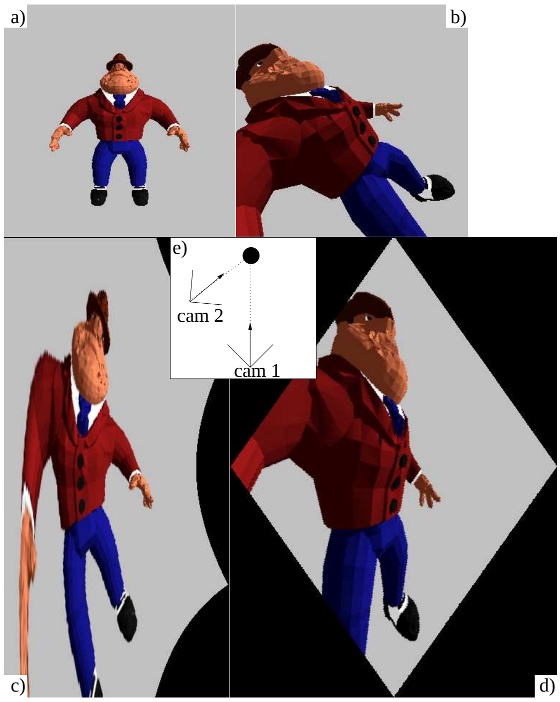


Figure 7: One epipole inside the image, one at infinity. a) First image recorded with camera 1 as shown in e). Its epipole lies inside the image. b) Second image recorded with camera 2 as shown in e). Its epipole lies at infinity. c) and d) are rectified versions of images a) and b), respectively, generated with the proposed method.

epipole of at least one image to be rectified lies at infinity. Additionally, the technique of line transfer used in the original paper of Pollefeys was substituted by the use of the fundamental matrix alone.

Given the proposed extension of the rectification process, it is now possible to deal with general camera positions, where former methods failed in special cases. As even extreme camera positions of an acquisition system can be evaluated now, e.g., for 3D reconstruction in an object acquisition system, this opens new possibilities for more flexible autonomous systems, where successive camera positions are unpredictable.

ACKNOWLEDGEMENTS

This research was funded by the German Research Association (DFG) under Grant PE 887/3-1.

REFERENCES

Hartley, R. I. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition.

Laveau, S. and Faugeras, O. D. (1996). Oriented projective geometry for computer vision. In *ECCV '96: Proceedings of the 4th European Conference on Computer Vision-Volume I*, pages 147–156, London, UK. Springer.

Peters, G. (2006). A Vision System for Interactive Object Learning. In *IEEE International Conference on Computer Vision Systems (ICVS 2006)*, New York, USA.

Pollefeys, M., Koch, R., and van Gool, L. J. (1999). A simple and efficient rectification method for general motion. In *Proc. International Conference on Computer Vision (ICCV 1999)*, pages 496–501.

ARTIFICIAL IMMUNE FILTER FOR VISUAL TRACKING

Alejandro Carrasco E. and Peter Goldsmith
Mechanical Enginering, University of Calgary, 2500 University Drive NW, Calgary, Canada
ecarrasc@ucalgary.ca, peter.goldsmith@ucalgary.ca

Keywords: Filtering, visual tracking, immune system, artificial intelligence.

Abstract: Visual tracking is an important part of artificial Vision for robotics. It allows robots to move towards a desired position using real world information. In this paper we present a novel particle filtering method for visual tracking, based on a clonal selection and a somatic mutation processes used by the natural immune system, which is excellent at identifying intrusion cells; antigens. This capability is used in this work to track motion of the object in a sequence of images.

1 INTRODUCTION

Artificial intelligence has found a source of ideas borrowed from biological systems such as swarms, ant colonies, neural networks, genetic algorithms and immune systems. They have been successfully used in many different areas: control (Macnab, 2000), optimization (Charbonneau, 2002), pattern recognition (Tashima, 2001), robotics (Ramirez-Serrano, 2004) and prediction (Connor, 1994). The immune system is composed of a complex constellation of cells, organs and tissues, arranged in an elaborate and dynamic communications network and equipped to optimize the response against invasion by pathogenic organisms. The immune system is, in it simplest form, a cascade of detection and adaptation culminating in a system that is remarkably effective, most of the time. It has many facets, a number of which can change to optimize the response to these unwanted intrusions (Dasgupta, 2002). The immune system has a series of dual natures, the most important of which is self - non-self recognition. The others are: general - specific, natural - adaptive, innate - acquired, cell_mediated - humoral, active – passive and primary - secondary. Parts of the immune system are antigen-specific (they recognize and act against particular antigens), systemic (not confined to the initial infection site, but work throughout the body), and have memory (recognize and mount an even stronger attack to the same antigen the next time) (Gilbert, C. , 1994). It can recognize and remember millions of different enemies, and it can produce secretions and cells to match up with and wipe out each one of them. The secret to its success is an elaborate and dynamic communications network (de Castro, 2002). Millions and millions of cells, organized into sets and subsets, gather like clouds of bees swarming around a hive and pass information back and forth. The key to a healthy immune system is its remarkable ability to distinguish between the body’s own cells and foreign cells (Bergstrom, 2004). The body’s immune defences normally coexist peacefully with cells that carry distinctive “self” marker molecules. But when immune defenders encounter cells or organisms carrying markers that say “foreign,” they quickly launch an attack. In this work, we use the intruder detection capability of artificial immune systems in order to track the object in a sequence of images.

2 VISUAL TRACKING

Visual tracking is the action of consistently locating a desired feature in each image of an input sequence. The problem is typically complicated by sensor noise, motion in the scene, motion on the part of the observer and real-time constraints. The problem can be further complicated when more than one identical feature must be tracked, in which case it is up to the observer to decide the optimal set of correspondences which are consistent with a priori assumptions about, and recent observations of, the behavioural characteristics of the features (Prassler, 1990)(Carlsson, 1990). Given an image $I(i, j) i, j \in \aleph^+$, the problem is to track a sub-image (object). In a sequence of images the object will be in different positions, moving in a determined pattern. Therefore the prediction part of the filter is needed to predict where the object *I(u,v)* will be in the image *I(i,j)*, giving a region of interest to accelerate the processing of recognizing the object. Recognizing the object by filtering the clutter and noise due to change of illumination, shadows, etc. is the second part of the filter. The use of filters such as the Kalman filter (Gutman, 1990)(Welch, 2001), which is based in optimal prediction for linear system and noise with Gaussian distribution, are excellent tools to overcome the problems in visual tracking. Extensions of the Kalman filter for non-linear systems have been developed such as Extended Kalman filter (Ribeiro, 2004) and Unscented Kalman filter (Jeffrey, 1997). Another algorithm of interest is the condensation (Conditional Density Propagation) (Isard, 1998), which is based on computing the Bayes' rule to a set of particles (particle filtering). In general the filters mentioned above can be seen as Bayesian filters, where the following density distributions are needed (Isard, 1996) (Grewal, 1993):

$p(x_k \mid Z_k)$: A posteriori density given the measurement.
$p(x_k \mid Z_{k-1})$: A priori density.
$p(x_x \mid x_{k-1})$: Process density describing the dynamics.
$p(z_k \mid x_k)$: Observation density

Bayes' Rule is

$$p(x_k \mid Z_k) = \frac{p(z_k \mid x_k) \int p(x_k \mid x_{k-1}) p(x_{k-1} \mid Z_{k-1}) dx_k}{\int p(z_k \mid x_k) p(x_k \mid Z_{k-1}) dx_k} \qquad (1)$$

One of the drawbacks in these algorithms is the assumption of priori density distribution, Gaussian distribution such in the case of Kalman filter. Particle filters use Bayes (equation 1) and Monte Carlo method to approximate the sequence of probability distribution; these required a large number of particles to converge towards the probability distribution. Therefore, the random sampling is the main drawback, due to in case that the population is not drawn to represent some of its statistical features makes a wrong estimation. Besides, due to the degeneration of the particles through time, re-sampling mechanisms are used. In the next section we introduce an artificial immune system to filter noisy signals and predict the state of a system.

3 ARTIFICIAL IMMUNE FILTER (AIF): CLONAL SELECTION AND SOMATIC MUTATION

The clonal selection theory, by immunologist Frank Macfarlane Burnet (Burnet, 1978), models the principles of an immune system. When an antigen is present in our body, the **B**-Lymphocyte cells produce antibodies **Ab** receptors. Each **B** cell has a specific antibody as a cell surface receptor. The arrangement and generation of antibody genes occurs prior to any exposure to antigen. When a soluble antigen is present, it binds to the antibody on the surface of **B** cells that have the correct specificity. These **B** cell clones develop into antibody-producing plasma cells or memory cells. Only **B** cells, which are antigen-specific, are capable of secreting antibodies. Memory cells remain in greater numbers than the initial **B** cells, allowing the body to quickly respond to a second exposure of that antigen, as show in Figure 1 (de Castro, 2002).

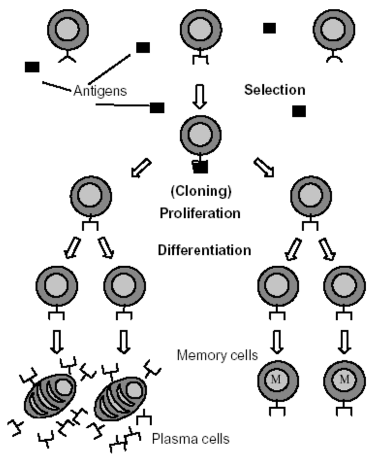


Figure 1: Clonal Selection Principle.

The higher affinity comes from a mechanism that alters the variable regions of the memory cells by specific somatic mutation. This is a random process that by chance can improve antigen binding. This same principle is the inspiration in this work to produce an artificial immune filter. Initial set of *n* **B**-cells (particles) $X = (x^1, x^1, ..., x^n)$, representing the features of our object to track (positions, velocity, etc), weights $W = (w^1, w^2, ..., w^n)$, representing its affinity between the antigens and the antibodies, and memory cells $S = (s^1, s^2, ..., s^m)$, are created. In the beginning our best affine cell to our antigen is our initial condition. Therefore we clone and slightly mutate the cell, using equation (2)

$$x_k^i = x_k^{best} + \alpha r_k^i \quad (2)$$

where r is a random variable normally distributed $r \sim N(0,1)$ and $\alpha \in \Re$ is a small constant. The affinity w_i is integrated by two distance measurements from our best B cell, before and after prediction. Equation (3) is the first part of affinity

$$af_1^i = \exp\left(-\left\|x_k^{best} - x_k^i\right\|\right) \quad (3)$$

The next step $k+1$ is the prediction part, given by $x_{k+1}^i = f(x_k^i)$ for nonlinear dynamics and by $x_{k+1}^i = Ax_k^i$ for linear dynamics, where A is known as the transition matrix. After all the cells have been through the dynamic system, it is time to obtain a new measurement z_k, which contains a certain level of noise. Then we apply equation (4) to obtain the second part of our affinity measurement, where H is the observation model in the case of a linear system and β is a constant.

$$af_2^i = \exp\left(-\beta\left\|z_k - Hx_{k+1}^i\right\|\right) \quad (4)$$

$$w_i = af_1^i + af_2^i \quad (5)$$

Equation (5) calculates the affinity of each **B**-cell to the antigen. The m best cells with high affinity will conform to our memory cells, and the highest affinity will be the estimation $\bar{x}_{k+1}$ and our next best **B**-cell x_{k+1}^{best}.

3.1 Application of Artificial Immune Filter to Noise Rejection

Before applying the artificial immune system to visual tracking, the filter was tested on a noisy signal and compared to a Kalman filter. The signal represents the antigen to be recognized. The best **B**-cell that binds the antigen is the estimation of the state of the signal. The next stage is choosing the parameter for mutation, α. Since the level of somatic mutation for the cells is a slight change on our best **B**-cell, a value equal or less than dt value, the step time of the system, is a good option, because it indicates that **B**-cells could vary $\pm dt$ (0.01 for this example) from their real values.

Given a linear stochastic difference equation in the next form

$$x_{k+1} = Ax_k + bu_k + w_k \quad (6)$$

$$z_k = Hx_k + v_k \quad (7)$$

where

$$A = \begin{bmatrix} 1 & dt \\ -dt & 1 \end{bmatrix} \quad b = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad H = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad u_k = 0$$

Noise is modelled by

$$w_k = \begin{bmatrix} 0.15 \cdot dt^2 \cdot r_k \\ 0.15 \cdot dt \cdot r_k \end{bmatrix} \quad (8)$$

$$v_k = \left[0.1 + 0.05\sinh(r_k)\right] \quad (9)$$

Equation (9) introduce a heavily spike noise with non zero mean, while equation (8) is a normal distribution, and r is random noise. Figure 2 shows the measured position with noise up to 50% of its maximum value.

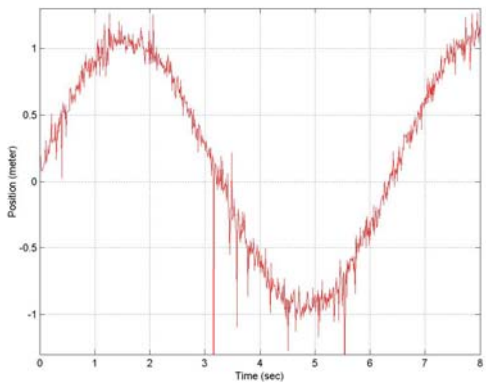


Figure 2: Measured position.

Using the proposed algorithm of Figure 3, we obtained the filtered signal in Figure 4.

```
    xbest=Xinitial_condition
    S = {φ}
1. Clonal Selection of B-cells and Somatic Mutation
    For i=1 to n
        xi=xbest+ α
          if S ≠ {φ}
            Replace xn-m =S(1..m)
            xi=xi+ α r
          endif
        af1=dist(xbest,xi)
    endfor

2. Prediction and affinity with measurement
    xk+1=f(xk)
    af2=dist(Z,Xk+1)

3. Total affinity and Selection of m Memory Cells
    W=af1+ af2
    S=Sort(W) from bigger to small
    Choose list of m cells S(1..m)
    Xbest=S(1)

4. Go to step 1
```

Figure 3: Pseudo-code for Artificial Immune Filter.

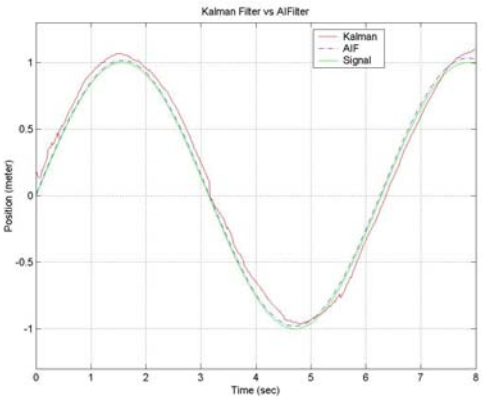


Figure 4: Performance of Filters.

It is well known that the uncertainty of the covariance parameters of the process noise, Q, and the observation errors, R, has a significant impact on Kalman filtering performance. Q and R influence the weight that the filter applies between the existing process information and the latest measurements. Errors in any of them may result in the filter being suboptimal or even cause it to diverge. The conventional way of determining Q and R requires good a priori knowledge of the process noises and measurement errors, which normally comes from intensive empirical analysis. Besides of the errors due to covariance parameters, the Kalman filter is based on the assumption of normal distribution noise with zero mean. Figure 4 shows the real signal with no noise and the filtered signal. It can be seen that the filter affectively attenuated the noise. In this example we use the following parameter settings for the Kalman filter,

$$Q = \begin{bmatrix} 0.00015 & 0.015 \\ 0.015 & 0.00015 \end{bmatrix} \qquad R = 0.0025 \qquad P = Q$$

The parameter settings for the AIF were, *n=20, m=5,* $\alpha = [0.001 \quad 0.01 \quad 0.01]$, $\beta = 0.1$.

3.2 Visual Tracking using AIF

Tracking an object in a sequence of images is a challenging problem. An elementary tracking approach could be to fit a curve, (contour of an object) to each image in a sequence, and an estimated curve is therefore required for each image. Then a fitted curve from one image is the estimation for the next image. This kind of algorithm will be affected by fast motion and become sensitive to distractions. Clutter in the background, either static or dynamic, noise of the sensor and change of illumination, are some factors to consider as noise in an image (Healey, 1994). The tracking performance can be greatly improved by a filter able to predict and correct the fitted curve, removing the noise from the image. Our artificial immune filter is used in this section to track an object in a sequence of images. The extension of the artificial immune filter from single variable to multivariable is straightforward. The contour of the object is a parametric curve

$$c(t) = (Ix(t), Iy(t)) \qquad t \in [0, L] \qquad (10)$$

where t is an independent parameter over the interval *[0,L]*, and *Ix(t)* and *Iy(t)* are known as spline functions (Foley, 1990). An important aspect to achieve real time tracking performance has been the restriction of measurements of the set of observations Z to a sparse set of lines normal to the contour of the object, as shown in Figure 5. In this case the affinity is given by

$$af_2^i = \sum_{j=1}^{P} \exp\left(-\left\|z_k^j - C_k^i(t)\right\|\right) \qquad (11)$$

where P is the number of searching lines and z_k^j is the edge closest to the hypothetical contour $C_k^i(t)$.

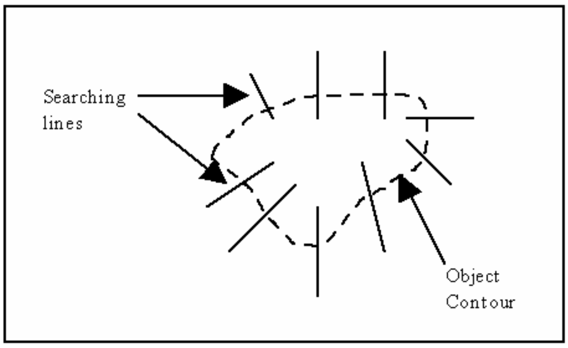


Figure 5: Normal lines of object contour to search the observation z_i.

Figure 6 shows a sequence for fast tracking motion of a ball with clutter added to background. This experiment used 100 cells and 10 memory cells, in real time (30 frames per second). In spite of the fast motion of the ball, the tracker never loses contact with the ball in a sequence of image, when we bounced the ball several times against the wall. The tracker shows the center of the ball with white dots.

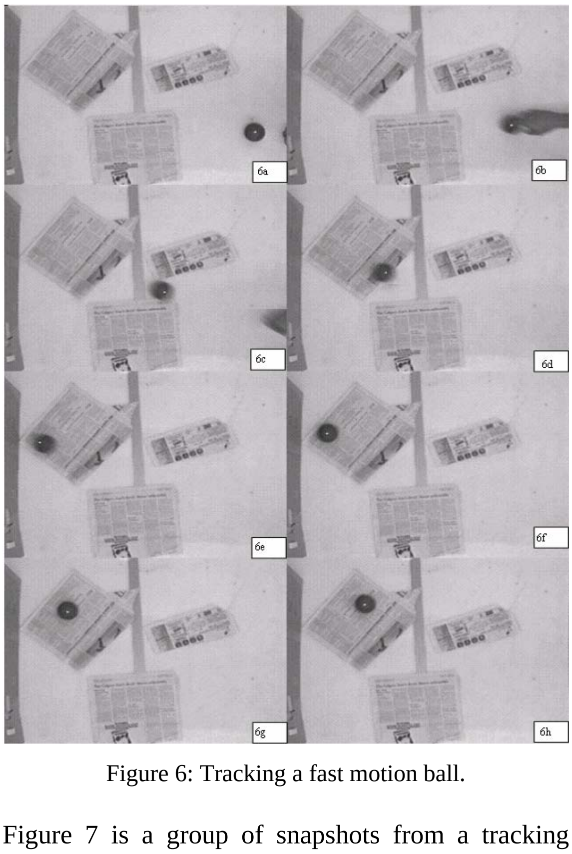


Figure 6: Tracking a fast motion ball.

Figure 7 is a group of snapshots from a tracking sequence of the ball under heavy clutter, dynamic background and partial occlusion.

Figure 7: Tracking using Artificial Immune Filter.

4 CONCLUSIONS

In this work we introduced a novel filter using a clonal selection and somatic mutation model of immune system. The filter does not require probability distributions or re-sampling, unlike other particle filters. The artificial immune filter was tested for signal processing and visual tracking, showing good performance in both applications. In the application of visual tracking of the ball, the filter was able to track fast ball motion in a non-smooth trajectory (bouncing) and clutter in the background. Future work will include the adaptation of parameters and tracking of several objects.

ACKNOWLEDGEMENTS

We would like to thanks to CONACyT for its support in this research.

REFERENCES

Macnab, C.J.B., D'Eleuterio, G.M.T., May 2000, *Discrete-time Lyapunov design for neuro-adaptive control of elastic-joint robots*, International Journal of Robotics Research vol. 19, no. 5, pp. 511-525.

Paul Charbonneau, 2002, *An Introduction to Genetic Algorithms for Numerical Optimization*, National Center for Atmospheric Research, Boulder, Colorado.

Koichi Tashima, Zheng Tang, Okihiko Ishizuka, Koichi Tanno, 2001, *An immune network with interactions* *between B cells for pattern recognition*. Scripta Technica, Syst Comp Jpn, 32(10): 31-41.

Ramirez-Serrano, A., and Pettinaro G.C., 2004, *"Origami robotics: swarm robots that can fold into diverse configurations"*, Mechatronics2004, pp. 171-182, Ankara, Turkey, August 30 - September 1.

Jerome T. Connor, R. Douglas Martin, 1994, *Recurrent Neural Networks and Robust Time Series Prediction*. IEEE Transactions on Neural Networks, Vol. 5 , No. 2.

Dasgupta, D., Gonzalez, F., 2002, *An immunity-based technique to characterize intrusions in computer networks*, IEEE Trans. Evol. Comput. 6:1081–1088.

Gibert, C. J. & Routen, T. W. 1994, *Associative memory in an immune-based system*. In: Proceedings of the Twelfth National Conference on Artificial Intelligence pp. 852–857, Cambridge, MA. AAAI Press/The MIT Press.

de Castro, L. N. and Von Zuben, F. J., 2002, *Learning and optimization using the clonal selection principle*, IEEE Trans. Evol. Comput. Special issue on artificial immune systems, 6:239–251.

Carl T. Bergstrom, Rustom Antia, 2004, *How do adaptive immune systems control pathogens while avoiding autoimmunity?* Department of Biology, University of Washington.

E. Prassler, J. Scholz, M. Schuster, and D. Schwammkrug, June 1998, *Tracking a large number of moving object in a crowded environment*, Proceedings of the IEEE Workshop on Perception for Mobile Agents (Santa Barbara) (G. Dudek, M. Jenkin, and E. Milios, eds.), pp. 28-36.

S.Carlsson and J.O. Eklundh, 1990, *Object detection using model based prediction and motion parallax*, Proceedings of the European Conference on Computer Vision (Antibes), Springer-Verlag, pp. 297-206.

Gutman, P., Velger, M., 1990, *Tracking Targets Using Adaptive Kalman Filtering*, IEEE Transactions on Aerospace and Electronic Systems Vol. 26, No. 5: pp. 691-699.

Welch, G, Bishop, G., 2001, *An Introduction to the Kalman Filter*, ACM, IGGRAPH 2001, Course8, www.cs.unc.edu/welch/kalman.

Maria Isabel Ribeiro, 2004, *Kalman and Extended Kalman Filters: Concept, Derivation and Properties*. Institute for Systems and Robotics. Lisboa PORTUGAL.

Simon J. Julier Jeffrey K. Uhlmann, 1997, *A New Extension of the Kalman Filter to Nonlinear Systems*. The Robotics Research Group, Department of Engineering Science, The University of Oxford.

M. Isard and A. Blake, 1998, *CONDENSATION: Conditional Density Propagation for visual tracking*, International Journal of Computer Vision, Vol. 29, No. 1, pp. 5-28.

M. Isard and A. Blake, 1996, *Contour tracking by stochastic propagation of conditional density*. In European Conference on Computer Vision, volume 1, pages 343–356.

M. S. Grewal and A. P. Andrews, 1993, *Kalman Filtering*. Prentice Hall.

Burnet, F. M., 1978, *"Clonal Selection and After"*, In Theoretical Immunology, (Eds.) G. I. Bell, A. S. Perelson & G. H. Pimbley Jr., Marcel Dekker Inc., 63-85.

Leandro Nunes de Castro, Fernando J. Von Zuben, 2002, *The Clonal Selection Algorithm with Engineering Applications*. School of Electrical and Computer Engineering, State University of Campinas, Brazil.

G. Healey, R. Kondepudy, 1994,*"Radiometric CCD camera calibration and noise estimation"*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 16, No. 3, pp. 267-276.

Foley, J., van Dan, A., Feiner, S., and Hughes, J., 1990,*Computer Graphics: Principles and Practice*, chapter 13, 563-604, Addison-Wesley.

A FEATURE DETECTION ALGORITHM FOR AUTONOMOUS CAMERA CALIBRATION

Kyung Min Han and Guilherme N. DeSouza
University of missouri-columbia, Columbia, Missouri, USA
Khx8d@mizzou.edu, desouzag@missouri.edu

Keywords: Autonomous camera calibration, Automatic feature detection, Line and Corner detection.

Abstract: This paper presents an adaptive and robust algorithm for automatic corner detection. Ordinary camera calibration methods require that a set of feature points – usually, corner points of a chessboard type of pattern – be presented to the camera in a controlled manner. On the other hand, the proposed approach automatically locates the feature points even in the presence of cluttered background, change in illumination, arbitrary poses of the pattern, etc. As the results demonstrate, the proposed technique is much more appropriate to automatic camera calibration than other existing methods.

1 INTRODUCTION

In any automatic camera calibration procedure, one must devise an algorithm that can – without human intervention – accurately and reliably: 1) identify special features in a set of images; and 2) correspond the features over the image set so they can be used as calibration points. In order to accomplish that, such algorithm must be able not only to detect as many features as possible, but it also needs to determine the pixel coordinates of the features in a consistent manner throughout the image set.

The existing algorithms for camera calibration (Zhang 1998) (Huang and Boufama 2006) (Weng 1992) (Tsai 1987) rely mainly on detecting corners on a chessboard-like calibration pattern, or the centroid of circles in a dotted calibration pattern (Kim and Kwon 2001). Other approaches (Chen 2005) (Baker and Aloimonos 2000) try to avoid using such patterns, but despite the method used, a set of corresponding points in multiple images must always be obtained

The main problem with some of these approaches (Zhang 1998) (Huang and Boufama 2006) (Weng 1992) (Tsai 1987) is that, while a large number of feature points can be easily obtained, the correspondences between features can be compromised by perspective distortions, changes in illumination, etc. That is, due to, for example, the relative pose of the pattern with respect to the illumination source, the same corner point found in one image of the set may be detected by the algorithm a few pixels off from its actual location. Moreover, many of the algorithms above mentioned require that the user define the location and/or size of a search window where the algorithm will look for the feature points (Harris and Stephen 1988). In an automatic calibration procedure, where the pattern may be presented to the camera at different depths (scale), a restriction on the size of the window would obviously render the algorithm useless.

In this paper, we present an algorithm for automatic camera calibration that relies on a line detection method (Hough Transforms) to find the feature points. In our system, a sequence of images is captured by the camera(s) while a calibration pattern is arbitrarily moved in front of the camera(s).

The proposed algorithm automatically searches for feature points on the pattern that will be used for calibration. As in the above algorithms, the feature points are the corners of the squares in a chessboard pattern, but unlike in these algorithms, the points are now defined by the intersection of the many vertical and horizontal lines running over the edge of the squares.

That is, instead of looking for localized feature discontinuities inside a small search window, as in traditional corner detection algorithms, our algorithm uses a global property of the pattern to localize the corner more accurately.

Our algorithm is very robust to cluttered background and it can reject points outside the perimeter of the pattern even if the background presents distinctive features similar to the ones in the pattern. Also, due to the use of global rather than local features, the calculated pixel coordinates of the corners are significatively more accurate than those obtained using corner detection algorithms, leading to a much more accurate final camera calibration.

2 PROPOSED ALGORITHM

The proposed algorithm consists of two main parts. In the first stage, the algorithm searches for visible features in a set of images of the calibration pattern. Once the features are located, the algorithm determines the feature correspondence between images. The output of this stage of the algorithm is a list of world coordinates of the features and their corresponding pixel coordinates in the various images used.

The second stage of the algorithm implements a camera calibration procedure based on Zhang's algorithm (Zhang 1998). This part of the algorithm is outside the scope of this paper and will not be covered here. In the next section we will present the first stage of the algorithm in more detail.

2.1 Feature Detection

Our algorithm uses a chess board pattern as depicted in Figure 1. The pattern contains one gray square in the middle, while all others are black. The reason for this special square is for the algorithm to be able to locate the origin of the pixel coordinate system and to assign coordinates to the features automatically. The main constrain imposed to this algorithm is to detect a significant number of points so that the calibration error can be minimized.

Through experimentation, it was determined that at least 150 points out a total of 196 points of the pattern must be detected for good calibration. Thus, in the ideal case, the algorithm must find a total of 28 lines – i.e. 14 horizontal lines and 14 vertical lines. The corner points are defined by the intersections of the two sets of fourteen lines.

2.2 The Hough Transform

The Hough Transform (Hough 1966) is one of the most popular methods for extracting lines from images. It is used to transform *u-v* pixel coordinates of points on a line into the parameters of such line. That is, consider, for example, the equation of a straight line in the image space, $v = m * u + c$. Where *m* is the slope and *c* is the vertical intercept. This equation can be represented by a single point in the parametric space. Since the actual *m* and *c* of such a line is initially unknown, the Hough transformation can be performed by accumulating "votes" from every point (u, v) on the line. That is, every point (u, v) will vote for all points of the line $c = m * u - v$ in the *m-c* space. Since all *u,v*-points on the same line will vote for a common *m-c* point in this parametric space, one point will accumulate more votes than any other point – which can be used to detect the original line in the image. Due to noise, the Hough algorithm could detect more than one set of parameters for a single line in the image. One of the key points in the proposed algorithm is to eliminate those erroneous detections. For that, the proposed algorithm must adapt to different situation, such as the orientation and size of the pattern, different illumination conditions, etc.

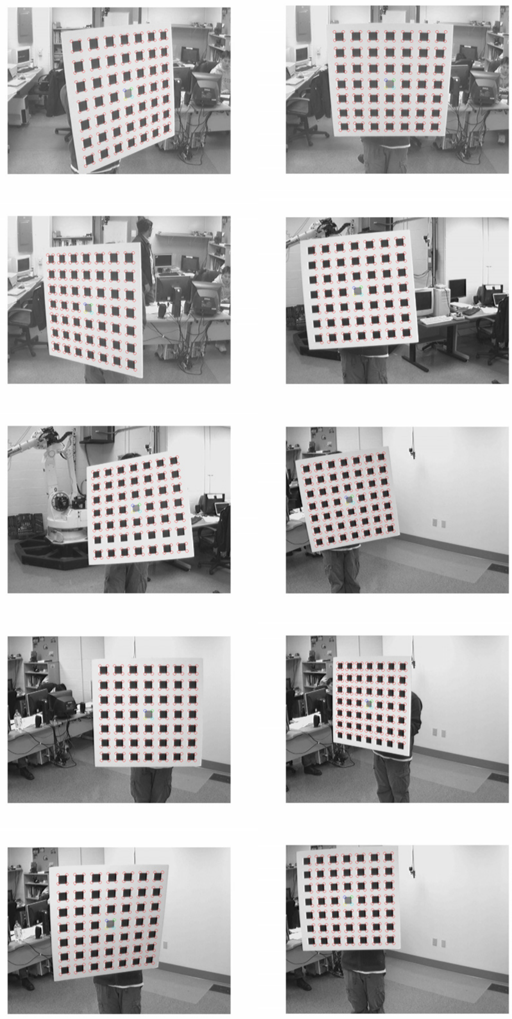


Figure 1: A sample of the typical poses of the pattern presented to the camera for calibration.

2.3 Detailed Algorithm

The first step of the algorithm is an edge detection. Then the Hough transformation is applied to all points on the edge images. Next, as we explained earlier, our algorithm searches for the intersections of all lines obtained from the Hough transform. At that point, due to noise in the images, two erroneous situations may arise. First, spurious lines outside the pattern may be detected. Second, multiple lines can be detected for a single line in the pattern.

The first erroneous case is handled by the algorithm using a set of simple but comprehensive heuristics, such as: 1) the slope of any line must be similar to the slope of thirteen other mostly vertical or horizontal lines; 2) the distance between lines must be consistent among the two sets of lines (vertical and horizontal); and 3) the number of expected lines.

It is important to mention here that the two sets of lines, vertical and horizontal, are not necessarily as so. That is, the algorithm allows for the pattern to be presented in any orientation – as it is demonstrated in Figure 1. The use of the term “vertical” and “horizontal” above is just for clarity of the explanation.

The second erroneous detection is illustrated by Figure 3(b). As it is shown in this figure, the Hough transform may detect multiple lines for a single line on the pattern. That results in multiple intersections for a single corner. In order to handle these cases, the algorithm first groups these points by their Euclidean distances. Once the clustering is obtained, the algorithm uses some stochastic criteria to eliminate erroneously detected corners. For example, the algorithm eliminates outliers farther than ½ standard deviations from the mean and recalculates the pixel coordinate of the corner afterwards. Once the algorithm processes the steps above, it then calculates the mean of each cluster. These means represent the corner points of the pattern. A predefined order of the corners allows us to search and label the corner points starting from the center of the pattern. For this reason, finding the exact position of the center square (gray square) is a critical step of the proposed algorithm. Figure 2 shows a brief flow chart of the proposed algorithm

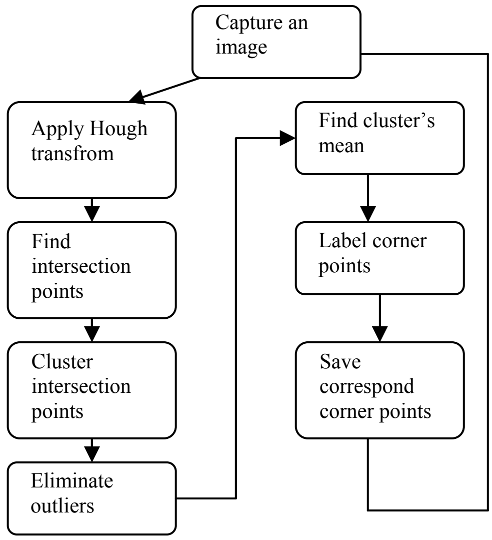


Figure 2: A brief flow chart of the proposed algorithm.

3 RESULTS

In this section we detailed two of the tests performed to validate our algorithm. In the first test, we compared a corner detection algorithm found in the literature (Harris and Stephens 1988) against our proposed method. In the second test, we present the final accuracy in 3D reconstruction after employing our algorithm to calibrate a multi-stereo rig composed of 6 cameras.

3.1 Corner Detection

In order to compare our method with a traditional corner detection algorithm, we collected 196 points in one image of the pattern at a typical position and orientation (Figure 4).

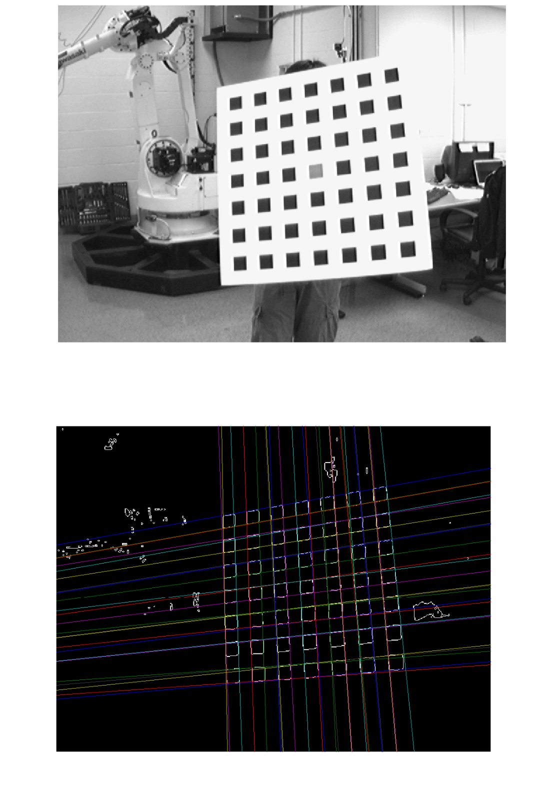


Figure 3: (a) Original image, (b) Detected Lines.

As the red circles in the figure depicts, the corner detection algorithm finds many spurious points in the image outside the boundaries of the pattern. As explained earlier, these types of algorithms require the delineation of regions of interests for their proper operation. Since our goal is to use the algorithm autonomously, such delineation must not be performed, which leads to a bad performance of the corner detection.

On the other hand, most of the points detected by our proposed algorithm lie within the pattern boundaries. However, even if one or more points happen to fall outside the pattern boundary – due to erroneous extraction of lines outside the pattern – the second stage of the algorithm can still reject those points (as explained in Section 2.3).

As it can be seen in the blown-up images of the pattern, the corner detection algorithm presents a very large variance in the actual determination of the pixel coordinates of the features.

Table 1 presents a quantitative measurement of the performance of both algorithms regarding this variation in the position of the corners.

In order to obtain such measurement, we defined a *ground truth* by manually clicking on 42 corner points in the image. The so defined ground truth was then used to compare both algorithms.

As it is demonstrated in Table 1, the proposed algorithm outperformed the corner detection algorithm in terms of the distance between the detected coordinate of the corner and the expected coordinate of that same corner. That average distance in the proposed algorithm is less than half of the distance from the other algorithm. That difference in performance can lead to a very bad calibration of the camera, as pointed out earlier.

Another important point to make about the advantage of the proposed algorithm can be demonstrated by Figure 1. As that figure shows, our algorithm is quite robust to changes in pose of the pattern and background. To validate that point, we took 100 snapshots of the pattern from 6 different cameras in our lab. In all cases, the algorithm detected the feature points in the pattern without any problems.

Table 1: Distance in pixels between detected features and ground truth.

	Average distance (in pixels)	**Stdrd deviation (in pixels)**
Proposed algorithm	0.955	0.7159
Algorithm in (Harris and Stephens 1988)	2.324	0.7883

3.2 Result from 3D Reconstruction

52Next, we tested our algorithm by carrying out the complete calibration of a total of 6 cameras and by determining the 3D coordinates of a set of arbitrary points in space using the calibrated camera. That is, using the calibration matrix obtained using the proposed algorithm and the pixel coordinates of a set of predefined points in all 6 cameras, we reconstruct the spatial coordinates of these points and compared the calculated values with the real ones. The points in space were defined by making special marks on a ruler.

The calibration error was measured by averaging the result from 20 different snapshots while holding the ruler. The marks on the ruler were placed at exactly 50cm apart. Each snapshot is taken by all 6 cameras, so a total of 120 images were used for this test. The accuracy of the final calibration was determined by calculating the distance between the two marks. Figure 5 illustrates the above procedure.

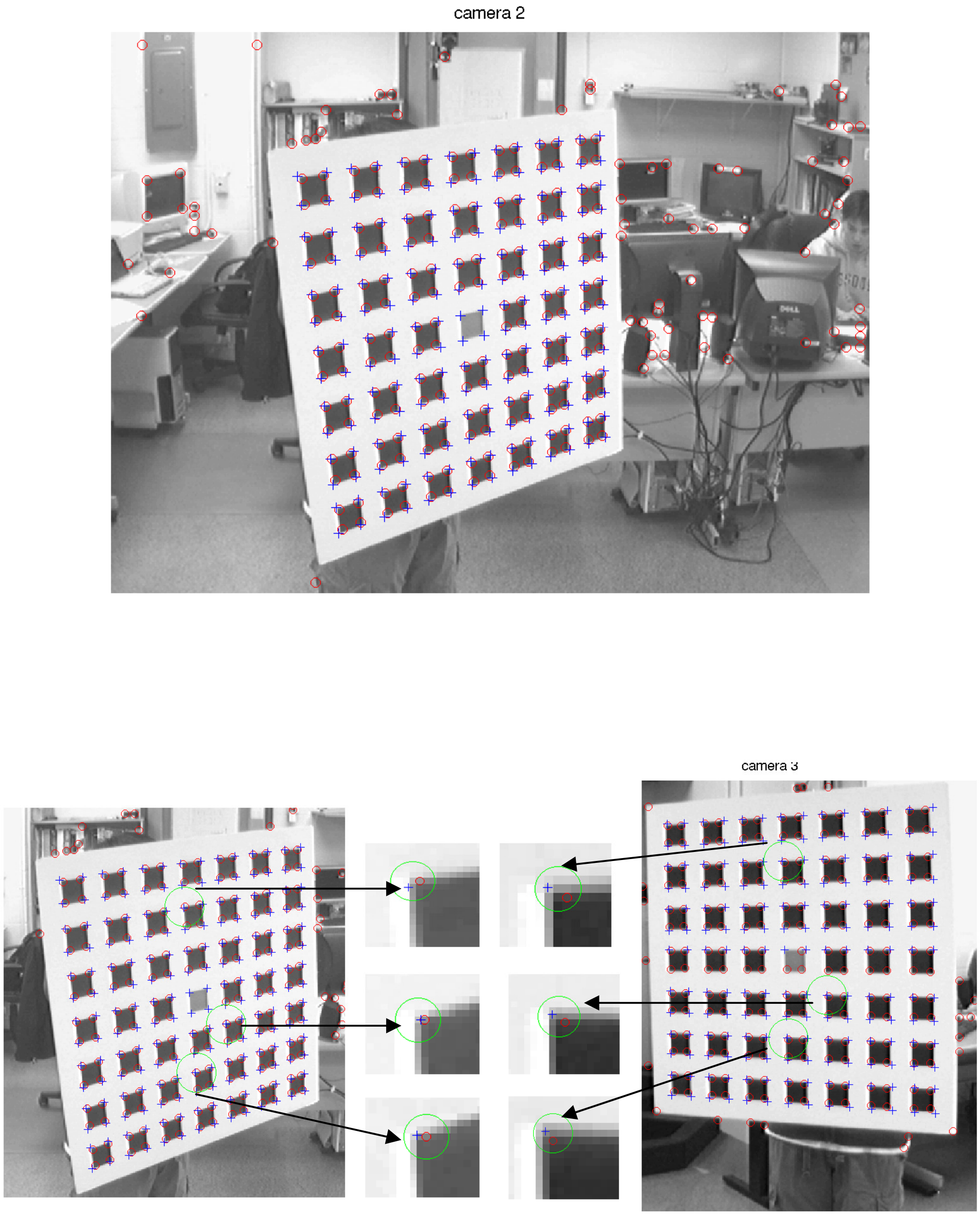


Figure 4: (a) Comparisons between a corner detection algorithm (Harris and Stephens 1988) and the proposed algorithm. The red circles indicate the result from the corner detection algorithm, while the crosses indicate the output of the proposed algorithm. (b) Discrepancies of feature points in (Harris and Stephens) corner detection technique.

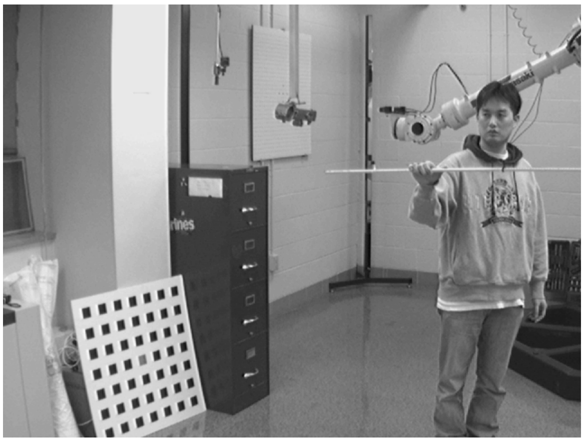


Figure 5: One of the 120 testing images used for 3D reconstruction.

As can be seen from Table 2, the accuracy in 3D reconstruction is quite reasonable – less than 1.5% of the actual distance. Also, the small standard deviation shows that the calibration obtained with our algorithm give a very consistent 3D reconstruction.

Table 2: Mean distances of 20 positions of the ruler.

The number of positions of the ruler	Mean distance between the two points (cm)	Error standard deviation (cm)
20	50.6264	0.2498

4 CONCLUSION

We presented an autonomous feature detection algorithm using Hough transforms. The proposed algorithm was compared against other traditional corner detection algorithms and the results indicate that not only our algorithm is more consistent regarding the detection of the feature points, but it is also more robust with respect to cluttered backgrounds. Both properties of the algorithm allow its use in an autonomous camera calibration procedure – which was the main motivation for this work.

Finally, the experimental results obtained demonstrate the superiority of our approach when compared to other existing algorithms. The proposed algorithm presented an average error of less than half of that of a traditional corner detection algorithm. Also, in terms of the final accuracy in 3D reconstruction using our algorithm, the results showed a quite insignificant error – just a few millimeters. In fact, such small error could be originated from the pixel quantization used in our tests. That is, as it is shown in Table 3, the simple quantization of one or two pixels can lead to approximately the same error in 3D reconstruction as the one from our algorithm.

Table 3: Error in 3D reconstruction due to pixel quantization.

Trial #	Error due to 1 pixel off (cm)	Error due to 2 pixel off (cm)
1	0.2130	0.4398
2	0.1576	0.3135
3	0.2420	0.4785

REFERENCES

P.V.C Hough, "*Methods and Means for Recognizing Complex Patterns*," U.S. Patent 069654, 1962

Zhengyou Zhang. *"A Flexible New Technique for Camera Calibration"* Microsoft Research Microsoft corporation, One Microsoft way 1998

Jun-Sik Kim, In-So Kweon, *"A New Camera Calibration Method for Robotic Applications,"* Proceedings of the IEEE, International Conference on Intelligent Robots and Systems, Oct 29- Nov 03, 2001

Zonglei Huang & Boubakeur Boufama, *"A Semi-automatic Camera Calibration Method for Augmented Reality,"* Proceedings from the 2002 IEEE International Conference on Systems, Man, and Cybernetics, vol. 4 no. pp.6,. Oct 2006.

Juyang Weng, Paul Cohen, and Marc Herniou, *"Camera Calibration with Distortion Models and Accuracy Evaluation,"* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol 14, no. 10, Oct 1992

Roger Y. Tsai. *"A versatile camera calibration technique for high accuracy 3D machine vision metrology using off-the-shelf TV camera and lense"*. IEEE Transactions on Robotics and Automation, vol. RA-3, no.4, August 1987

C. Harris and M. J. Stephens, *"A combined corner and edge detector,"* Proceedings of the 4th Alvey Conference, pages 147--152, Manchester, UK, August 1988.

Kuan-Wen Chen, Yi-Ping Hung, and Yong-Sheng Chen, *"On Calibrating a Camera Network using Parabolic Trajectories of a Bouncing Ball,"* Proceedings of the 2nd IEEE international Workshop on VS-PETS, Beijing, China, October 15-16, 2005

Patrick Baker and Yiannis Aloimonos, *"Complete Calibration of a Multi-camera Network,"* Omnidirectional Vision, 2000, Proceedings, IEEE Workshop, page 134-141, Jun 12, 2000

Hirotake Yamazoe, Akira Utsumi, and Shinji Abe *"Multiple Camera Calibration with Bundled Optimization using Silhouette Geometry Constraints,"* ICPR, Hong Kong, China, August 20-24, 2006

COMPARING COMBINATIONS OF FEATURE REGIONS FOR PANORAMIC VSLAM

Arnau Ramisa, Ramón López de Mántaras
Artificial Intelligence Research Institute, UAB Campus, 08193, Bellaterra, Spain
aramisa@iiia.csic.es, mantaras@iiia.csic.es

David Aldavert, Ricardo Toledo
Computer Vision Center, UAB Campus, 08193, Bellaterra, Spain
aldavert@cvc.uab.es, ricardo.toledo@cvc.uab.es

Keywords: Affine covariant regions, local descriptors, interest points, matching, robot navigation, panoramic images.

Abstract: Invariant (or covariant) image feature region detectors and descriptors are useful in visual robot navigation because they provide a fast and reliable way to extract relevant and discriminative information from an image and, at the same time, avoid the problems of changes in illumination or in point of view. Furthermore, complementary types of image features can be used simultaneously to extract even more information. However, this advantage always entails the cost of more processing time and sometimes, if not used wisely, the performance can be even worse. In this paper we present the results of a comparison between various combinations of region detectors and descriptors. The test performed consists in computing the essential matrix between panoramic images using correspondences established with these methods. Different combinations of region detectors and descriptors are evaluated and validated using ground truth data. The results will help us to find the best combination to use it in an autonomous robot navigation system.

1 INTRODUCTION

Autonomous robot navigation is one of the most challenging problems of mobile robotics and, although it has been widely studied, it is far from being solved completely. To date, the most successful approaches are a set of techniques known as SLAM (Simultaneous Localization And Mapping) (Thrun, 2002). These methods consist in iteratively searching for an optimal solution to both problems: self localization and map building. SLAM methods can be classified amongst three main categories: metric SLAM, for methods that give an accurate localization and mapping (Thrun, 2002; Castellanos and Tardos, 2000); topologic SLAM, where the environment is usually represented as a graph of “places” and the connectivity information amongst them (Tapus and Siegwart, 2006); and finally hybrid approaches, which try to combine the advantages of both techniques and reduce their drawbacks (Tomatis et al., 2002).

To correct the accumulative errors of odometry, SLAM techniques use additional information from the environment acquired with sensors like sonars, laser range-scanners, etc. When a camera is used to obtain such data, the method is known as Visual SLAM. Recently, this approach has gained strength thanks to the development of new computer vision algorithms that extract discriminative and meaningful information from the images. One promising line of research consists in using invariant visual features extracted from images to construct a map of the environment and locate the robot in it (Se et al., 2001; Booij et al., 2006; Ramisa et al., 2006). The core of these methods consist in finding corresponding features between two or more images acquired by the robot and set up relations between the places where this images were taken. One particularly interesting subset of invariant features are the affine covariant regions, which can be correctly detected in a wide range of acquisition conditions (Mikolajczyk et al., 2005). Equally important are the local descriptors such as SIFT (Lowe, 2004), which make the matching of local regions acquired in different conditions possible.

In (Ramisa et al., 2006), the authors developed a topological localization method which uses constellations of affine covariant regions from a panoramic image to describe a place (for example a room). When a new panorama of features is acquired, it is compared to all the stored panoramas of the map, and the most similar one is selected as the location of the robot. Using different types of feature detectors and descriptors simultaneously increases the probability of finding good correspondences, but at the same time can cause other problems, such as more processing time and more false correspondences. As means to improve the results of their approach, in this article various of these covariant region detectors and descriptors are compared. Our objective is to evaluate the performance of different combinations of these methods in order to find the best one for visual navigation of an autonomous robot. The results of this comparison will reflect the performance of these detectors and descriptors under severe changes in the point of view in a real office environment. With the results of the comparison, we intend to find the combination of detectors and descriptors that gives better results with widely separated views.

The remainder of the paper is organized as follows. Section 2 provides some background information in affine covariant region detectors and descriptors. Section 3 explains the experimental setup used in the comparison and section 4 presents the results obtained. Finally, in section 5 we close the paper with the conclusions.

2 DETECTORS AND DESCRIPTORS

Affine covariant regions can be defined as sets of pixels with high information content, which usually correspond to local extrema of a function over the image. A requirement for these type of regions is that they should be covariant with transformations introduced by changes in the point of view, which makes them well suited for tasks where corresponding points between different views of a scene have to be found. In addition, its local nature makes them resistant to partial occlusion and background clutter.

Various affine covariant region detectors have been developed recently. Furthermore, different methods detect different types of features, for example Harris-Affine detects corners while Hessian-Affine detects blobs. In consequence, multiple region detectors can be used simultaneously to increase the number of detected features and thus of potential matches.

However, using various region detectors can also introduce new problems. In applications such as VSLAM, storing an arbitrary number of different affine covariant region types can increase considerably the size of the map and the computational time needed to manage it. Another problem may arise if one of the region detectors or descriptors gives rise to a high amount of false matches, as the mismatches can confuse the model fitting method and a worse estimation could be obtained.

Recently Mikolajczyk et al. (Mikolajczyk et al., 2005) reviewed the state of the art of affine covariant region detectors individually. Based on Mikolajczyk et al. work, we have chosen three types of affine covariant region detectors for our evaluation of combinations: Harris-Affine, Hessian-Affine and MSER (Maximally Stable Extremal Regions). These three region detectors have a good repeatability rate and a reasonable computational cost.

Harris-Affine first detects Harris corners in the scale-space using the approach proposed by Lindeberg (Lindeberg, 1998). Then the parameters of an elliptical region are estimated minimizing the difference between the eigenvalues of the second order moment matrix of the selected region. This iterative procedure finds an isotropic region, which is covariant under affine transformations.

The Hessian-Affine is similar to the Harris-Affine, but the detected regions are blobs instead of corners. Local maximums of the determinant of the Hessian matrix are used as base points, and the remainder of the procedure is the same as the Harris-Affine.

The Maximally Stable Extremal region detector proposed by Matas et al. (Matas et al., 2002) detects connected components where the intensity of the pixels is several levels higher or lower than all the neighboring pixels of the region.

Matching local features between different views implicitly involves the use of local descriptors. Many descriptors with wide-ranging degrees of complexity exist in the literature. The most simplest descriptor is the region pixels alone, but it is very sensitive to noise and illumination changes. More sophisticated descriptors make use of image derivatives, gradient histograms, or information from the frequency domain to increase the robustness.

Recently, Mikolajczyk and Schmid published a performance evaluation of various local descriptors (Mikolajczyk and Schmid, 2005). In this review more than ten different descriptors are compared for affine transformations, rotation, scale changes, jpeg compression, illumination changes, and blur. The conclusions of their analysis showed an advantage in performance of the Scale Invariant Feature Transform (SIFT) introduced by Lowe (Lowe, 2004) and one of its variants: Gradient Location Orientation Histogram

(GLOH) (Mikolajczyk and Schmid, 2005). Based on these results, we use these two local descriptors in our experiments. Both SIFT and GLOH descriptors divide the affine covariant region in several subregions and construct a histogram with the orientations of the gradient for each subregion. The output of both methods is a 128-dimension descriptor vector computed from the histograms.

3 EXPERIMENTAL SETUP

In this section, we describe our experimental setup. The data set of images is composed of six sequences of panoramas from different rooms of our research center. Each sequence consists of 11 to 25 panoramas taken every 20 cm. moving along a straight line predefined path. The panoramas have been constructed by stitching together multiple views taken from a fixed optical center with a Directed Perception PTU-46-70 pan-tilt unit and a Sony DFW-VL500 camera.

Apart from the changes in point of view, the images exhibit different problems such as illumination changes, repetitive textures, wide areas with no texture and reflecting surfaces. These nuisances are common in uncontrolled environments.

From each panorama, a constellation of affine covariant regions is extracted and the SIFT and the GLOH descriptors are computed for each region. In Figure 1 a fragment of a panorama with several detected Hessian-Affine regions can be seen.

To find matches between the feature constellations of two panoramas, the matching method proposed by Lowe in (Lowe, 2004) is used. According to this strategy, one descriptor is compared using euclidean distance with all the descriptors of another constellation, and the nearest-neighbor wins. Bad matches need to be rejected, but a global threshold on the distance is impossible to be found for all situations. Instead, Lowe proposes to compare the nearest-neighbor and the second nearest-neighbor distances and reject the point if the ratio is greater than a certain value, which typically is 0.8. Lowe determined, using a database of 40,000 descriptors, that rejecting all matches with a distance ratio higher than this value, 90% of the false matches were eliminated while only 5% of correct matches were discarded.

Finally, the matches found comparing the descriptors of two constellations are used to estimate the essential matrix between the two views with the RANSAC algorithm. As in the case of conventional cameras, the essential matrix in cylindrical panoramic cameras verifies,

$$p_0^\top E p_1 = 0, \tag{1}$$

where p_0 and p_1 are projections of a scene point P in the two cylindrical images related by the essential matrix E. However, the epipolar constraint defines a sinusoid instead of a line. This sinusoid can be parameterized with the following equation,

$$z_1(\phi) = -\frac{n_x cos(\phi) + n_y sin(\phi)}{n_z}, \tag{2}$$

where $z_1(\phi)$ is the height corresponding to the angle ϕ in the panorama, $n_1 = [n_x, n_y, n_z]$ is the epipolar plane normal, obtained with the following expression,

$$n_1 = p_0^\top E. \tag{3}$$

The test performed consists in estimating the essential matrix between the first panorama of the sequence and all the remaining panoramas using different combinations of detectors and descriptors. As random false matches will rarely define a widely supported epipolar geometry, finding a high number of inliers reflects a good performance. To validate the results, ground truth essential matrices between the reference image and all the other images of each sequence have been computed using manually selected corresponding points. These essential matrices are then used to compute the error of the inliers of each combination of detectors and descriptors to the ground truth epipolar sinusoid.

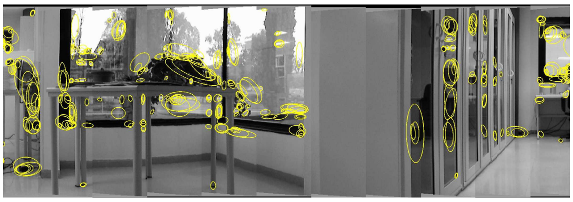


Figure 1: Some Hessian-Affine regions in a fragment of a panorama.

4 RESULTS

To evaluate the performance of each combination of methods, we measured the maximum distance at which each combination of methods passed the three different tests that are explained in the following paragraph. The results of the tests are presented in the Table 1. It is important to notice that these distances are the mean across all the panorama sequences.

Since a minimum of 7 inliers are required in order to find the essential matrix, the first test shows at which distance each method achieves less than 7 inliers. For the second test, the inliers that do not follow equation 1 for the ground truth essential matrices are rejected as false matches. Again, the distance at which the number of correct inliers drops below 7 is checked. Finally, the third test evaluates at which distance the percentage of correct inliers drops below 50%, which is the theoretic breakdown point for the RANSAC algorithm. This third test is the hardest and the more realistic one.

Table 1: Results of the comparison. For convenience we have labeled M:MSER, HA:Harris-Affine, HE:Hessian-Affine, S:SIFT, G:GLOH.

	Test 1	Test 2	Test 3
M+G	180cm	140cm	83cm
HA+G	320cm	200cm	106cm
HE+G	380cm	220cm	108cm
M+S	180cm	120cm	84cm
HA+S	400cm	220cm	101cm
HE+S	480cm	200cm	107cm
M+HE+G	480cm	200cm	106cm
HA+HE+G	480cm	220cm	100cm
M+HA+G	480cm	220cm	99cm
M+HE+S	480cm	180cm	99cm
HA+HE+S	480cm	260cm	111cm
M+HE+S	480cm	220cm	109cm
M+HA+HE+G	480cm	240cm	87cm
M+HA+HE+S	480cm	260cm	82cm

The results of the first test show that, except for the Hessian-Affine and SIFT, the combination of various detectors performs better than one detector alone. In the second test we can see that the performance of all the methods is greately reduced, and the combination of two methods (except for the Harris-Affine, Hessian-Affine and SIFT) drops to a similar level to that of one method alone. Finally, regarding the third test, the performance drops again, putting all combinations at a similar level (around 100 cm). For the third test an exponential function has been fitted to the data sets to aproximate the behaviour of the noisy data and find the estimated point where the ratio falls below 0.5.

In Figure 2 the ratio of inliers for the best combinations of each category is shown (namely, Harris-Affine and GLOH, Hessian-Affine and Harris-Affine and SIFT, and all the detectors and GLOH) as well as the fitted exponential of each of the three combinations. As can be observed in the point's cloud, the method using two regions in general achieved better results than the other two methods, and several times achieved a performance above 0.5 after the estimated point.

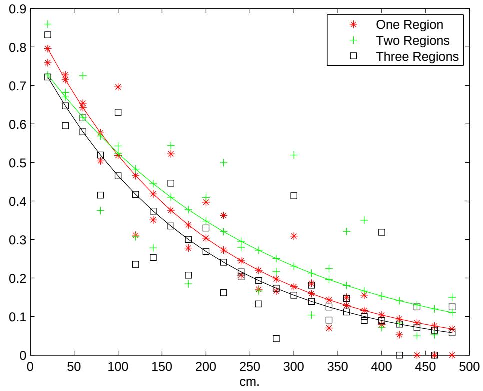


Figure 2: Ratio of inliers for the best combinations of one region (Hessian-Affine and GLOH), two regions (Harris-Affine and Hessian-Affine and SIFT) and three regions (MSER, Harris-Affine, Hessian-Affine and GLOH). Additionally, the exponential fitting of the different data sets is shown.

To obtain an estimation of the precision, the mean distance error of the inliers to the estimated epipolar sinusoid and the corresponding ground truth epipolar sinusoid has been computed for the three selected combinations (Hessian-Affine and GLOH, Hessian-Affine and Harris-Affine and SIFT, and all the detectors and GLOH). The results of this comparison are presented in figure 3. It can be seen that for the first 250 cm. all the methods have a similar error, both for the estimated epipolar sinusoid and for the ground truth one. Discontinuities are due to failures of the combinations to compute a valid essential matrix at a given distance.

Finally, a performance test has been done to compare the processing speed of the different region detectors and descriptors. This results, shown in Table

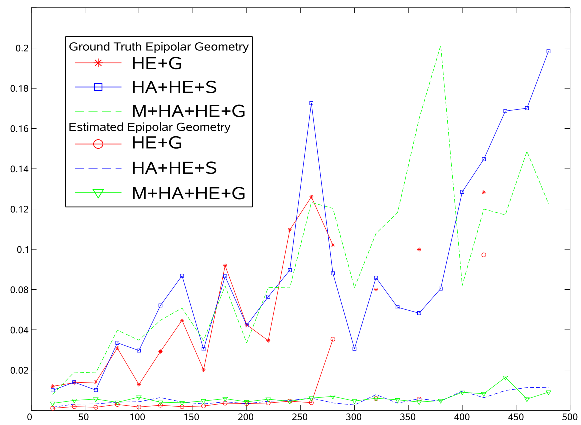


Figure 3: Mean distance error of a match to the estimated epipolar sinusoid and to the ground truth epipolar sinusoid.

2, are the mean of 50 runs of a 4946x483 panoramic image. The implementation used to perform the tests is the one given in `http://www.robots.ox.ac.uk/~vgg/research/affine/` by the authors of the different region detectors (Mikolajczyk et al., 2005). It is important to note that this implementations are not optimal, and a constant disk reading and writting time is entailed. The tests where done in a AMD Athlon 3000MHz computer.

Table 2: Time Comparision of the different region detectors and descriptors.

	Time (sec)	Regions Processed
MSER	0.95	828
Harris-Affine	8.47	3379
Hessian-Affine	3.34	1769
SIFT	14.87	3379
GLOH	16.64	3379

5 CONCLUSIONS

In this paper, we have evaluated different combinations of affine covariant region detectors and descriptors to correclty estimate the essential matrix between pairs of panoramic images. It has been shown that the direct combination of region detectors finds a higher number of corresponding points but this, in general, does not translate in a direct improvement of the final result, because also a higher number of new outliers are introduced.

No significant differences in performance have been found between the detectors individually, except that MSER is notably faster than the other methods thanks to its very simple algorithm; nevertheless the detected regions are very robust. However MSER finds a low number of regions and, as the robot moves away from the original point, the matches become to few to have a reliable estimation of the essential matrix. Regarding combinations of two region detectors, they find estimations of the essential matrix at longer distances. However, as shown in the results, the distance at which the estimations are reliable is similar to that of one detector alone. Finally, the combinations of three descriptors have shown to perform worse than the combinations of two. The reason is probably the number of false matches, that confuse the RANSAC method.

Regarding the descriptors, no significant differences have been found between the SIFT and the GLOH. The GLOH gives a slightly better performance than the SIFT, but also requires a bit more processing time.

Additionally we have found the practical limits in distance between the panoramic images in order to have a reliable estimation using these methods. This value can be used as the distance that a robot using this method to navigate in an office environment is allowed to travel before storing a new node in the map.

Future work includes experimentig with a larger data set and with different kinds of environments to verify and extend the presented results. Another interesting line of continuation would be investigating better matching and model fitting methods to reduce the proportion of false matches, and also researching ways to combine the different types of regions taking into account the scene content or estimated reliability of each region detector.

ACKNOWLEDGEMENTS

The authors want to thank Adriana Tapus for her valuable comments and help. This work has been partially supported by the FI grant from the Generalitat de Catalunya, the European Social Fund and the MID-CBR project grant TIN2006-15140-C03-01 and FEDER funds.

REFERENCES

Booij, O., Zivkovic, Z., and Kröse, B. (2006). From sensors to rooms. In *Proc. IROS Workshop From Sensors to Human Spatial Concepts*, pages 53–58. IEEE.

Castellanos, J. A. and Tardos, J. D. (2000). *Mobile Robot Localization and Map Building: A Multisensor Fusion Approach*. Kluwer Academic Publishers, Norwell, MA, USA.

Lindeberg, T. (1998). Feature detection with automatic scale selection. *Int. J. Comput. Vision*, 30(2):79–116.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *Interantional Journal of Computer Vision*, 60(2):91–110.

Matas, J., Chum, O., Urban, M., and Pajdla, T. (2002). Robust wide baseline stereo from maximally stable extremal regions. In *Proceedings of the British Machine Vision Conference 2002, BMVC 2002, Cardiff, UK, 2-5 September 2002*. British Machine Vision Association.

Mikolajczyk, K. and Schmid, C. (2005). A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 27(10):1615–1630.

Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., and Gool, L. V.

(2005). A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1/2):43–72.

Ramisa, A., Aldavert, D., and Toledo, R. (2006). A panorama based localization system. In *1st CVC Research and Development Workshop, 2006*, pages 36–41. Computer Vision Center.

Se, S., Lowe, D., and Little, J. (2001). Vision-based mobile robot localization and mapping using scale-invariant features. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2051–2058, Seoul, Korea.

Tapus, A. and Siegwart, R. (2006). A cognitive modeling of space using fingerprints of places for mobile robot navigation. In *Proceedings the IEEE International Conference on Robotics and Automation (ICRA), Orlando, U.S.A., May 2006*.

Thrun, S. (2002). Robotic mapping: A survey. In Lakemeyer, G. and Nebel, B., editors, *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann.

Tomatis, N., Nourbakhsh, I., and Siegwart, R. (2002). Hybrid simultaneous localization and map building: Closing the loop with multi-hypotheses tracking. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Washington DC, USA, May 2002*.

NAVIGATION SYSTEM FOR INDOOR MOBILE ROBOTS BASED ON RFID TAGS

Toshifumi Tsukiyama and Atsushi Suzuki
School of Information Environment,Tokyo Denki University, Inzai, Chiba, Japan
tsukiyama@sie.dendai.ac.jp

Keywords: Mobile Robot, Navigation, RFID Tag, IC Memory, Landmark, Topological Map.

Abstract: A new navigation method is described for an indoor mobile robot. The robot system is composed of a Radio Frequency Identification (RFID) tag sensor and a commercial three-wheel mobile platform with ultrasonic rangefinders. The RFID tags are used as landmarks for navigation and the topological relation map which shows the connection of scattered tags through the environment is used as course instructions to a goal. The robot automatically follows paths using the ultrasonic rangefinders until a tag is found and then refers the next movement to the topological map for a decision. Our proposed technique would be useful for real-world robotic applications such as intelligent navigation for motorized wheelchairs.

1 INTRODUCTION

This paper describes a navigation system for mobile robots which are assumed to move autonomously to a given goal in man-made environments such as hallways in a building. A key function of the navigation system is to choose a direction to a goal at a particular place such the intersection of two hallways in a building. The navigation system requires a mechanism for recognizing such particular places in the building and locating them on a world map that gives course directions to a goal.

Two common approaches to robot navigation are metric-based and landmark-based navigation (Murphy, 2000). Metric-based navigation relies on metric maps of the environment, resulting in navigation plans such as move forward five meters, turn right ninety degrees and move forward another eight meters. For position-sensing schemes, this approach relies on dead-reckoning based on information about the motion of the robot derived from the wheel encoders, or absolute position estimation using the global positioning system (GPS) (Hofmann-Wellenof, 2003). These metric data are, however, likely to be corrupted by sensor noise and this navigation method is vulnerable to inaccuracies in position estimates.

To avoid reliance on error-prone metric data, an alternative approach is landmark-based navigation. Landmark-based navigation relies on topological maps whose nodes correspond to landmarks (locally distinctive places) such as corridor junctions or doors. Map edges indicate how the landmarks connect and how the robot should navigate between them. A typical landmark-based navigation plan might be to move forward to the junction, turn into the corridor on the right, move to its end, and stop. This may involve a complete absence of metric data and then the method does not depend on geometric accuracy (Kawamura, 2002). It has apparent analogies with human spatial perception, so it is easy to make a map of the environment. In addition, topological representations avoid the potentially massive storage costs associated with metric representations.

One problem to be solved in this method is to decide what are suitable for landmarks in the environment. Landmarks should be a distinctive one and easy to recognize without special costs. In a building intersections, corners and doors are very important places for navigation and they could be a landmark. However, they are often repetitively similar and suffer from problem of occasionally sensors not being able to distinguish between similar landmarks, such as different doors of the same size. This can lead to both inefficiency and mistakes. Although such landmarks with an artificial sign could be reliable and useful, painted marks on walls would require special image processing to extract them from the scene. This would entail a complicated and costly process.

Therefore, we propose a method using Radio Frequency Identification (RFID) tags as a sign. The RFID tags are a passive, non-contact, read-only memory system based on electromagnetic wave and can store a unique number for identification of the location. The tags allow the acquisition of location information at remarkable speeds without any distance information and the accurate control of robot positions for sensing landmarks. Micro-processors with a wireless modem may be possible to give location information as a landmark (Fujii, 1997). Although they have the ability to handle data processing, they need an on-board power supply like a battery. Landmarks should be embedded in the environment and offer a virtually unlimited operational lifetime. The passive RFID tags operate without a separate external power source and obtain operating power generated from the reader. The tags are consequently much lighter, thinner and less expensive than the active devices with a microprocessor. The RFID tags can give enough location information to achieve the robustness and efficiency of navigation.

2 RFID TAGS AS LANDMARKS

Figure 1 shows an RFID tags and the antenna box of the RFID tag sensor system on a mobile robot. The RFID tag (shielded in a 12 cm square plastic plate) is an IC memory (115 Bytes) with a built-in antenna, which is pasted on walls at particular places in a building. Each IC memory has a unique ID number which can provide information on its location within the building. Figure 2 illustrates the way to get ID number in a tag through the RFID tag sensor system on the robot. The RFID tag sensor consists of an RF transceiver and an antenna. The RF transceiver illuminates an RFID tag with a short pulse of electromagnetic waves (2.45 GHz, 0.3 w). The tag receives the RF transmission, rectifies the signal to obtain DC power, reads its memory to get the ID number, and modulates the antenna backscatter in response to the interrogation. The RF transceiver obtains the ID number and reports it to the navigation system running on a Linux computer through an RS-232c serial port. Figure 3 illustrates the sensitivity map of the RFID tag sensor system. Since the induction area of a tag is 40 cm wide and 100 cm depth from the antenna the robot does not need precise positioning mechanisms to locate and access the tags. The robot just passes by tags without the accurate control of position for sensing their numbers.

Figure 1: RFID tag pasted on a wall (white panel on right) and the antenna of a RFID tag sensor mounted on a mobile robot.

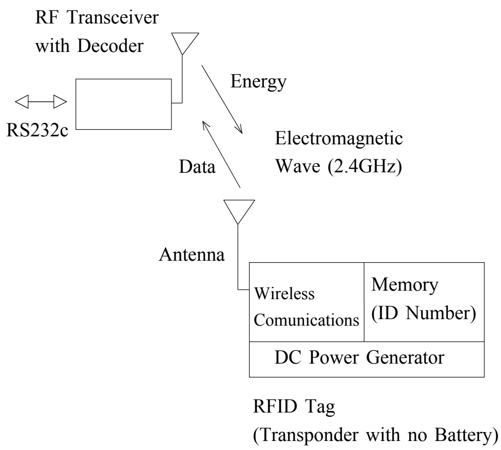


Figure 2: Architecture of the RFID tag sensor system.

3 MAP FOR NAVIGATION

Figure 4(a) shows an example of a floor plan in a building. The letters a, b, c, d, e, f, g, and h in the floor plan denote the intersections of two hallways, the junctions or near the door, which can be a particular place for a mobile robot. At these places the robot has to choose an action to reach a given goal; left-turn, right-turn, straight ahead, U-turn or stop if the place is the destination. The robot repeats one of these actions at every particular place. The particular places can be considered a sub-goal. Global tasks (i.e.

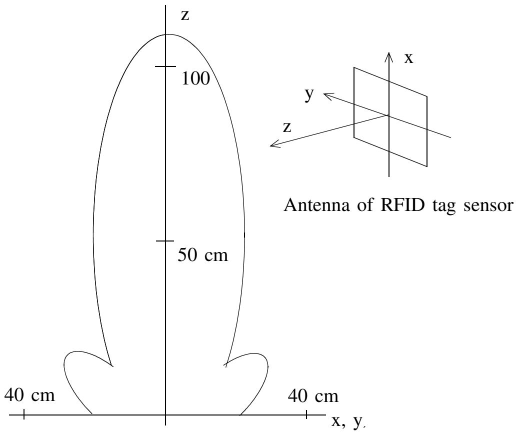


Figure 3: Sensitivity range of the RFID tag sensor system.

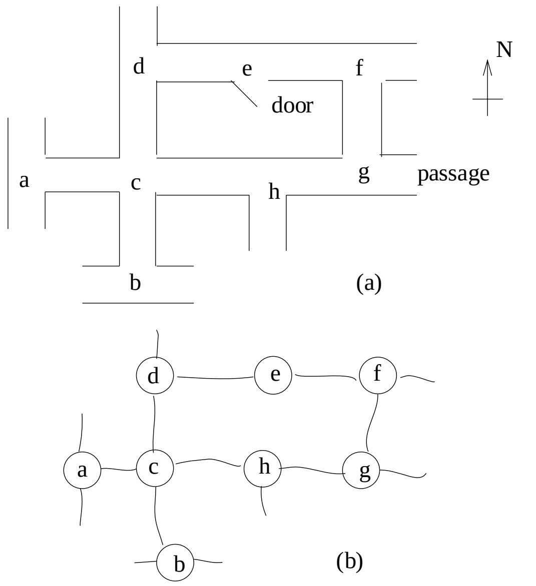
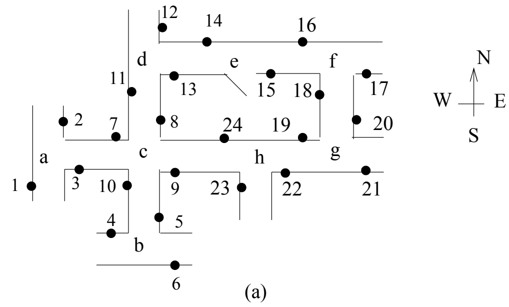


Figure 4: Example of a floor plan (a) and its topological map (b).

going to a distant goal) require a world map for deciding a sequence of sub-goals to a given goal. Figure 4(b) describes the topological relation of these particular places in the building. The nodes of the graph with a letter correspond to the particular places such as an intersection in the floor plan, respectively. An edge between two nodes denotes that a hallway exists and a robot can pass along it to the next particular place. Using graph search techniques you can generate a sequence of particular places to lead the robot to a given goal, if a starting node is given on the graph. The topological relation of particular places through hallways can be used as a world map.

To distinguish particular places in the building we



Node c			Node d		
Direction	Node	Tag ID	Direction	Node	Tag ID
N	d	8	N	?	12
E	h	9	E	e	13
S	b	10	S	c	11
W	a	7	W	--	--

(b)

Figure 5: Configuration of RFID tags for the floor plan (a) and examples of the data structure of a node (b).

use the RFID tags. The tags are pasted on the left side walls near the intersections of two hallways, the junctions or doors. Actually, it is very difficult to paste a tag precisely in a fixed position in a hallway or to measure its position. The role of tags is to give just information that the robot is coming upon an intersection and what the name of the intersection is. Figure 5(a) illustrates the configuration of the tags in the floor plan. The dots near particular places show the positions of RFID tags and the numbers are an ID number which the navigation system uses to identify upcoming intersections in the robot path. The scattered tags through hallways are used as a cue to decide the next action. In our scenario the robot moves along the left side of hallways finding tags and then the navigation system recognizes the robot's location on a world map based on the tag's ID number and decides the next movement of the robot toward a given goal.

In path planning the navigation system must decide a travel direction to the next sub-goal at an intersection. To do this the order of the adjacent edges which join at a node should be explicitly described in the world map. In other words this means to describe on the map which hallway is on the left or right side with respect to one hallway. One possible way is to draw up a list of hallways in the data structure of each node, maintaining the clockwise order of the hallways. However, this is likely to confuse the order when you trace the topological map from an entire floor plan. In our scenario robots pass through hallways in a building. They are usually intersected like a cross. Therefore, we use compass points such as North (N), South (S), East(E), West (W) for a rule of the notation of directions. When you make a world map, first, you should assign the North direction on the floor plan for the reference bearing.

The data structure of a node include the list of adjacent nodes with a compass point and tag ID numbers which a robot will find coming to a particular place. Figure 5(b) illustrates examples of the data structure of a node of the graph. The mark ”–” in the node d denotes there is no hallway in this direction, because the node d is a junction. The mark ”?” means that a hallway exists but the name of an adjacent node is unknown at the moment. If a robot finds a tag with the ID number 9 in a hallway, for example, the navigation system searches all the node data of the graph for the tag number and then finds it is in the data structure of the node c. Consequently, the navigation system recognizes the robot is coming into the intersection c from the East side. Suppose that the next sub-goal is the junction d, the navigation system searches the node c for the next node direction. It finds that the node d is adjacent to the node c and the direction is the North. The robot is coming from the East side, turns to the right at the intersection and going out to the North side. Finally it will find another tag with the ID number 11. This process is repeated until the robot finds the goal.

We have developed an interactive system for making the database of map information from a floor plan. The system was built up on a Linux computer with the graphical toll kit GTK+ and the graphical user interface (GUI) is shown in Figure 6. The small window in the upper right corner displays the menus for editing. The main window is used to draw the topological connection of intersections and junctions the robot can pass through. First, the user assigns the North direction on the floor plan for the reference bearing. The mouse button is clicked on the screen and the mouse is dragged in one of the four directions (North, South, East or West), then two nodes with an edge is displayed and the data structures for the nodes as shown in Figure 5(b) are created in the system. After this the mouse button is clicked on one of the nodes and the mouse is dragged to extend the graph. Next, the user changes the editing mode to compile the data structure of a node and clicks each node on the screen. Then another small window (in the lower left corner) appears to input the RFID tag numbers which are set near the node. This process is repeated for every particular place on the floor plan. Finally, the user selects the menu to save the database in a file, which the robot can use for path planning.

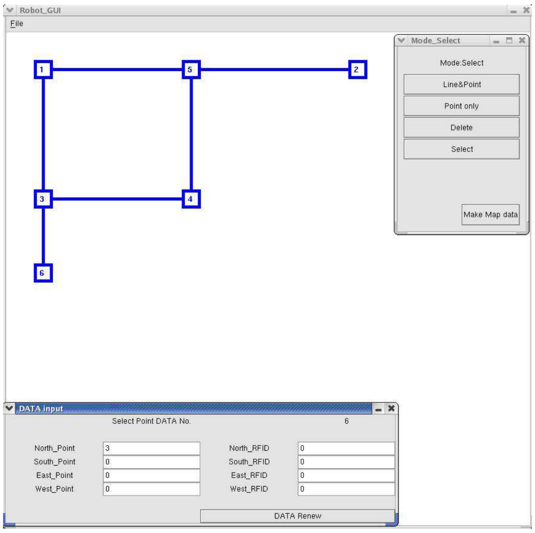


Figure 6: Interface for tracing a map.

4 ROBOT SYSTEM

Figure 7 illustrates the architecture of the navigation system. The system mainly consists of a navigation planning module, a graphical user interface (GUI) for tracing maps and a database of map information. The GUI as shown in Figure 6 enables us to make the database of map information from a floor plan. The database is a set of the data structure of nodes shown in Figure 5(b), which shows the relation of the particular places of the floor plan and assigned RFID tag numbers to each particular place. The navigation planning module decide a route to a given goal from sensed RFID tag numbers and the map information.

The navigation system was implemented on a host computer running on Linux and the mobile robot shown in Figure 1. The host computer manages the functions of the navigation planning module shown in Figure 7. The computer has two wireless RS-232c serial ports. It is responsible for handling the data from the RFID tag sensor through a wireless serial port and sending commands to the motor control module on the micro-controller of the mobile robot through another wireless serial port. The robot consists of a mobile platform and an RFID tag sensor. The mobile platform (approx. 35 cm square) is equipped with an on-board micro-controller, a two-wheel drive system with one rear free wheel, an odometer with optical encoders, seven ultrasonic rangefinders and an RS-232c serial port. The rangefinder units are mounted at the front of the robot as shown in Figure 8(a) for forward and lateral sensing and the sensitivity range is up to 5 meters. The micro-controller manages the motion control module for running the drive system and collecting position and speed information from the drive encoders, including firing the sonar sensors and retrieving echo signals.

In our method the precise positions of RFID tags in a hallway is not known. Also, the robot is not aware of the length of hallways. The robot just follows the left side walls of the hallway until it finds a tag. It is essential that an mobile robot be able to realign itself relative to a wall and then proceed if it becomes disoriented. Man-made environments such as hallways are usually constructed with a horizontal plane (a floor area) and vertical planes (walls, pillars and doors). The boundary line formed by the floor and a wall becomes a long straight line which can be seen at any point in the scene. Since the RFID tags are pasted on vertical planes, the walls, the boundary lines can be used as a guide for navigation. The geometry of sensing the distance to a wall and measuring the orientation of the robot with respect to the wall is shown in Figure 8(b). The robot is equipped with seven rangefinders as shown in Figure 8(a). The lateral rangefinder views the left side wall at two points and measure the distance ($d1$, $d2$) to the wall. The orientation of the robot with respect to the wall is calculated from the difference of the distances and the moving distance L.

Figure 9 shows the flow of motion control for the robot. The rangefinders are invoked every few meters of movement and generate the angle formed by a wall and the robot's direction and the perpendicular distance to the wall from the robot. The navigation planning module uses these data to change the orientation of the robot parallel to the wall and drives it along the wall maintaining the distance between the robot and the wall. The rangefinders also check for sudden obstacles for emergency stops. If a tag is detected while the robot is moving the navigation planning module recognizes the robot's location on the topological map from the ID number and then decides the next direction: the robot turns to the left, turns to the right, goes ahead or turns back. When changing the direction at an intersection the movement is controlled mainly by the rangefinders. The robot moves forward measuring the distance to both side walls with the rangefinders and find the center of the intersection from the measured distance. At the center the distance becomes huge because of no walls. Then, the robot turn to the indicated direction. After the robot passes an intersection the rangefinders are invoked again and the robot follows another hallway until it finds a tag.

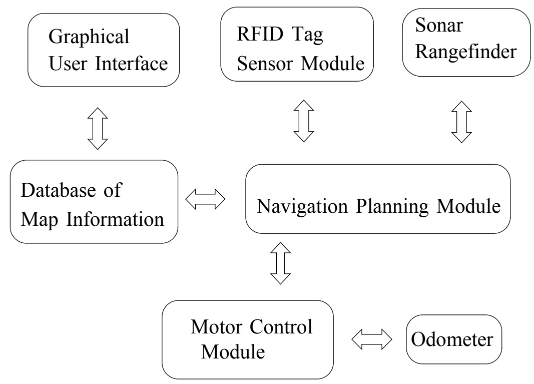


Figure 7: Architecture of the navigation system.

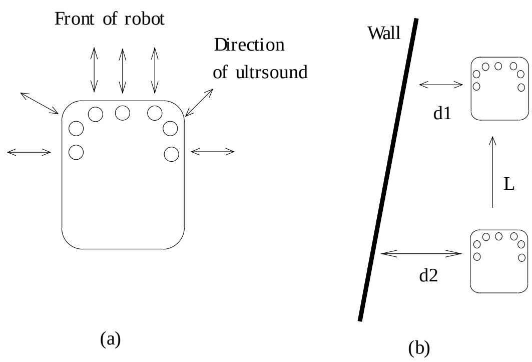


Figure 8: Sonar positions (a) and measuring the orientation of the mobile robot(b).

5 EXPERIMENTAL RESULTS

A typical test run of the mobile robot based on the navigation method is shown in the series of photographs in Figure 10. The sequence shows the robot starts near an RFID tag, orients itself in a junction and then traveling until it finds another RFID tag. This movement is similar to one along the route from the tag number 23 to 19 in Figure 5(a). The width of the passage was 1.6 m and the traveling distance along the passage was about 8 m. As shown in Figure 10(a) a tag was fixed on the low pole near the exit to a passage. Figure 10 (b) shows that the robot passed by the tag and the tag's ID number was detected by the RFID tag sensor. The navigation planning module decided next movement based on the tag's ID number. In this test run the robot was scheduled to turn to the right.

The robot moved for a while under the control of the odometer of the robot and then checked if the right side of the robot is a free space (i.e. no walls) using the ultrasonic rangefinders. This process was repeated until an enough space was found on the right side (see

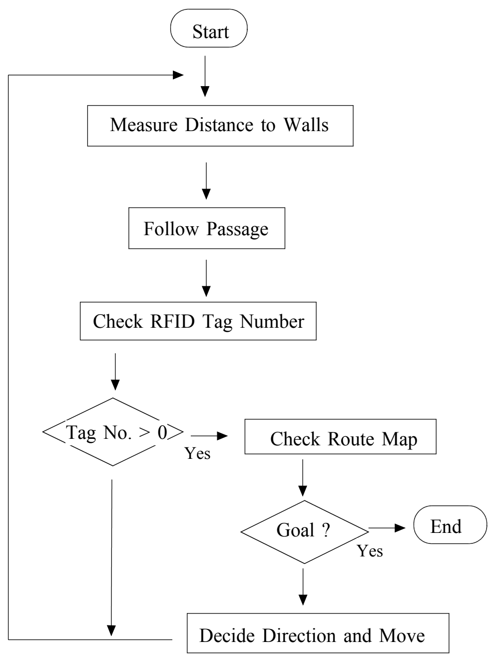


Figure 9: Flow of the navigation method.

Figure 10(c) and (d)). After that the robot turned to the right as shown in Figure 10(e) and (f). The robot moved along the passage (see Figure 10(g), (h) and (i)) and after every 1 m of movement, the rangefinder module was invoked. From the geometrical information the navigation planning module decided the rotation angle of the robot to reorient the robot parallel to the wall, and the distance needed to draw the robot near to the wall. To sense the RFID tags reliably the robot must move along the wall at a distance of less than 1 m. In this experiment the distance was 0.8 m. The sensor guidance with iterative sensing and motion continues until an RFID tag is found. In Figure 10(j) the robot suddenly faced the wall obliquely due to slip or something. The navigation plan module attempted to reorient the robot parallel to the wall (Figure 10(k)). After that, another RFID tag was detected and the robot stopped.

6 CONCLUSIONS

Mobile robots are a very imprecise mechanism. Navigation systems for mobile robots should have a mechanism to accomplish tasks with an adequate degree of precision. We thus proposed a topological navigation system using RFID tags and a sonar rangefinder system. The rangefinder was used to reorient the robot parallel to a wall keeping the distance to it. We also introduced a topological connection map of the RFID tags, which can be built without precise 3-D representation of the environment. Robots just follow the tags under instructions of the topological map. The equipment setup is very simple and the navigation system is easily combined with the robot's computer systems. This is both practical and acceptable for the application of mobile robots to the real world.

A graphical user interface for making the world map conveniently is indispensable for the application of the navigation method. We have built an interactive system running on the Linux. This is an on-going project.

REFERENCES

Murphy, R. R. (2000). Introduction to AI Robotics. MIT Press.

Hofmann-Wellenhof, B and et al. (2003). Navigation - principles of positioning and guidance. Springer.

Fujii, T. and et al.(1997). Intelligent Data Carrier System for Cooperative Behaviors Emerge among Collective Robots. Proc. of IEEE International Conference on Systems, Man, and Cybernetics, Vol. 1, pp. 299-304.

Kawamura, k and et. al.(2002). Toward Perception-Based Navigation Using EgoSphere. Proc. of SPIE, Vol. 4573, pp. 137-147.

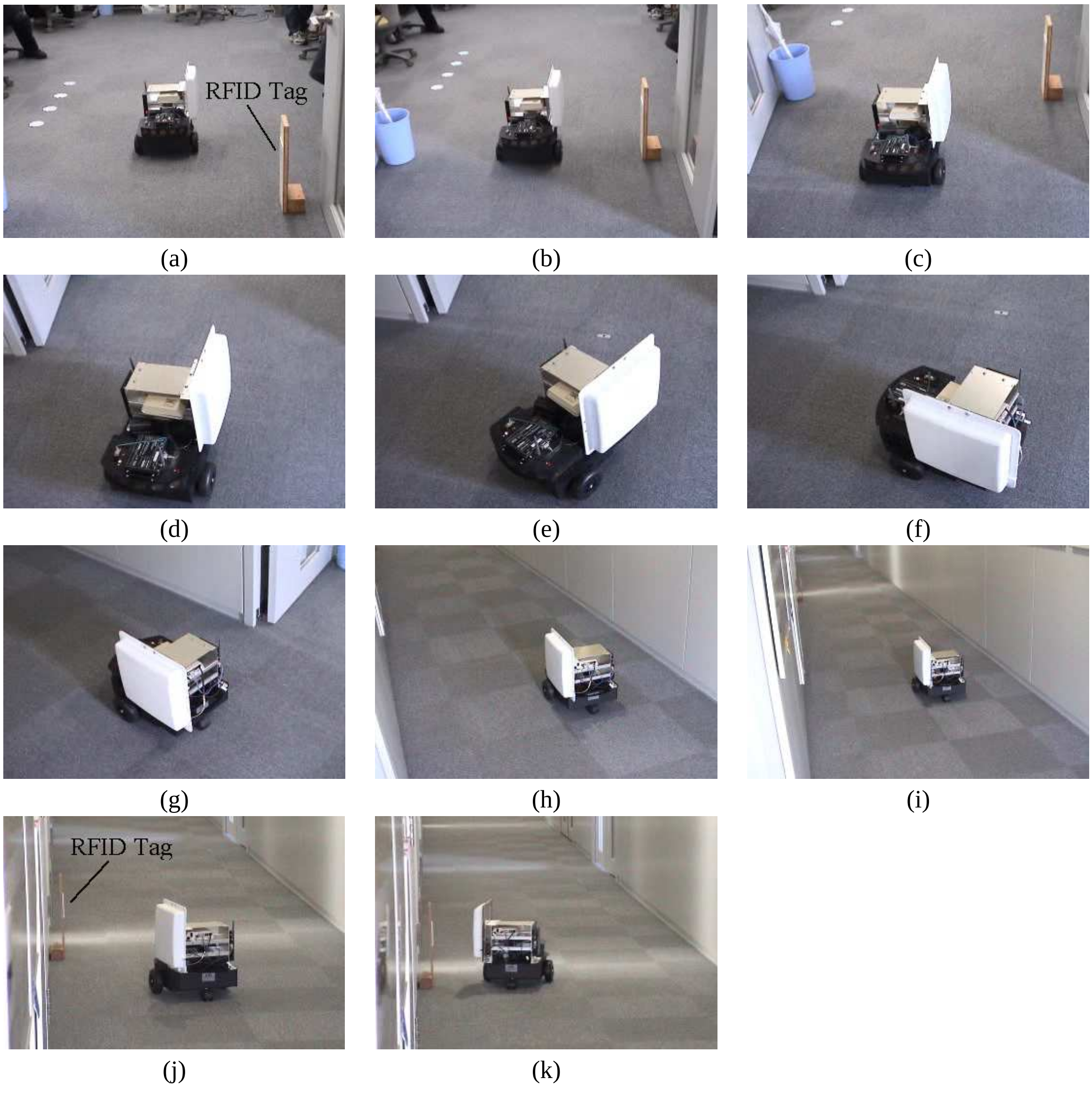


Figure 10: Test run of the mobile robot.

COMPARISON OF FINE NEEDLE BIOPSY CYTOLOGICAL IMAGE SEGMENTATION METHODS

Maciej Hrebień, Piotr Steć and Józef Korbicz
Institute of Control and Computation Engineering, University of Zielona Góra
ul. Podgórna 50, 65-246 Zielona Góra, Poland
{m.hrebien, p.stec, j.korbicz}@issi.uz.zgora.pl

Keywords: Cytology, image processing, segmentation.

Abstract: This paper describes an early stage of cytological image recognition and presents a comparision of two hybrid segmentation methods. The analysis includes the Hough transform with conjunction to the watershed algorithm and with conjunction to the active contours techniques. One can also find here a short description of image pre-processing and an automatic nucleuses localization mechanisms used in our approach. Preliminary experimental results collected on a hand-prepared benchmark database are also presented with short discussion of common errors and possible future problems.

1 INTRODUCTION

Construction of a fully automatic cancer diagnosis system is a challenging task. In last decade we can observe a very dynamic growth in number of researches conducted in this area not only by university centers but also by commercial institutions (Kimmel et al., 2003). Because the breast cancer is becoming most common disease of the present female population, much attention of the present-day researchers is directed to this issue. The attention covers not only curing the external effects of the disease but also its fast detection in its early stadium.

The nucleus of the cell is the place where breast cancer malignancy can be observed. Therefore, it is crucial for any camera based automatic diagnosis system to separate the cells and their nuclei from the rest of the image content. Until now many segmentation methods were proposed (Gonzalez and Woods, 2002; Pratt, 2001; Russ, 1999) but unfortunately each of them introduces different kinds of additional problems and usually works in practice under given assumptions and/or needs end-user's interaction/cooperation. Since many cytological projects assume rather full automation and real-time operation with high degree of efficacy, a method free of drawbacks of already known approaches has to be constructed.

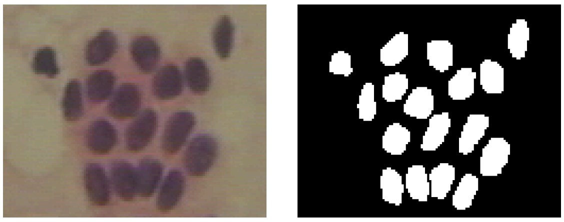
In this paper two hybrid methods of cytological image segmentation are presented, that is the Hough transform with conjunction to the watershed algorithm and with conjunction to the active contours techniques. One can also find here a short description of image pre-processing and fully automatic nuclei localization mechanisms used in our approach.

2 PROBLEM FORMULATION

Mathematical formulation of the segmentation process is very difficult because it is a poorly conditioned problem. Thus we give here only some informal definition of the problem we have to face.

What we have on input is a cytological material obtained using the Fine Needle Biopsy technique and imagined with a *Sony CCD Iris* camera mounted atop of an *Axiophot* microscope. The material comes from female patients of Zielona Góra's *Onkomed* medical center (Marciniak et al., 2005). The 704×576 pixel image itself is coded using the RGB colourspace and is not subject of any kind of lossy compression.

What we expect on output is a binary segmentation mask with one pixel separation rule which will allow us to more robust morphometric parameters estimation in our future work. Additionally, the algorithm should be insensitive to colours of contrasting



(a) (b)

Figure 1: Exemplary fragment of: (a) cytological image, (b) appropriate segmentation mask.

pigments used for preparation of the cytological material (see an example in Fig. 1).

3 IMAGE PRE-SEGMENTATION

3.1 Pre-processing

The colour components of an image do not carry as important information as the luminosity does, so they can be removed to reduce processing complexity in stages that require only e.g. gradient estimations. An RGB colour image can be converted to greyscale by removing blue and red chrominance components from the image defined in YCbCr colour space (Pratt, 2001).

Since the majority of images we deal with have low contrast, an enhancement technique is needed to improve their quality. In our approach we use simple histogram processing with linear transform of image levels of intensities, that is the cumulated sum approach (Russ, 1999). The contrast correction operation is conducted for each colour channel separately resulting in an image being better-defined for later stages of the presented hybrid segmentation methods (see Fig. 2).

3.2 The Background of the Algorithm

If we look closely at the nuclei we have to segment, they all have elliptical shape. Most of them remind ellipse but unfortunately detection of ellipse which is described by two parameters a and b ($x = acos\alpha$, $y = bsin\alpha$) and which can be additionally rotated is computationally expensive. The shape of ellipse can be approximated by a given number of circles. Detection of circles is much more simpler in the sense of required computations because we have only one parameter, that is the radius R. This observations and simplifications constitute grounding for fast nucleus pre-segmentation algorithm – in our approach we try

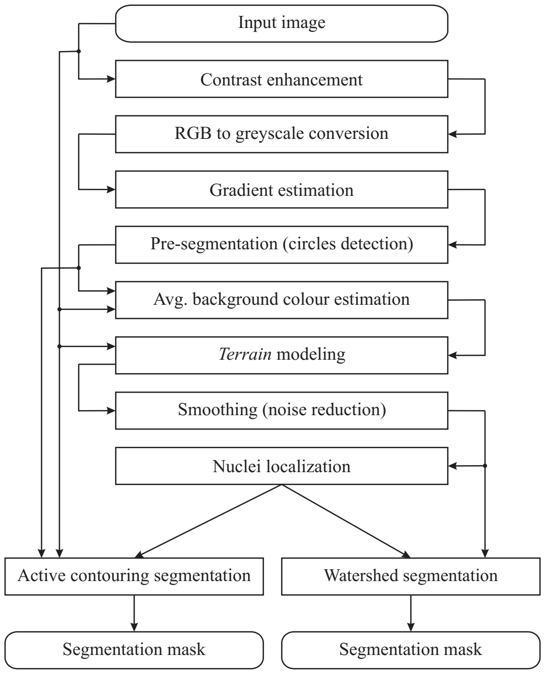


Figure 2: Flow graph of the presented solutions.

to find such circles with different radii in a given feature space.

3.3 Circles Detection

The Hough transform (Toft, 1996; Żorski, 2000) can be easily adopted for the purpose of circle detection. The transform in the discrete space can be defined as:

$$HT_{discr}(R,\hat{i},\hat{j}) = \sum_{i=\hat{i}-R}^{\hat{i}+R} \sum_{j=\hat{j}-R}^{\hat{j}+R} g(i,j)\delta\Big((i-\hat{i})^2+(j-\hat{j})^2-R^2\Big), \quad (1)$$

where g is a two dimensional feature image and δ is the Kronecker's delta (equal to unity at zero) which defines sum only over the circle. The HT_{discr} plays the role of accumulator which accumulates levels of feature image g similarity to circle placed at the $(\hat{i},\hat{j})$ position and defined by the radius R.

The feature space g can be created by many different ways. In our approach we use gradient image as the feature indicating nucleus' occurrence or absence in a given fragment of cytological image. The gradient image is a saturated sum of gradients estimated in eight directions on greyscale image prepared in the pre-processing stage. The base gradients can be calculated using e.g. Prewitt's, Sobel's mask methods or

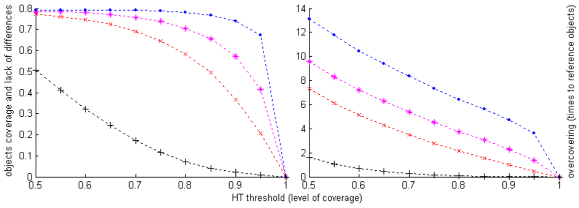
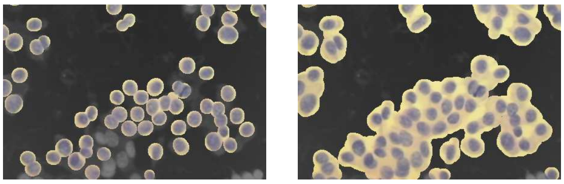


Figure 3: Influence of θ threshold value on objects's cover and lack of differences (left) and overcovering (right) for Prewitt (×), Sobel (∗), heavy (•) and light (+) base gradient masks (experiments performed on a randomly selected 346 element Zielona Góra's *Onkomed* (Marciniak et al., 2005) cytological benchmark database for radii in the 4-21 pixel range).



(a) (b)

Figure 4: Exemplary results of the pre-segmentation stage for two different θ threshold strategies: (a) high and (b) low.

their heavy or light versions (Gonzalez and Woods, 2002; Tadeusiewicz, 1992).

3.4 Final Actions

Thresholding the values in the accumulator by a given θ value can lead us to a very good pre-segmentation mechanism with the lower threshold strategy (see for instance Fig. 4). Since the threshold value strongly depends on the database and used feature image g (Fig. 3), the method can only be used as a pre-segmentation stage. Smaller value of the threshold causes fast removal of non-important information from the background what can constitute a base for more sophisticated and going into details algorithms.

4 IMAGE SEGMENTATION

4.1 *Terrain* Modeling

The results obtained from the pre-segmentation stage can lead us to the estimation of average background colour. This information can be used to model the nuclei as a colour distance between background and objects what fulfils requirements of lack of any colour

(a) (b) (c)

Figure 5: Exemplary fragment of: (a) cytological image, (b) Euclidian distance to the mean background colour, (c) smoothed out version of (b).

dependency in imaged material (the colour of contrasting pigments may change in the future). In our research we tried few distance metrics: Manhattan's, Chebyshev's, absolute Hue value from HSV colourspace but Euclidian one gives the best visual results (Fig. 5ab).

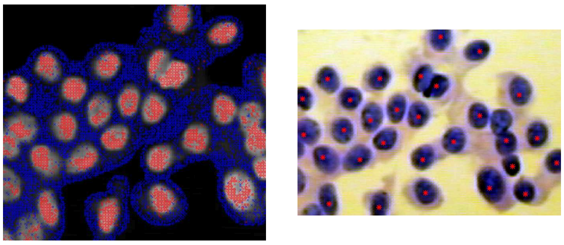
Since the modeling distance can vary in local neighborhood (see Fig. 5b) mostly because of camera sensor simplifications, a smoothing technique is needed to reconstruct the nuclei shape. The smoothing operation in our approach relies on the fact that this sort of 2D signal can be modeled as a sum of sinusoids (Madisetti and Williams, 1997) with defined amplitudes, phase shifts and frequencies. Cutting all low amplitude frequencies off (leaving only a few significant ones with the highest amplitude) will result in a signal deprived our problematic local noise effect (Fig. 5c).

4.2 Nuclei Localization

Localization of objects on a modeled map of nuclei can be performed locally using various methods. In our approach we have chosen evolutionary (1+1) search strategy (Arabas, 2004) mostly because it is simple, quite fast despite appearances, can be easily parallelized due to its nature and it settles very good in local extrema what is very important in our case.

The used watershed segmentation algorithm forced us to create two population of individuals. The first population is localizing the background. Specimens are moved with a constant movement step equal to unity and the movement is preferred to the places with a smaller density of population to maximize background coverage. The second population is localizing the nuclei. Specimens are moved with a exponentially decreasing movement step to very fast group the population near local extrema in first few epochs and to finally work on details in the ending ones. The movement of individuals is preferred to the places with a higher population density to create the effect of nuclei localization.

The change for the better position of an individ-



(a) (b)

Figure 6: Exemplary localization: (a) screenshot after 8 epochs, (b) final result (localization points are marked with red asterisks).

ual searching for nuclei is calculated as a product of randomly generated distance with normal distribution $N(0,1)$ and an decreasing in time radius $r^t = R_{max}\left(\frac{1}{R_{max}}\right)^{\frac{t}{t_{max}}}$, where R_{max} is the maximal radius detected by the Hough transform. Specimens covering background are generated in a similar way except $R = 1$ during mutation.

The fitness function calculates the average *height* of the terrain in a given position including nearest neighborhood defined by the smallest radius detected by the Hough transform in the pre-segmentation stage. Such definition of the fitness function avoids a possible split of population, localized near a nucleus with multimodal character of its shape, giving only one marker for a nucleus (Fig. 6b).

Finally, the nucleus is localized in the place where the density of the population searching for hilltops in the modeled terrain is locally maximal.

The used active contours techniques have lower requirements concerning nucleus localization. In this approach it is allowed to have more than one marker pointing the same nucleus. Thus the localization algorithm in this case can be much simplified. We need only one population, that is the one searching for nuclei and the fitness function is simply the terrain *height* at an individuals position. Additionally, it is allowed to have not optimal or even false localization points what reduces number of needed iterations of the algorithm.

4.3 Building Watersheds

The watershed segmentation algorithm is inspired by natural observations, that is a rainy day in mountains (Gonzalez and Woods, 2002; Pratt, 2001; Russ, 1999). A given image can be defined as a terrain on which nuclei correspond to valleys (upside down terrain modeled in previous steps). The terrain is flooded by rainwater and arising puddles are starting to turn into basins. When the water from one basin begins to pour away to another, a separating watershed is created.

The flooding operation have to be stopped when the water level reaches a given θ threshold. The threshold should preferably be placed somewhere in the middle between the background and a nucleus localization point. In our approach nuclei are flooded to the half of the altitude between nucleus localization point and the average height of the background in the local neighborhood. Since the images we have to deal with are spot illuminated during imaging operation (resulting in a modeled terrain being higher in the center of the image and much lower in the corners) this mechanism protects the basins against being overflooded and in consequence nuclei being undersegmented. To satisfy the one pixel separation rule the algorithm needs to have multi-label extension and the watersheds are built only when there is a neighbor nearby with other label.

4.4 Active Contours

An active contour segmentation is performed using multilabel fast marching algorithm presented in (Steć, 2005; Hrebień and Steć, 2006) which is extension to the original fast marching method (FMM) developed by Sethian (Sethian, 1998). The problem with the original FMM is that the contour can be moved only in one direction. This means that any error in segmentation cannot be corrected and algorithm requires additional stop condition. To deal with this problem, multilabel extension to the classical FMM was proposed.

Initialization of the multilabel fast marching is done in similar way as it was done for the watershed algorithm. The difference is that the watershed initialization image requires an additional processing to leave exactly one seed per nucleus while the FMM allows more seeds in one nucleus. Similar method of initialization will allow direct comparison of the segmentation results.

Initial contour propagation is similar to original FMM method. Expansion of the contour is governed by a propagation speed defined globally for all the contours. Speed is based on the difference between mean colour in the initialization area and colour of the pixel under the contour:

$$F = \frac{1}{|g(x,y) - \overline{g}(i)|^3 + 1}, \tag{2}$$

where $g(x,y)$ is the colour under the contour and $\overline{g}(i)$ is the mean colour under the i-th segment. Such a speed definition slows down the contour near the detected object boundary what increases probability of contours meeting near nucleus boundary.

When two segments meet, mean colour of the segments is compared. Comparison is taken at the point where contours start to overlap. When difference between mean colours from these two segments is below certain threshold segments are merged into one. To ensure maximum efficiency, labels from the smaller segment are changed to the value of those from the larger segment. Additionally, new mean colour for the segment is calculated from mean colours of connected segments.

If two segments that meet are not classified to be merged, the propagating segment can push back another segment under certain circumstances. At the meeting point differences between current pixel colour and mean colour of each segment is compared. Segment with lower difference value wins and replaces current label with its own. Replacement is performed as long as condition is meet. Contour that was pushed back cannot be propagated farther at places where its labels was replaced by another contour. Contour points that cannot be moved are no longer considered during calculations. Since contour can be pushed back only once, there is no oscillation at the object boundary known from the classical active contour methods. Additionally reduction of the contour length increases performance of the algorithm.

The presented algorithm stops propagation when all image points are assigned to segments and there is no segment that could push back another segment. The algorithm cannot run infinitely because oscillations between segments are impossible. No segment can visit twice the same area. Namely, when a segment was pushed back by another segment, it cannot get the lost pixels back.

4.5 Exemplary Results

Exemplary results of the presented watershed segmentation method and common errors observed on our hand-prepared benchmark database can be divided into four classes:

- *class 1*: good quality images with only small irregularities and rarely generated subbasins (basin in another basin) (Fig. 7ab),
- *class 2*: errors caused by fake circles created by spots of fat (Fig. 7cd),
- *class 3*: mixed nucleus types: red and purple in this case and those reds which are more purple than yellow (background) are also segmented what is erroneous (Fig. 7ef),

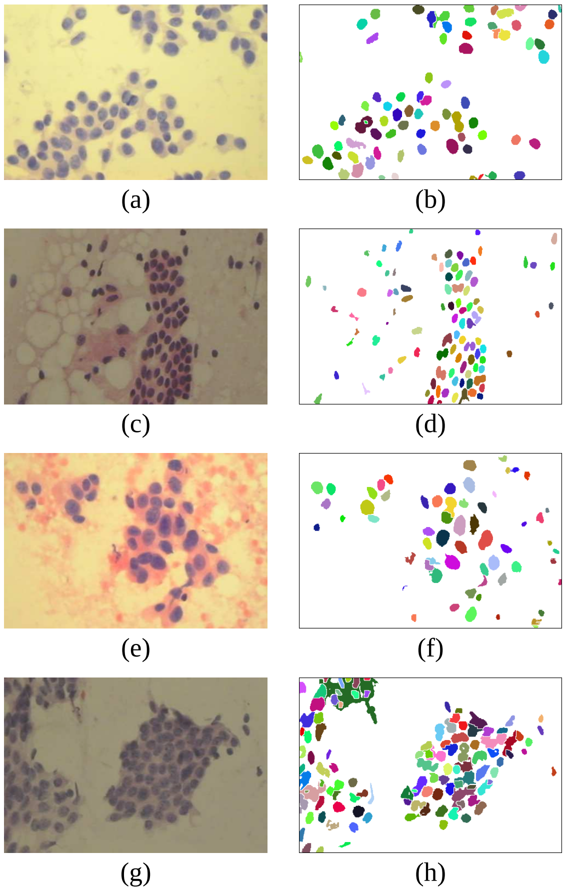
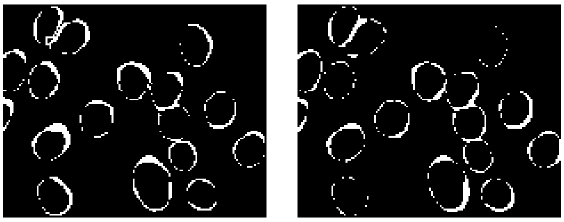


Figure 7: Exemplary results of the watershed segmentation.

- *class 4*: poor quality image with a bunch of nuclei glued together what causes basin's overflooding and in consequence undersegmentation (Fig. 7gh).

Conducted experiments show that the watershed algorithm gives 68.74% on the average agreement with the hand-prepared templates using simple XOR metric. As one can easily notice most errors are located at boundaries of nuclei (see for instance Fig. 8) where the average distance between edges of segmented and reference objects is about 3.28 pixles on the average. This causes the XOR metric to be underestimated as a consequence of not very heigh level of water flooding the modeled terrains. For the active contours algorithm the situation is very similar, that is the XOR metric gives 22.32% score and the average distance is equal to 4.1 pixels. Despite the underestimation fact the shape of nuclei seems to be preserved what is important for our future work, that is estimation of morphometric parameters of segmented nuclei.



(a) (b)

Figure 8: Common XOR metric errors for: (a) the watershed and (b) the active contours method.

5 CONCLUSIONS

Conducted preliminary experiments show that the Hough transform adopted for circle detection in the pre-segmentation stage, the (1+1) search strategy used for automatic nuclei localization, the watershed algorithm and the active contours techniques used for the final segmentation stage can be effectively used for the segmentation of cytological images.

The problem regarding fake circles created by spots of fat and unwanted effects it gives in the final output should also be considered and eliminated in future work. Images with mixed nucleus type still constitute a challenge because it seems to be impossible to detect only one type without end-user's interaction and when there should not be any dependencies and assumptions concerning colour of contrasting pigments used to prepare cytological material. The proposal hybrid methods should also be extended to perform better on poor quality images or a fast classifier should be constructed to reject too poor (or even fake) inputs.

Summarizing, the presented solutions are promising and give a good base for our further research in the area of cytological image segmentation. Additionally, all preparation steps including pre-segmentation and the automatic nucleus localization stage can be reused with other segmentation algorithms which need such a information.

Performance of both algorithms is comparable. There was no result that clearly shows superiority of one algorithm above the other. The outcome was dependent on the used metric. Visually, segmentation results from both algorithms look very similar (see Fig. 8). Both algorithms have problems with the tight clusters of nuclei. They are usually detected as a single object.

Time reaction of both algorithms is similar too and it takes several seconds on today's PCs per image to give the final segmentation mask. All preparation steps are much more time consuming (2-3 minutes) but authors believe that it can be significantly reduced mostly because of the fact that this steps were simulated in MATLAB environment. Taking the advantage of today's multi-core machines, thread-oriented operating systems, the nature of used algorithms which are easy to parallelize and rewriting them using native code generating programming language can speed up the whole process significantly. A dedicated hardware could also be considered.

REFERENCES

Arabas, J. (2004). *Lectures on Evolutionary Algorithms*. WNT. (in Polish).

Gonzalez, R. and Woods, R. (2002). *Digital Image Processing*. Prentice Hall.

Hrebień, M. and Steć, P. (2006). The Hough transform and active contours in segmentation of cytological images. In *Proc. of the 9th Int. Conf. on Medical Informatics and Technology – MIT 2006*, pages 62–68, Wisła-Malinka, Poland.

Kimmel, M., Lachowicz, M., and Świerniak, A. (2003). Cancer growth and progression, mathematical problems and computer simulations. *Int. Journal of Appl. Math. and Comput. Science*, Vol. 13, No. 3, Special Issue.

Madisetti, V. and Williams, D. (1997). *The Digital Signal Processing Handbook*. CRC Press.

Marciniak, A., Obuchowicz, A., Mończak, R., and Kołodziński, M. (2005). Cytomorphometry of fine needle biopsy material from the breast cancer. In *Proc. of the 4th Int. Conf. on Comp. Recogn. Systems CORES'05*, Adv. in Soft Computing. Springer.

Pratt, W. (2001). *Digital Image Processing*. John Wiley & Sons.

Russ, J. (1999). *The Image Processing Handbook*. CRC Press.

Sethian, J. (1998). Fast Marching Methods and Level Set Methods for propagating interfaces. In *29th Computational Fluid Dynamics*, volume 1 of *VKI Lectures series*. von Karman Institute.

Steć, P. (2005). *Segmentation of Colour Video Sequences Using Fast Marching Method*, volume 6 of *Lecture Notes in Control and Computer Science*. University of Zielona Góra Press, Zielona Góra, Poland.

Tadeusiewicz, R. (1992). *Vision Systems of Industrial Robots*. WNT. (in Polish).

Toft, P. (1996). *The Radon Transform*. Technical University of Denmark. Ph.D. Thesis.

Żorski, W. (2000). *Image Segmentation Methods Based on the Hough Transform*. Studio GiZ Warszawa. (in Polish).

ESTIMATION OF CAMERA 3D-POSITION TO MINIMIZE OCCLUSIONS

Pablo Gil, Fernando Torres
Department of Physics, Systems Engineering and Signal Theory, University of Alicante, Crta. San Vicente s/n Alicante, Spain, Pablo.Gil@ua.es, ftorres@dfists.ua.es

Oscar Reinoso
Department of Industrial Systems Engineering, Miguel Hernandez University Elche, Spain o.reinoso@umh.es

Keywords: Occlusions, Camera viewpoint, Camera pose, Segmentation.

Abstract: Occlusions are almost always seen as undesirable singularities that pose difficult challenges to recognition processes of objects which have to be manipulated by a robot. Often, the occlusions are perceived because the viewpoint with which a scene is observed is not adapted. In this paper, a strategy to determine the location, orientation and position, more suitable so that a camera has the best viewpoint to capture a scene composed by several objects is presented. The estimation for the best location of the camera is based on minimizing the zones of occlusion by the analysis of a virtual image sequence in which is represented the virtual projection of the objects. These virtual projections represent the images as if they were captured by a camera with different viewpoints without moving it.

1 INTRODUCTION

In Computer vision, one of the critical problems for object recognition is that the recognition methods should be able to handle partial occlusion of objects. The spatial distribution of structural features of an object is the most important information that an object represents. Recognition partially occluded objects have been a formidable problem in recognition processes. In recent years, several object recognition methods have been proposed to recognize occluded objects. Some of them are based on statistical models (Chan, 2002)(Ying, 2000), on graph models (Boshra, 2000)(El-Sonbaty, 2003) or based on a mix (Park, 2003). Also it is important to emphasize others studies which are based on the eigenspace analysis of images taken in the same environment. Thus an object model is built as a vector in a low dimensional eigenspace, and this way objects are recognized by comparing the model with image vectors.

The ability to recognize an object in an image is limited if it is impossible to see all the surface of the object. Not only self-occlusion are present in opaque objects since we are not able to see the back of the object, but also other objects may occlude some portion of the object that we wish to recognize and that it would otherwise be visible. In our approach, we look to change the location of the camera which observes the objects in order to improve the viewpoint and reduce the occluded portion of them. Other works, such as (Silva, 2001) show in their studies the importance of the occlusions for motion detection and the relation between the observable occlusions and a camera motion. Also, in recent years, some works have shown how compute 3D-structure from camera motion using the matching process among image plane projections by employing the Shift Invariant Feature Transform, SIFT features (Fiala, 2005)(Ohayon, 2006).

This paper is organized as follows: The mathematical principles to understand the camera motion are described in Section II. Section III shows the relationship between the 3D-position of a camera and the position of an object projected in an image captured by it. In Section IV, a strategy to determine zones of occlusion between objects from information of an image is presented and experimental results are shown. Section V describes the process to evaluate and verify the best viewpoint which minimizes the zone of occlusion detected in the image. Finally, the validity of the method proposed is confirmed with a camera at the end of an arm robot.

2 CAMERA MOTION AS RIGID BODY

As starting point a camera moving in front of several objects in a scene can be considered. The camera and objects are modelled as rigid objects. Therefore, each motion can be specified as the motion of one point on the object in respect to another on the camera, the reason being because the distance between any two points which belong to the same object does not change when this object is moved. Consequently, it is not necessary to specify the trajectory of every point on the object in respect to the camera. So, the inertial central moment is the only point we have to consider on the objects, and the optical center is the only point we have to consider on the camera.

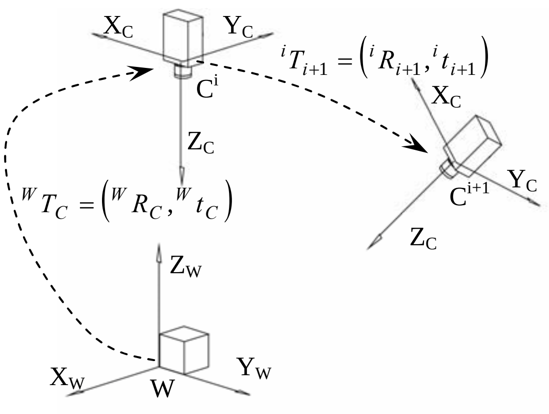


Figure 1: Camera movement relative to world reference frame *W*.

Thus, if C^0 are the coordinates of the camera C in the time i=0, and C^i, the coordinates of the same point on Camera in the time i>0, the geometric transformation which is experimented by camera, is given by:

$$T : R^3 \to R^3 / C^0 \to T \cdot C^i \tag{1}$$

If the camera motion is represented in relation to a world reference frame *W,* and this one is considered fixed, without movement, then the camera motion C is defined by rotational and translational movements which are relative to W in the Euclidean Space. These Euclidean transformations are denoted by ${}^{W}R_C$ and ${}^{W}t_C$, respectively. So, any point which is relative to W can be posed relative to C with this equation.

$$P_W = {}^{W}T_C \cdot P_C = {}^{W}R_C \cdot P_C + {}^{W}t_C \tag{2}$$

where ${}^{M}T_C$ denotes an Euclidean transformation that represents rotation and translation movements of *W* in relation to *C*, and P_C is the point relative to *C*. To this end, the equation 2, is converted to homogeneous representation, appending a '1' to the coordinates of P_C, that is $P_C = \left(P_C, 1\right)^T \in R^4$. Thus a matrix form is used to rewrite it since a linear form is more suitable to operate.

$$P_W = \begin{bmatrix} {}^{W}R_C & {}^{W}t_C \\ 0 & 1 \end{bmatrix} \cdot P_C \tag{3}$$

3 A GEOMETRIC MODEL OF IMAGE FORMATION

In this section, the mathematical model of the image formation process is introduced (Hartley, 2000)(Gruen, 2001). The pinhole camera is chosen as the ideal model to define computer vision processes, and to specify the image formation process, particularly. This process can be described as a set of transformation of coordinates between the camera frame and the world frame (Section 2) and transformations of projection of 3-D object coordinates relative to camera onto 2-D image coordinates. The transformations which determine the movement of a rigid body are defined in Euclidean space, and the transformations which determine the projection onto the image plane are defined in the Projection space.

The pinhole camera model assumes that each 3D-point on an object is projected onto a camera sensor through a point called the optical center. The origin of the reference frame *C* is at this optical center and the coordinates of a 3D-point relative to *C* are denoted by $P_C(X, Y, Z)$. In addition, the coordinates of the same point on the image plane which is projected through the camera sensor are $p_I(x, y)$, where the reference frame image is called *I.* The origin of the reference frame *I* is the principal point *o* which is the intersection point of the optical axis with the image plane, and the parameter *f* defines the distance of image plane from the optical center.

With reference to the pinhole camera model, the transformation between the reference frames *C* and *I* can be represented in homogeneous coordinates and matrices, as follows:

$$Z\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} fX \\ fY \\ Z \end{bmatrix} = \left|\begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}\right| \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \qquad (4)$$

where Z is the depth of the object and f is the focal length.

If the equation is decomposed in two matrices, K_f y Π_0, where the first matrix is called focal matrix, and the second matrix is often referred to as canonical projection matrix, the ideal model can be described as

$$p_I = K_f \cdot \Pi_0 \cdot P_C \qquad (5)$$

However, in practice, the projection transformation between C and I, is very much complex. The matrix K_f depends on others parameters as the size of the pixels, the form of the pixels, etc. Therefore, a method to make the geometric formation image more suitable, is needed. The practice model has to consider (according to (Ma, 2004)):

- The size of pixels. The reason is because the size of the sensor is different to the size of the image. Therefore, the scaling factors (s_x, s_y) must be considered. A point on image plane $p_I(x, y)$ in terms of mm is projected as a point on image $p(u, v)$ in terms of pixels.
- The pixels are not square and do not have orthogonal axis. Therefore, a factor called skew, s_θ, can be used to define the angle between the image axes.
- In addition, the origin of image plane does not coincide with the intersection of the optical axis and the image plane. There is a translational movement between the geometric center on image and the principal point. For this reason the principal point is computed by a calibration process (Zhang, 1999).

If the ideal projection model, pinhole model camera, is modified with these parameters to adapt it to the formation image process and a CCD camera is used, the Equation 5 can be rewritten in the following way:

$$p = K_s \cdot K_f \cdot \Pi_0 \cdot P_C = K \cdot \Pi_0 \cdot P_C \qquad (6)$$

where the triangular matrix, $K=K_s \cdot K_f$, is known as the calibration matrix or intrinsic parameters matrix and its general form is:

$$K = \begin{bmatrix} fs_x & fs_\theta & o_x \\ 0 & fs_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \qquad (7)$$

A calibrated camera with a checkboard pattern has been used in this work. The values for K are: f=21mm, s_x=96.6pixels/mm, s_y=89.3pixels/mm, s_θ=0, o_x=331pixels y o_y=240 pixels.

4 MINIMIZING OCCLUSIONS

The aim of the work presented in this paper is to minimize the zones of occlusions of an object in a scene in which several objects are present. This object has part of its surface occluded by other objects. A camera moving in front of the scene is used to obtain a viewpoint that reduces the occlusion zone of the object desired. Thus, the visibility of the object partially occluded will be improved. This means that more surface of the object desired can be captured by camera.

In the time 0, the initial camera position is given by C^0. On the other hand C^i where i >0 represents the camera position at every moment of time. In addition, the camera position which offers the best viewpoint is represented by C^{i*}. This camera position is the position which minimizes the occlusion zone of the desired object, and which maximizes its visible surface.

In order to avoid modifying the intrinsic parameters matrix of the camera, the space of movements for the camera has been limited. Thus the movement of the camera has been planned like a point which is moved on a hemispheric surface. This way the objects in the scene are located in the center of the hemisphere and the camera can only be located in positions C^i which maintain the same distance to the objects that shown by C^0. Therefore, the distance between camera and objects does not change. This distance is determined by the ratio of the hemisphere, r, which limits the space search of the possible position for the camera. As a result, the camera does not need to be recalibrated because it is not necessary to obtain a new focal length. A study about the suitable movement of a camera into regions sphere is shown in (Wunsch, 1997).

Each camera position is determined by two parameters: length and latitude. Each position, $P_C(X, Y, Z)$ is defined by the displacements in length relative to the initial position, C^0, which is determined by the angle $\theta \in [0, 2\pi]$ and by the displacements in latitude which is determined by the angle $\varphi \in [0, \pi/2]$. This way, it is possible to define any possible position which can be adopted by the camera in the hemispheric search space. The greatest number of positions that the camera can adopt is defined by $\forall C^i / i = 0..\pi^2$. This defines the complete space of camera position. Displacements of 1 degree for length and latitude have been respectively taken.

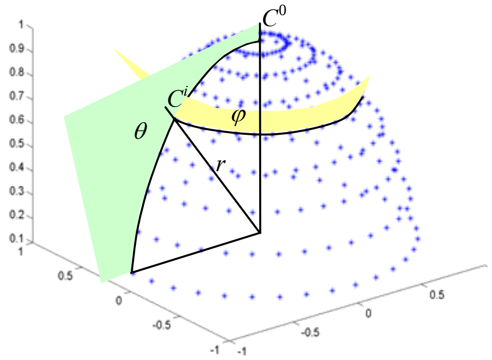


Figure 2: Space of movements for the camera.

The value for each iteration, determined by length and latitude angles, can be modified to increase or to reduce the search space of camera positions. These parameters are chosen depending on the velocity of computation for the analysis of camera positions, and the precision to compute the best camera position for a good viewpoint.

To obtain the best camera position, two parameters have been evaluated: distances and areas. If the zones of occlusion detected in the image must be diminished, the objects in image space must be separated as far as possible.

$$C^{i*} \text{ is } C^i / \max\{d(o_k, o_{k+1})\} \text{ in image i} \qquad (8)$$

Therefore, the first evaluated parameter is the distance between objects. The minimum distance between two objects represented in image space, o_k and o_{k+1} is chosen as the minimum distance between the points of each object. It is described as:

$$d(o_k, o_{k+1}) = \min\{d(\bar{p}_k, \bar{p}_{k+1})\} \qquad (9)$$

where $\bar{p}_k = (p_{k1}, ..., p_{kn})$ is the vector of points which represent the object o_k in space image, and where each point of object $p_{ki} = (u_{ki}, v_{ki}) \in R^2$.

The distance is computed as the length of the line segment between them. Two kinds of distance are used: the distance between the centroid of objects and the distance between the points of edges, which represent the object boundaries. In three-space, the distance does not change because the objects are not in movement. Nevertheless, in image space, the distance changes because the position of an object in relation to another depends on the viewpoint of the camera which is used to capture the image. A comparison of both distance parameters is shown in Figure 3. The distances computed among edge points decreases and converge to zero when an object occludes another. Although, if the distance is computed from the centroids of the segmented region, it can be unstable because when an object occludes another, the first modifies the centroid of the second. The second parameter is the area of each object. This is the visible surface of each object. For the study of the object areas, two cases can be considered.

First case: The viewpoint of camera is not changed and only the location of an object is modified until another is occluded (the movement is in the same orthogonal plane relative to the camera). So, the visible surface of the object occluded has been decreased and the rest of the objects maintain the same area visible. This is shown in Figure 4.

Second case: The viewpoint of the camera is changed and a new perspective is made in the image captured by it. Thus, the area of each object is changed (see Figure 5). But also, when several poses of camera compute a measure of distance, this fact indicates that occlusions are not present.

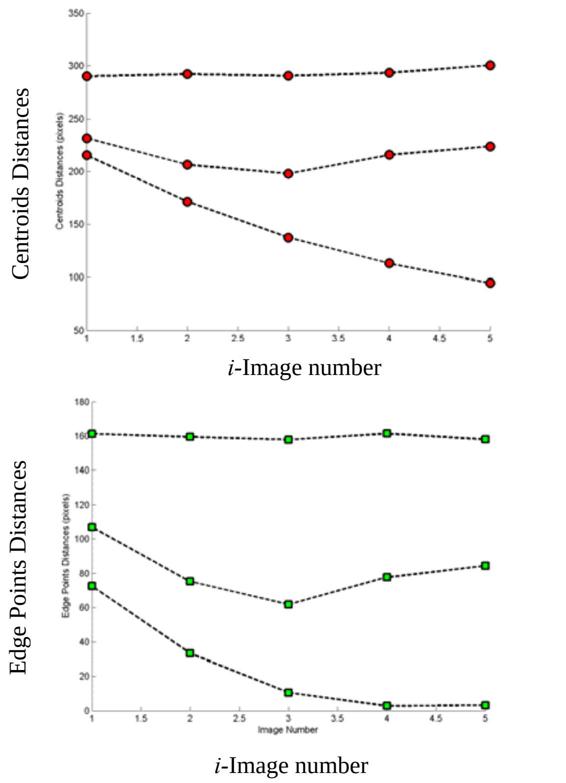


Figure 3: Distances between objects evaluated for the images shown in Figure 4.

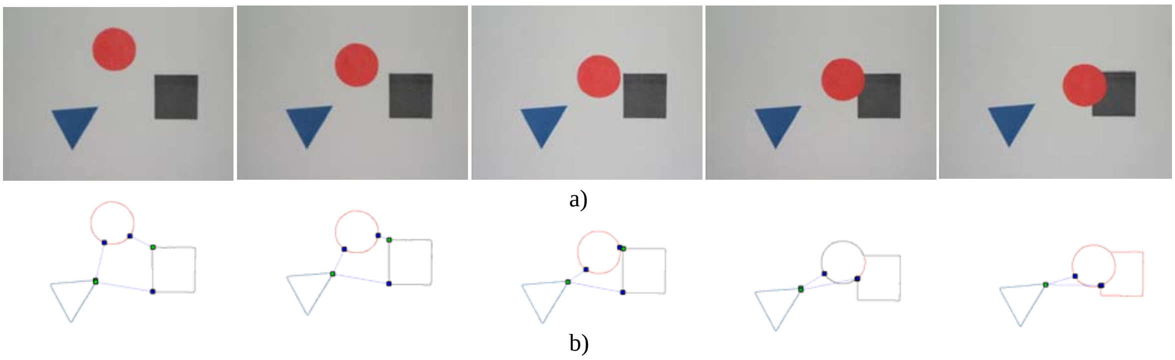


Figure 4: Distances computed among synthetic objects in real images. a) Image sequence with movement of an object. b) Objects segmented from colour and edges, and distances computed from points of edges.

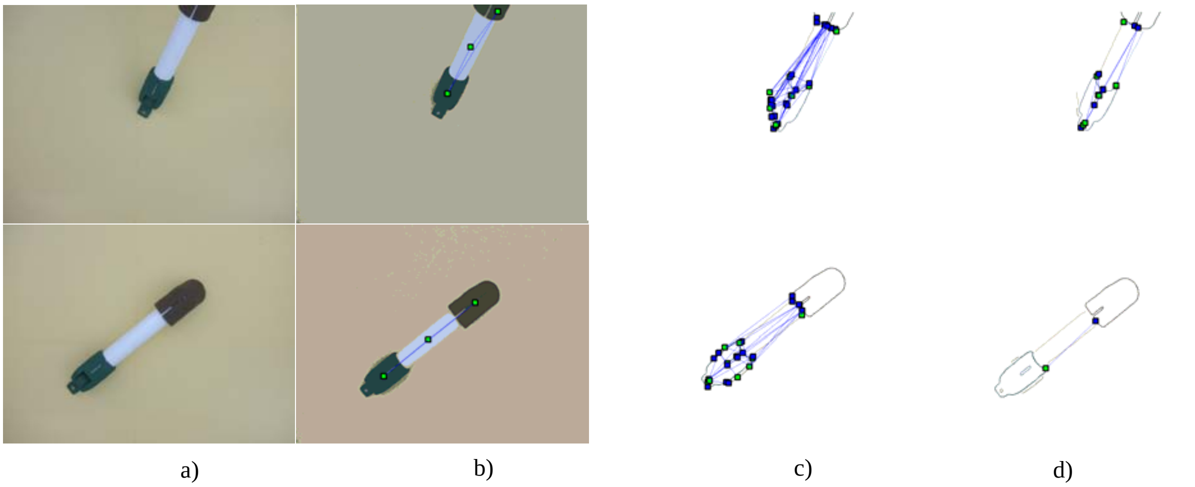


Figure 5: Distances computed among real objects in images. a) An assembly of several objects. b) Distance considering center of gravity. c) Distance considering points of edges. d) Distance considering only optimized points of edges.

Thus, the measure of area of each object must be evaluated and the best pose of camera is one in which the perspective in the image maximizes the sum of areas of each object.

Figure 5b shows the experimental results of applying a colour segmentation process to obtain object regions and, in this way, computing areas (Gil, 2006). In addition, only the segmented regions with a number of pixels major than 5% of pixels in image are taken as objects except the background which is the region with the major number of pixels. For these regions, not only the centroids are computed but also the distances between them. For this experiment, the segmentation process detects 6 regions, however only 3 regions are marked as objects from 3 automatic thresholds by each colour component.

Figure 5c shows the edge segmentation process of colour regions computed in Figure 5b and the distance computed between points of edges belongs to different objects. Finally, Figure 5d shows the distance computed between objects when only optimized edges are considered. The optimized edges are the detected edges which have a number of points major than the standard deviation. The standard deviation determines the measure of variability of the number points which determine an edge from its mean. For this experiment, the detected edges have been 9 and 13 respectively, and the optimized edges 3 and 5 respectively. These edges are approached by segments.

Table 1: Distances between objects computed from the two real views shown in Figure 5 using centroids and points of edge.

Objects	View 1	View 2
d(1,2)	208,129 (129,448)	252,621 (145)
d(1,3)	112,397 (6,403)	123,465 (4,123)
d(2,3)	96,714 (14,422)	129,176 (9,055)

Table 1 shows how the distances computed from centroids are increased if the camera makes the movement shown in Figure 5. Nevertheless, the distances computed from the points of edges can be increased slowly if the real distance between objects is closely near to zero. This fact is due to small instabilities when two real images with different perspective of a same scene are used to obtain segments of boundary. Then, not always the same points are detected in both images.

Although, this is not a problem because the computed distances are always calculated from edges back-projected in virtual images. Therefore, the same points of edges appear in all the virtual images, and only their positions in image are changed.

5 POSE CAMERA TO MINIMIZE OCCLUSION

RGB-colour images with size 640x480 have been used in this work. The steps to explain this work are detailed as follows.

In the first step, an initial image is captured from the initial camera pose, C^i. The 2D-points into initial image are obtained from a colour and edges segmentation process (Gil, 2006). An example of this process has been shown in Figure 5. Next, the distances between objects and its areas are computed for this initial image. Afterwards, for this first image, Equation 3 give the transformation between world coordinates and camera coordinates to obtain 3D-points relative to C^i. Also, the projective matrix, Π, maps 3D-points, relative to C^i, to image points $p(u,v)$ according to Equation 6. Given the points in an image, a set of 3D-points in space that map these points can be determined by means of back-projection of 2D-points. That is:

$$P = \Pi^+ p^i = (K \cdot \Pi_0 \cdot {}^W T_{C^i})^+ p^i \,, \; / \, i = 1..n \qquad (10)$$

where n is number of position for the camera and the pseudo-inverse of Π is the matrix $\Pi^+ = \Pi^T (\Pi\Pi^T)^{-1}$ which verify that $\Pi\Pi^+ = I$. Equation 10 can be rewritten as a homography matrix, so that:

$$P = H_i^{-1} p^i \qquad (11)$$

When the 3D-points are known, the second step consists of computing the projections of the 3D-points which belong to objects from space of camera poses (see Figure 2). This means that the 3D-poins are mapped onto each virtual image for each camera pose, as shown in Figure 6. Thus, virtual 2D-points are computed. It is given by:

$$p^{i+1} = H_{i+1} \cdot P = H_{i+1} H_i^{-1} p^i \qquad (12)$$

These virtual points determine the regions and edges of objects in virtual images. Finally, the distances between objects and areas of each object are computed in each virtual image (Section 4).

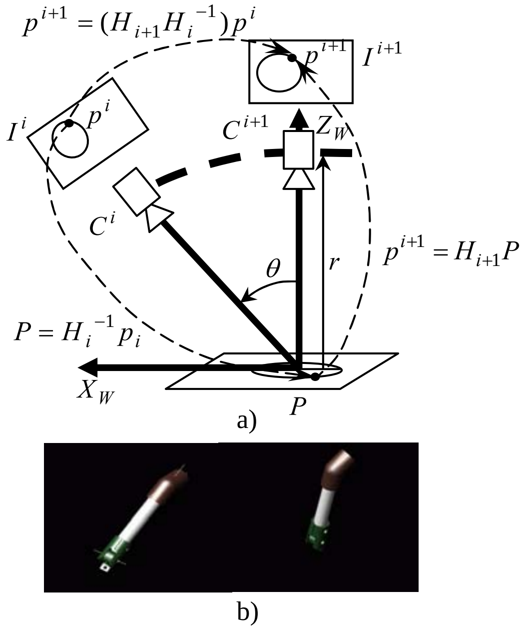


Figure 6: a) Mapping points onto virtual images according to camera movement. A displacement in length has been mapped. b) CAD-Model of assembly used in Figure 5.

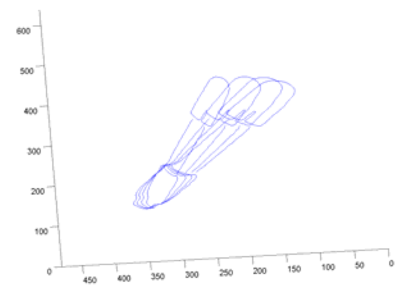


Figure 7: Back-projections onto virtual images from camera movements.

Therefore, a set of ${}^M T_{C^i}$ are evaluated as shown in Figure 2. And the back-projection for each ${}^M T_{C^i}$ is computed (see Figure 7). Concluding, the best pose of the camera is determined by the transformation which maximizes the distances and the areas in the space of virtual images. This transformation is given by the perspective matrix, and it determines what transformations ${}^{W}R_{C^i}$ and ${}^{W}t_{C^i}$ are more suitable. Afterwards, a robot PA-10 from Mitsubishi, with 7 degrees of freedom, moves the camera mounted at its end to more suitable computed pose.

6 CONCLUSIONS

The presented work provides an input to an object recognition process. Thus, a method based on extraction of characteristics in image, which is based on the evaluation of the distances among these characteristics, is used to determine when an occlusion can appear. In addition, the method evaluates the camera pose of a virtual way from the back-projections of the characteristics detected in a real image. The back-projections determine how the characteristics are projected in virtual images defined by different camera poses without the necessity of camera is really moved. The experimental results have shown that the proposed estimation can successfully be used to determine the camera pose that is not too sensitive to occlusions. However, the approach proposed does not provide an optimal solution. This could be solved by applying visual control techniques which are currently under investigation.

Our future work will extend this approach to incorporate visual servoing in camera pose, allowing for a robust positioning camera. A visual servoing system with a configuration ‘eye-in-hand’ can be used to evaluate each camera pose (Pomares, 2006). Thus, the errors can be decreased and the trajectory can be changed during the movement. In addition, the information provided from a model CAD of the objects (see Figure 6b) can be used to verify camera poses in which it is located.

ACKNOWLEDGEMENTS

This work was funded by the Spanish MCYT project “Diseño, implementación y experimentación de escenarios de manipulación inteligentes para aplicaciones de ensamblado y desensamblado automático (DPI2005- 06222)”.

REFERENCES

Boshra, M., Ismail, M.A., 2000. Recognition of occluded polyhedra from range images. *Pattern Recognition.* Vol. 3, No. 8, 1351-1367.

Chan, C.J., Chen S.Y., 2002. Recognition Partially Occluded Objects Using Markov Model. *Int. J. Pattern Recognition and Artificial Intelligence.* Vol. 16, No. 2, 161-191.

El-Sonbaty, Y., Ismael, M.A, 2003. Matching Occluded Objects Invariant to Rotations, Translations, Reflections, and Scale Changes. *Lecture Notes in Computer Science*. Vol. 2749, 836-843.

Fiala, M., 2005. Structure From Motion Using SIFT Features and PH Transform with Panoramic Imagery. *Second Canadian Conference on Computer and Robot Vision.* Victoria, BC, Canada.

Gil, P., Torres, F., Ortiz, F.G., Reinoso, O., 2006. Detection of partial occlusions of assembled components to simplify the disassembly tasks. *International Journal of Advanced Manufacturing Technology.* No. 30, 530-539.

Gruen, A., Huang, T.S., 2001. *Springer Series in Information Sciences. Calibration and Orientation of Cameras in Computer Vision*. Springer-Verlag Berling Heidelberg New York.

Hartley, R., Zisserman, A., 2000. *Multiple View Geometry in Computer Vision*. Cambridge University Press.

Ma, Y., Soato S., Kosecka J., Shankar S., 2004. *An Invitation to 3-D Vision from Images to Geometric Models*. Springer-Verlag, New York Berlin Heidelberg.

Ohba, K., Sato, Y., Ikeuchi, K., 2000. Appearance-based visual learning and object recognition wirh illumination invariance. *Machine Vision and Appplications* 12, 189-196.

Ohayon, S., Rivlin, E., 2006. Robust 3D Head Tracking Using Camera Pose Estimation. *18th International Conference on Pattern Recognition*. Hong Kong.

Park, B.G., Lee K.Y., Lee S.U., Lee J.H., 2003. Recognition of partially occluded objects using probabilistic ARG (attributed relational graph)-based matching. *Computer Vision and Image Understanding* 90, 217-241.

Pomares, J., Gil, P., Garcia, G.J., Torres, F., 2006. Visual-force control and structured Light fusion improve object discontinuities recognition. *11th IEEE International Conference on Emerging Technologies and Factory Automation.* Praga.

Silva, C., Victor, J.S., 2001. Motion from Occlusions. *Robotics and Autonomous Systems* 35, 153-162.

Ying, Z., Castañon, D., 2000. Partially Occluded Object Recognition Using Statical Models. *Int. J. Computer Vision*. Vol. 49, No. 1, 57-78.

Wunsch, P., Winkler S., Hirzinger, G., 1997. Real-Time Pose Estimation of 3-D Objects from Camera Images Using Neural Networks. *IEEE International Conference on Robotics and Automation.*Albuquerque, New Mexico, USA.

Zhang, Z., 2000. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol 22. No. 11, 1330-1334.

ACTIVE 3D RECOGNITION SYSTEM BASED ON FOURIER DESCRIPTORS

E. González, V. Feliú, A. Adán* and Luis Sánchez**
H.T.S. of Industrial Engineering, University of Castilla La Mancha, Ave. Camilo José Cela s/n
Ciudad Real, 13071, Spain
{vicente.feliu,eliza.glez}@uclm.es
**Higher School of Informatics,University of Castilla La Mancha, Ronda de Calatrava 5*
Ciudad Real, 13071, Spain
antonio.adan@uclm.es
***U.E of Technical Industrial Engineering,University of Castilla La Mancha, Ave Carlos III, s/ n*
Toledo, 45071, Spain
luis.sanchez@uclm.es

Keywords: Active recognition system, next best view, silhouette shape, Fourier descriptors.

Abstract: This paper presents a new 3D object recognition/pose strategy based on reduced sets of Fourier descriptors on silhouettes. The method consists of two parts. First, an off-line process calculates and stores a clustered Fourier descriptors database corresponding to the silhouettes of the synthetic model of the object viewed from multiple viewpoints. Next, an on-line process solves the recognition/pose problem for an object that is sensed by a real camera placed at the end of a robotic arm. The method avoids ambiguity problems (object symmetries or similar projections belonging to different objects) and erroneous results by taking additional views which are selected through an original next best view (NBV) algorithm. The method provides, in very reduced computation time, the object identification and pose of the object. A validation test of this method has been carried out in our lab yielding excellent results.

1 INTRODUCTION

Most computer vision systems used in robotics environments perform 3D object recognition tasks using a single view of the scene (Bustos et al., 2005). Commonly, a set of features is extracted and matched with features belonging to an object database. This is why so many researchers focus their recognition strategy on finding features which are capable of discriminating objects efficiently (Helmer and Lowe, 2004). However, these approaches may fail in many circumstances due to the fact that a single 2D image may be insufficient. For instance, this happens when there are objects that are very similar from certain viewpoints in the database (ambiguous objects); a difficulty that is compounded when we have large object databases (Deinzer et al., 2003).

A well known strategy that solves the ambiguity problem is based on using multiple views of the object. Active recognition systems provide the framework to efficiently collect views until the sufficient level of information for developing the identification and posing estimation tasks is obtained (Niku, 2001).

Previous works on active recognition differ in the way they represent objects, the way they combine information and the way of they plan the next observation (Roy et al., 2004). These systems use 3D representation schemes based on the object geometric model or on the object appearance. Although X recognition based on geometric models might be potentially more effective and allow for the identification of objects in any position, they raise important problems of practical aplicability. Moreover, the methods based on X appearance are currently the most successful approaches for dealing with 3D recognition of arbitrary objects.

Many strategies for solving the 3D object recognition problem using multiple views have been proposed: an aspect graph is used in Hutchinson and Kak (1992) to represent the objects. This criterion handles a set of current hypotheses about the object identity and position. It characterizes the recognition ambiguity by an entropy measure (Dempster-Shafer theory) and evaluates the next best sensing operation by minimizing this ambiguity. Borotsching et al.

(1999) represent the objects by some appearance-based information, namely the parametric eigenspace. This representation is augmented by adding some probability distributions. These probability distributions are then used to provide a gauge for performing the view planning. Sipe and Casasent (2002) use a probabilistic extension of the feature space trajectory (FST) in a global eigenspace to represent 3D views of an object. View planning is accomplished by determining - for each pair of objects – the most discriminating view point in an off-line training stage. Their approach assumes that the cost of making a mistake is higher than the cost of moving the sensor.

In general, most of these approaches solve the 3D object recognition problem using stochastic or probabilistic models and, consequently, they require a large dataset for training (Deinzer et al., 2006). Here we present a different way to focus on the problem.

The key to our active recognition system consists of using a reduced set of Fourier descriptors to connect and develop the recognition phases: object representation, classification, identification, pose estimation and next best view planning.

We focus the object representation on silhouettes because: they can be robustly extracted from images, they are insensitive to surface feature variations - such as color and texture - and, finally, they easily encode the shape information (Pope et al. 2005). The most popular methods for 2D object recognition from silhouettes are based on invariant moments or Fourier descriptors. Invariant moments exhibit the drawback that two completely different silhouettes may have the same low order invariant moments, which may lead to ambiguities in the recognition process. Fourier descriptors yield much more information about the silhouette, and only similar silhouettes exhibit similar Fourier descriptors. Since we consider the objects to be non-occluded and the background to be uncluttered, we use a representation scheme in which the silhouettes from different viewpoints are represented by their Fourier descriptors.

This paper is organized as follows. Section 2 presents an overview of the method. Section 3 describes our object identification/pose estimation approach. Section 4 details the next best view method. Section 5 shows the performance of our method by carrying out experiments on a real platform, and some conclusions are stated in Section 6.

2 OVERVIEW OF THE METHOD

In this method the scene silhouette (silhouette of the 3D object to be recognized) is recognized among a set of silhouettes (in our case, 80 or 320 per object) of a group of objects through an algorithm based on Fourier descriptors. Therefore, X recognition of the silhouette of the scene involves both object identification and pose. The method consists of off-line and on-line parts.

The off-line process consists of building a structured database of silhouettes belonging to a generic set of objects. Firstly, a high precision three-dimensional model of each object is obtained by means of a laser scanner sensor. Next, this model is viewed from a set of homogeneous viewpoints obtaining the corresponding set of 2D silhouettes.

The viewpoints correspond to the vertexes of a tessellated sphere with origin in the centre of mass of the object. Figure 1 shows an object model inside the tessellated sphere, the projected image of the model and its silhouette from a specific viewpoint.

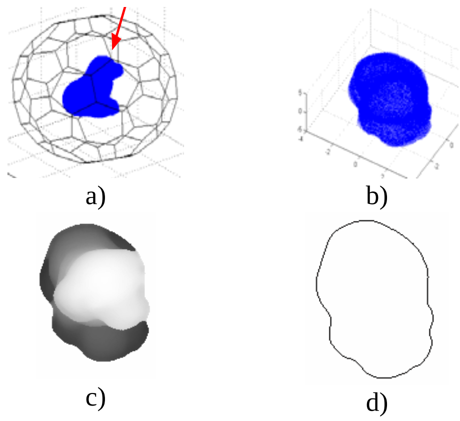
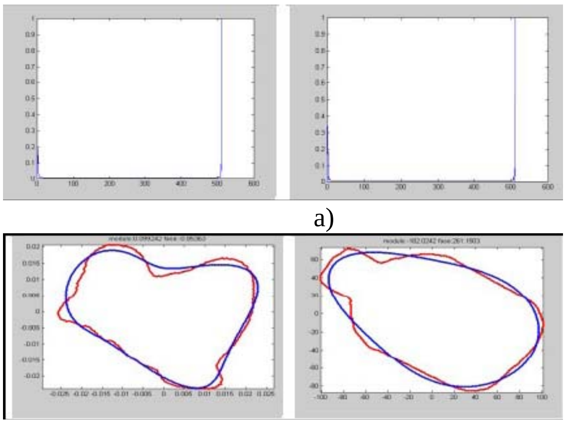


Figure 1: a) Object model put into the tessellated sphere b) View of the object model from a specific viewpoint, c) Depth image d) 2D silhouette.

The database is structured in clusters using only three Fourier descriptors. To build the clustering we used a k-means algorithm (Netanyahu et al., 2002). This strategy allows us to split the silhouette search space in zones where the silhouettes are roughly similar. Consequently, the cost of the recognition process is dramatically reduced. Figure 2 a) shows the most important Fourier descriptors modules for a couple of silhouettes. In our case, we have taken the second, third and next to the last values. Figure 2 b) presents the reconstructed silhouette with the three Fourier descriptors superimposed on the original one. Note that by selecting the most meaningful Fourier components it is possible to work with approximate shapes. Figure 3 shows a spatial representation of the clusters that have been extracted in our database.

The on-line process is designed to solve the recognition/pose problem of an object that is viewed by a camera in a real environment. The essential steps are: Fourier descriptor calculation, classification (discrimination) process, identification/pose calculation and next view algorithm. Next, a brief explanation of these steps is provided.



a)

b)

Figure 2: a) Fourier descriptors modules b) silhouette (red) and silhouette recovered with three Fourier descriptors (blue).

To calculate the Fourier descriptors a suitable image preprocessing is carried out on the original image. Specifically this process consists of filtering, thresholding and contour extraction. Next the points of the contour are taken as a sequence of complex numbers and the Fourier descriptors are finally computed.

The discrimination phase classifies the silhouette of the scene into a single or a set of clusters. The selected clusters constitute the work sub-space in the pose phase. Formally,

Let $O_1, O_2, ...O_N$ a database of N objects, C_k the *kth* cluster, $k \in [1..K]$, S_k^{nm} the *n-th* silhouette of the object m, p_k the *k-th* cluster prototype, D the Euclidean distance, R_k the *k-th* cluster radius where $R_k = \max D(p_k, S_k^{nm})$ and z the silhouette of the scene to be matched. The subspace S_{sub} will be formed by the clusters, which verify one or both of the following conditions:

Criterion 1: If $D(p_k, z) < R_k$ *then* $C_k \in S_{sub}$

Criterion 2:
If $\cdot| D_i - \min D_k | < \varepsilon$ then $C_i \in S_{sub}, i \in [1..K]$

The criterion 1 is satisfied for cases where z is inside a cluster whereas criterion 2 corresponds to cases where the silhouette z is inside an area with very high cluster density or where the scene silhouette falls outside the clusters. Thus, the discrimination process sets a work subspace S_{sub} with a reduced database of silhouettes.

The identification phase, which is carried out in S_{sub}, yields, in general, a reduced set of candidate silhouettes. The reason for taking only a few candidates is as follows. Matching and alignment techniques based on contour representations are usually effective in 2D environments. Nevertheless, in 3D environments these techniques have serious limitations. The main problems with the contour based techniques occur due to the fact that the information on the silhouettes may be insufficient and ambiguous. Thus, similar silhouettes might correspond to different objects from different viewpoints. Consequently, a representation based on the object contour may be ambiguous, especially when occlusion circumstances occur.

In essence, the identification phase compares the silhouette from the scene with the silhouettes in the subspace S_{sub} by means a quadratic error minimization applied on the modulus of the Fourier descriptors. If the identification/pose process proposes more than one candidate silhouette, then the solution is ambiguous and it is necessary to apply the Next Best View planning method (NBV). Figure 4 shows a scheme with the main process.

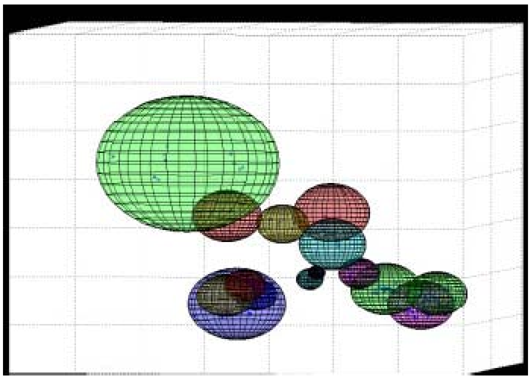


Figure 3: Spatial representation of the silhouette clusters.

In most cases, the recognition and pose estimation phase is finished after several views of the object are taken and only one candidate is found. In this process, the position of the next view is calculated through an algorithm based on the set of candidate silhouettes obtained in the previous view. This will be explained in Section IV. Figure 4 shows a scheme of the main process.

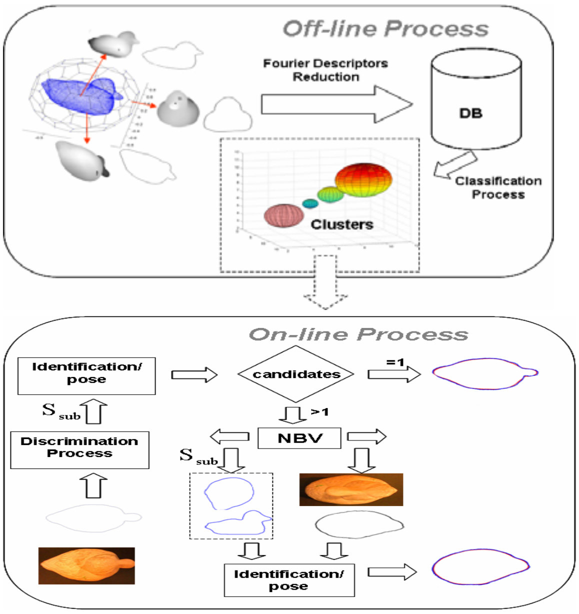


Figure 4: Diagram of the active recognition system.

3 OBJECT RECOGNITION AND POSE ESTIMATION PROCEDURE

Fourier descriptors can be used to represent closed lines (silhouettes). They can be made invariant to translations and rotations, and allow easy filtering of image noise (Deinzer et al., 2003). Assume a contour *l(n)* composed of N points on the XY plane:

$$l(n) = [x(n), y(n)], \quad n = 0..N-1 \tag{1}$$

where the origin of index n is an arbitrary point of the curve, and n and n+1 are consecutive points according to a given direction (for example clockwise direction) over the silhouette. Assume also that points over the curve have been regularized in the sense that two consecutive points are always at the same Euclidean distance. Let us define the complex sequence and its discrete Fourier transforms Z(k)=F(z(n))as:

$$z(n) = x(n) + jy(n) \tag{2}$$

$$Z(k) = \sum_{n=0}^{N-1} z(n)\exp(-j2\pi kn/N) \quad 0\leq k\leq N-1 \tag{3}$$

Assume also a data base of R silhouettes $s_r(n)$, $0 \leq n \leq N_r - 1$ with $1 \leq r \leq R$, whose respective discrete Fourier transforms are $S_r(k)$.

A critical aspect of our method is its computation speed because we want to recognize objects in real time. Then the FFT algorithm is used to obtain the Fourier descriptors. Then N must be power of 2 and both the scene silhouette and the silhouettes of the data base must be regularized to a number of $N = 2^{N'}$ points.

The basic problem to be solved in our method is to match the scene silhouette *z(n)* to some silhouette $s_{r^*}(n)$ of the data base, under the assumptions that *z(n)* may be scaled (λ), translated ($c_x + jc_y$) and rotated (φ) with respect to the best matching silhouette of the data base, and that the reference point on *z(n)* (n = 0) may be different from the reference point of that data base silhouette (we denote that displacement δ). The next section deals with selecting the silhouettes and obtaining X c, δ, φ, λ.

3.1 Close Silhouettes Selection

Suppose that z is the silhouette of an object captured by the camera and that it corresponds to the silhouette $s_{r^*}(n)$ that belongs to the silhouette database. In general, z is matched to $s_{r^*}(n)$ after displacement, rotation, scaling and centre translation parameters are found. In general for $s_r(n)$, in the space domain:

$$z(n) = D(s_r(n), \delta)\lambda \exp(j\varphi) + c \tag{4}$$

where $D(s_r(n))$ displaces δ units the origin of the sequences $s_r(n)$. Taking Fourier transform:

$$Z(k) = \lambda \exp(j\varphi)\exp(j2\pi k\delta/N)S_r(k) + cN \tag{5}$$

- Translation: Since all silhouettes have the coordinate origin at their centre of mass $S_r(0) = 0$ and from expression (5) , $c = Z(0)/N$.
- Close silhouettes identification.: Defining $\hat{Z}(k) = Z(k) - cN$, the modulus of expression (5) holds:

$$|\hat{Z}(k)| = \lambda |S_r(k)| \tag{6}$$

The matching procedure minimizes the mean squared error between the Fourier descriptors of the scene silhouette and the silhouettes of the database. Given a pair of silhouettes $(z(n), s_r(n))$, the similarity index J_r is defined as:

$$J_r(\lambda) = (|\hat{Z}| - \lambda|S_r|)^t (|\hat{Z}| - \lambda|S_r|) \tag{7}$$

where $|\hat{Z}| = (|\hat{Z}(0)|, \ldots |\hat{Z}(N-1)|)^t$, $|S_r| = (|S_r(0)| \ldots |S_r(N-1)|)^t$, $|.|$=absolute value.

Minimizing $J_r(\lambda)$ with respect to the scaling factor λ, we obtain:

$$\lambda_r^\circ = \frac{\left|\hat{Z}\right|^t |S_r|}{|S_r|^t |S_r|} \quad , \; J_r^\circ = \left|\hat{Z}\right|^t \left|\hat{Z}\right| - \frac{(\left|\hat{Z}\right|^t |S_r|)^2}{|S_r|^t |S_r|} \tag{8}$$

After calculating J_r° for all silhouettes of the data base we select the silhouettes which verify $J_r^\circ \le U$, U being a specific threshold. In this case, we have a set of many ambiguous silhouettes $\{S_{c1}, S_{c2}, \ldots S_{cf}\}$ and it is necessary to select another viewpoint to solve the ambiguity problem (NBV section).

3.2 Pose Calculation

Let us denote $L = (r_1, r_2, \ldots, r_m)$ the set of indexes of the candidate silhouettes. In order to select the best candidate among L candidates, a more accurate procedure is carried out. This procedure uses the complete complex Fourier descriptors (not only the modules as in the previous process). As a result of this process, a new similarity index f° is obtained and the pose parameters λ, φ, δ are calculated.

The cost function to be minimized is (see (4)):

$$f\;(\lambda,\theta,\delta) = (\hat{Z} - q\,P\,(\delta))^{t^*}\;(\hat{Z} - q\,P\,(\delta)) \tag{9}$$

where

$\hat{Z} = (\hat{Z}(0), \ldots, \hat{Z}(N-1))^t$,

$P_r(\delta) = (s_r(0), s_r(1)\exp(j2\pi\delta/N), \ldots, s_r(N-1)$
$\exp(j2\pi\delta(N-1)/N))^t$,

$*$ denotes conjugate, t^* denotes transpose conjugate, and r is restricted now to set L

Let us denote $\lambda\exp(j\varphi) = q$, optimizing (9) respect to the complex number q:

$$q^\circ(\delta) = \frac{Z^t\;P^*(\delta)}{S^{t^*}\;S},\; f^\circ(\delta) = Z^{t^*}\;Z - \frac{\left|Z^{t^*}\;P(\delta)\right|^2}{S^{t^*}\;S} \tag{10}$$

(notice that $P_r^{t^*}(\delta)\;P_r(\delta) = S_r^{t^*} \cdot S_r$).

Taking into account that $0 \le \delta \le N-1$ is integer, the right expression of (10) is calculated for all possible values of δ and $f_r^\circ = \min_\delta f_r^\circ(\delta)$ is determined.

Then f_r° is the similarity index of the silhouette r in the fine matching process, δ_r° is the corresponding displacement and q_r° is obtained from ¿the? left equation of (10) particularized to δ_r°. Rotation and scaling are estimated from q_r°:

$$\lambda_r^\circ = \left|q_r^\circ\right|\;;\; \varphi_r^\circ = \angle q_r^\circ \tag{11}$$

4 NEXT BEST VIEW PLANNING

The goal of this phase is to provide a solution to the ambiguity problem by taking a set of optimal viewpoints. When an ambiguous case occurs, we move the camera to another viewpoint from which the silhouettes of the candidate objects are theoretically very dissimilar.

As said before, in our scheme representation we associate each silhouette stored in the database with a viewpoint of the tessellated sphere. Then, the first step in the NBV consists of aligning the candidate spheres (corresponding to the viewpoints in our models) with the scene sphere.

Let $T_R(S)$ be the tessellated sphere, N'_{Rx} the camera position and N_{R1} the viewpoint that corresponds to the candidate silhouette S_{ci}. To align the two spheres a rotation must be applied to make N'_{Rx} and N_{R1} coincident (Adán et al., 2000). Formally:

Let $\vec{\mathbf{u}}(u_x, u_y, u_z) = \dfrac{\overrightarrow{ON}_{R1} \times \overrightarrow{ON'}_{Rx}}{\left|\overrightarrow{ON}_{R1} \times \overrightarrow{ON'}_{Rx}\right|}$ be the normal vector to the plane defined by $\overrightarrow{ON}_{R1}$ and $\overrightarrow{ON'}_{Rx}$, O being the center of T_I. Let θ be the angle between the last two vectors. Then, a rotation θ around the u axis can first be applied to $T_R(S)$. This spatial transformation is defined by the following rotation matrix $\mathbf{R_u}(\theta)$:

$$\mathbf{R_u}(\theta) = \begin{pmatrix} u_xu_x(1-c\theta)+c\theta & u_xu_y(1-c\theta)-u_zs\theta & u_xu_z(1-c\theta)+u_ys\theta \\ u_xu_y(1-c\theta)+u_zs\theta & u_yu_y(1-c\theta)+c\theta & u_yu_z(1-c\theta)-u_xs\theta \\ u_xu_z(1-c\theta)-u_ys\theta & u_yu_z(1-c\theta)+u_xs\theta & u_zu_z(1-c\theta)+c\theta \end{pmatrix} \tag{12}$$

where $c\theta = \cos\theta$ and $s\theta = \sin\theta$.

A second rotation φ around the axis $\vec{\mathbf{v}}(v_x, v_y, v_z) = \dfrac{\overrightarrow{ON'}_{Rx}}{\left|\overrightarrow{ON'}_{Rx}\right|}$ is required to achieve the best fitting of $T_R(S)$ to $T_R(S')$ (see Figure 5). The swing angle φ is determined by (14). This last set of points can be obtained by applying a rotation matrix Rv(φ) that depends on a single parameter φ and can be formally expressed as:

$$\mathbf{R}_v(\varphi)=\begin{pmatrix} v_xv_x(1-c\varphi)+c\varphi & v_xv_y(1-c\varphi)-v_zs\varphi & v_xv_z(1-c\varphi)+v_ys\varphi \\ v_xv_y(1-c\varphi)+v_zs\varphi & v_yv_y(1-c\varphi)+c\varphi & v_yv_z(1-c\varphi)-v_xs\varphi \\ v_xv_z(1-c\varphi)-v_ys\varphi & v_yv_z(1-c\varphi)+v_xs\varphi & v_zv_z(1-c\varphi)+c\varphi \end{pmatrix} \quad (13)$$

$$R_1(\varphi,\theta)=R_v(\varphi)\cdot R_u(\theta) \quad (14)$$

Finally the alignment of the spheres is:

$$\mathrm{T}_{\mathrm{R}}^{'}(S)=\mathrm{R}_1(\varphi,\theta)\cdot\mathrm{T}_{\mathrm{R}}(S) \quad (15)$$

In the next step the Fourier energy is calculated for each viewpoint.

Defining $S_{cp} \in O_i$, $S_{cq} \in O_j$ where $S_{cp}, S_{cq} \in [S_{c1}, S_{C2} ... S_{cf}]$, where f is the number of candidate silhouettes, and vp is a viewpoint from $T_R(S)$. The energy is computed for all couples of silhouettes as follows:

$$E_{Oi,Oj}^{vp}=\frac{1}{N}\sum_{k=0}^{N-1}|Z_{Oi}^{vp}(k)- \quad Z_{Oj}^{vp}(k)|^2, \quad \forall i\neq j, i\leq f, j\leq f \quad (16)$$

$$E^{vp}=\min(E_{Oi,Oj}^{vp}) \quad (17)$$

The NBV ν is defined as the viewpoint that verifies:

$$E^{\nu}=\max(E^{vp})\ \forall vp \quad (18)$$

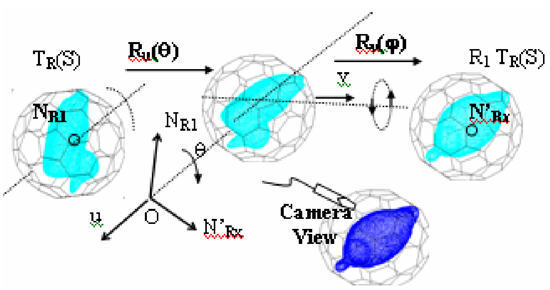


Figure 5: Alignment process between candidate spheres and the scene sphere.

Finally, the camera is moved to the best viewpoint and a new image of the scene is captured and matched with the model silhouette correspondents to the best viewpoint using (9) and (11) equations.

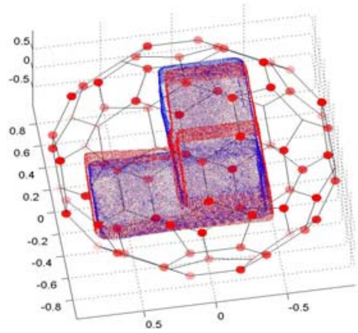


Figure 6: Superimposed models after the alignment process and energy values plotted on the nodes.

5 EXPERIMENTATION

A validation test of this method has been carried out in our lab. The experimental setup is composed of a Stäubli RX 90 Robot with a micro camera Jai-CVM1000 at its end. This system controls the position and vision direction on the camera, the object always being centered in the scene. Figure 7 shows the experimental setup.

In the off-line process, the synthesized models (with 80000 polygons/object) are built through a VI-910 Konica Minolta 3D Laser scanner. At the same time the silhouette database with their respective Fourier descriptors are obtained and stored. Currently we have used databases of 80 and 320 silhouettes/model.

In order to reduce redundant information and to optimize the recognition/pose time, a Fourier descriptors reduction has been carried out on the silhouette models. Figure 2.a shows Fourier descriptors modulus for an example. As X can be seen, the first and the last descriptors are the most meaningful. The reduction procedure consists of considering the intervals [1,X(k)], [X(N-k),X(N)], k=1,…N, until the error/pixel between the original and reduced silhouettes is less than a threshold χ.

a) b)

Figure 7: a) Experimental Setup. b) Examples of synthetic models.

The experimentation has been carried out with 19 objects that have been previously modelled with 80 and 320 views, considering descriptor reductions of χ =0.05 and χ =0.5.

In the clustering process we have used 50 clusters. During this phase we use images with resolution 640x480. Each silhouette in the database was stored with a resolution of 512 points and the database size was 12 MB (80 views) and 42 MB (320 views).

Table 1.

# sil.	χ	t_A	ρ_A	t_B	ρ_B
80	0.05	2.652	1.640	2.475	2.382
	0.5	2.047	1.653	1.863	2.473
320	0.05	4.901	1.089	4.739	2.107
	0.5	3.336	1.339	3.096	2.261

The active 3D recognition system worked in all tests achieving 100% X effectiveness. Table 1 shows the results obtained during the recognition process without (A) and with (B) discrimination phase. The results are compared taking into account: the number of silhouettes of the model, threshold for Fourier descriptors reduction (χ), mean square error between the silhouette of the scene and the estimated silhouette (ρ). Variable t is the computation time (seconds) on a Pentium III 800 Hhz processor.

Table II shows in detail the main process rates using a database with 80 silhouettes/model and a reduction factor χ =0.5.

From Tables 1 and 2 the following comments can be made.

- In the whole process, most of the time is devoted to extracting the object's silhouette (88,6% and 94,7%). Note that, this stage includes several image preprocessing tasks like filtering, thresholdin, etc. In part, such a high percentage is also due to the fact that we have used a reduced object database in our experimentation. For large databases (>100-500 objects) this percentage will decrease at the same time that the percentage corresponding to the candidates selection stage will increase.
- Using 320 silhouettes per model increases in a double the execution times with respect the use of 80 silhouettes per model but the ρ decreases by 0,3 percent.

Table 2.

	Algorithm	time (%)
Without clustering	Silhouette extraction	88.6
	Identification	7.9
	Pose estimation	1.4
	NBV	2.1
With clustering	Silhouette extraction	94.7
	Discrimination	0.8
	Identification	2.2
	Pose estimation	0.6
	NBV	1.7

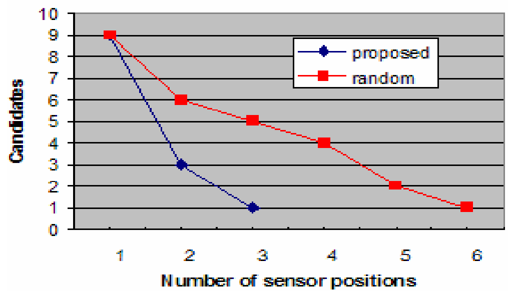


Figure 8: Comparison of discrimination between a random method and our proposed method.

Two experiments were carried out: one running our active recognition system which uses a random selection of the next view, and another computing the next best view from our D-Sphere structure. In Figure 8 we can see the number of candidates in each sensor position for a real case. The test average reported that our method considerably reduced the number of sensor movements: about 62%. The time needed to calculate the next observation position is very short: approximately 1.7% of all the time needed to carry out a complete step of the recognition process. Calculations were performed on a Pentium III 800 Hhz processor. The active 3D recognition system worked in all tests achieving 100% X effectiveness. Figure 9 illustrates a case of ambiguity between two objects and how the system solves the problem. Our NBV method shows much higher discriminative capability than the random method. Thus, the proposed strategy significantly improves the recognition efficiency.

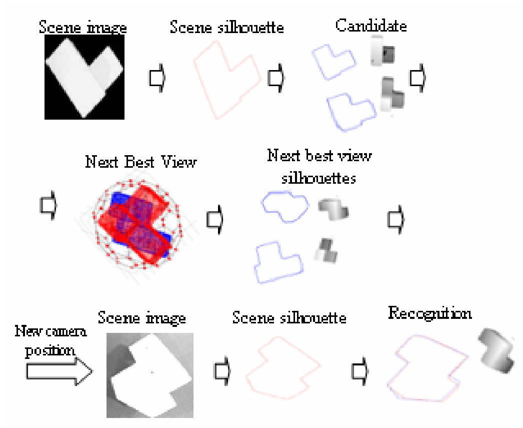


Figure 9: Solving an ambiguous case.

Figure 10 presents some matching and pose estimation results using the proposed algorithm.

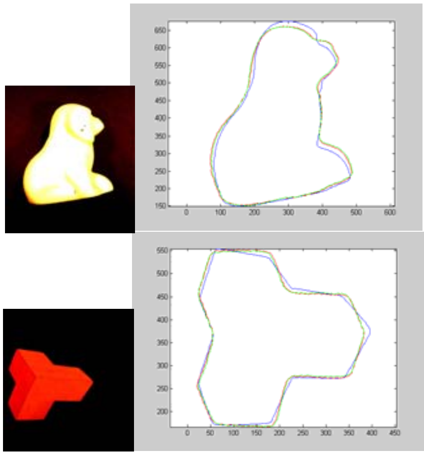


Figure 10: Some examples of 3D recognition without ambiguity.

6 CONCLUSION

This paper has presented a new active recognition system. The system turns a 3D object recognition problem into a multiple silhouette recognition problem where images of the same object from multiple viewpoints are considered. Fourier descriptors properties have been used to carry out the clustering, matching and pose processes.

Our method implies the use of databases with a very large number of stored silhouettes, but an efficient version of the matching process with Fourier descriptors make it possible to solve the object recognition and pose estimation problems in a greatly reduced computation time.

On the other hand, the next best view (NBV) method efficiently solves the frequent ambiguity problem in recognition systems. This method is very robust and fast, and is able to discriminate among very close silhouettes.

REFERENCES

Adán, A., Cerrada, C., Feliu, V., 2000. Modeling Wave Set: Definition and Application of a New Topological Organization For 3D Object Modeling. Computer Vision and Image Understanding 79, pp 281-307.

Bustos, B., Kein, D.A., Saupe, D., Schreck, T., Vranic, D., 2005. Feature-based Similarity Search in 3D Object Databases. ACM Computing Surveys (CSUR) 37(4):345-387, Association For Computing Machinery.

Borotschnig, H. Paletta, L., Pranti, M. and Pinz, A. H. 1999. A comparison of probabilistic, possibilistic and evidence theoretic fusion schemes for active object recognition. Computing, 62:293–319.

Deinzer, F., Denzler, J. and Niemann, H., 2003. Viewpoint Selection. Planning Optimal Sequences of Views for Object Recognition. In Computer Analysis of Images and Patterns, pages 65-73, Groningen, Netherlands, Springer.

Deinzer, F., Denzler, J., Derichs, C., Niemann, H., 2006: Integrated Viewpoint Fusion and Viewpoint Selection for Optimal Object Recognition. In: Chanteler, M.J. ; Trucco, E. ; Fisher, R.B. (Eds.) : British Machine Vision Conference

Helmer S. and Lowe D. G, 2004. Object Class Recognition with Many Local Features. In Workshop on Generative Model Based Vision 2004 (GMBV), July.

Hutchinson S.A. and Kak. A.C., 1992. Multisensor Strategies Using Dempster-Shafer Belief Accumulation. In M.A. Abidi and R.C. Gonzalez, editors, Data Fusion in Robotics and Machine Intelligence, chapter 4, pages 165–209. Academic Press,.

Netanyahu, N. S., Piatko, C., Silverman, R., Kanungo, T., Mount, D. M. and Wu, Y., 2002. An efficient k-means clustering algorithm: Analysis and implementation. vol 24, pages 881–892, july.

Niku S. B., 2001. Introduction to Robotics, Analysis, Systems, Applications. Prentice Hall.

Poppe, R.W. and Poel, M., 2005. Example-based pose estimation in monocular images using compact fourier descriptors. CTIT Technical Report series TR-CTIT-05-49 Centre for Telematics and Information Technology, University of Twente, Enschede. ISSN 1381-3625

Roy, S.D., Chaudhury, S. and Banerjee. S., 2004. Active recognition through next view planning: a survey. Pattern Recognition 37(3):429–446, March.

Sipe M. and Casasent, D., 2002. Feature space trajectory methods for active computer vision. IEEE Trans PAMI, Vol. 24, pp. 1634-1643, December.

USING SIMPLE NUMERICAL SCHEMES TO COMPUTE VISUAL FEATURES WHENEVER UNAVAILABLE

Application to a Vision-Based Task in a Cluttered Environment

FOLIO David and CADENAT Viviane
LAAS/CNRS, 7 avenue du Colonel ROCHE,Toulouse, France
Paul SABATIER University, 118 route de Narbonne, Toulouse, France
dfolio@laas.fr, cadenat@laas.fr

Keywords: Visual features estimation, visual servoing, collision avoidance.

Abstract: In this paper, we address the problem of estimating image features whenever they become unavailable during a vision-based task. The method consists in using numerical algorithm to compute the lacking data and allows to treat both partial and total visual features loss. Simulation and experimental results validate our work for two different visual-servoing navigation tasks. A comparative analysis allows to select the most efficient algorithm.

1 INTRODUCTION

Visual servoing techniques aim at controlling the robot motion using visual features provided by a camera and require that image features remain always visible (Corke, 1996; Hutchinson et al., 1996). However, different problems may occur during the execution of a given task: visual features loss or occlusions, camera failure, and so on. In such cases, the above mentioned techniques cannot be used anymore and the corresponding task will not be achieved. Thus the visual features visibility during a vision-based task appears as an interesting and challenging problem which must be addressed. Classically, the proposed solutions aim at *avoiding* occlusions and loss. Most of these solutions are dedicated to manipulator arms because such robotic systems offer a sufficient number of degrees of freedom (DOF) to benefit from redundancy to treat this kind of problem (Marchand and Hager, 1998; Mansard and Chaumette, 2005). Other techniques preserve visibility by path-planning in the image (Mezouar and Chaumette, 2002), by acting on specific DOFs (Corke and Hutchinson, 2001; Malis et al., 1999; Kyrki et al., 2004), by controlling the zoom (Benhimane and Malis, 2003) or by making a tradeoff with the nominal vision-based task (Remazeilles et al., 2006). In a mobile robotic context, when executing a vision-based navigation task in a cluttered environment, it is necessary to preserve not only the visual features visibility but also the robot safety. A first answer to this double problem has been proposed in (Folio and Cadenat, 2005a; Folio and Cadenat, 2005b). The developed methods allow to avoid collisions, occlusions and target loss when executing a vision-based task amidst obstacles. However they are restricted to missions where it is possible to avoid both occlusions and collisions without leading to local minima. Therefore, a true extension of these works would be to provide methods which accept that occlusions may effectively occur. A first solution is to allow *some* of the features to appear and disappear temporarily from the image as done in (Garcia-Aracil et al., 2005). However, this approach is limited to partial occlusions. Another solution which is considered in this paper is to compute the visual features as soon as some or all of them become unavailable. Total visual features loss can then be specifically treated.

The paper is organized as follows. In section 2, we propose a method allowing to compute the visual features when they become unavailable. Section 3 describes the application context, and shows some simulation and experimental results validating our work.

2 VISUAL DATA ESTIMATION

In this section, we address the visual data estimation problem whenever they become unavailable. We first introduce some preliminaries and state the problem before presenting our estimation method.

2.1 Preliminaries

We consider a mobile camera and a vision-based task with respect to a static landmark. We assume that the camera motion is holonomic, characterized by its kinematic screw: $\mathcal{T}_c = (V_{C/\mathcal{F}_0}^{\mathcal{F}_c\,T}, \Omega_{\mathcal{F}_c/\mathcal{F}_0}^{\mathcal{F}_c\,T})^T$ where $V_{C/\mathcal{F}_0}^{\mathcal{F}_c} = (V_{X_c}, V_{Y_c}, V_{Z_c})^T$ and $\Omega_{\mathcal{F}_c/\mathcal{F}_0}^{\mathcal{F}_c} = (\Omega_{X_c}, \Omega_{Y_c}, \Omega_{Z_c})^T$ represent the translational and rotational velocity of the camera frame $\mathcal{F}_c$ with respect to the world frame $\mathcal{F}_0$ expressed in $\mathcal{F}_c$ (see figure 1).

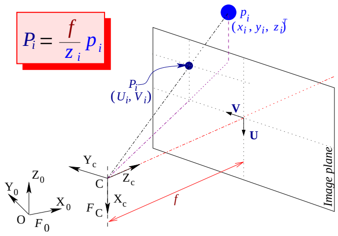


Figure 1: The pinhole camera model.

Remark 1 *We do not make any hypothesis about the robot on which is embedded the camera. Two cases may occur: either the robot is holonomic and so is the camera motion; or the robot is not, and we suppose that the camera is able to move independantly from it.*

Now, let s be a set of visual data provided by the camera, and $\mathbf{z}$ a vector describing its depth. For a fixed landmark, the variation of the visual signals $\dot{s}$ is related to $\mathcal{T}_c$ by means of the interaction matrix $\mathcal{L}$ as shown below (Espiau et al., 1992):

$$\dot{s} = \mathcal{L}(s, \mathbf{z})\,\mathcal{T}_c \tag{1}$$

This matrix allows to link the visual features motion in the image to the 3D camera motion. It depends mainly on the depth $\mathbf{z}$ (which is not always available on line) and on the considered visual data. We suppose in the sequel that we will only use image features for which (1) can be computed analytically. Such expressions are available for different kinds of features such as points, straight lines, circles (Espiau et al., 1992), or image moments (Chaumette, 2004)...

2.2 Problem Statement

Now, we focus on the problem of estimating (*all* or some) visual data s whenever they become unavailable. Different approaches, such as tracking methods and signal processing techniques, may be used to deal with this kind of problem. Here, we have chosen to use a simpler approach for several reasons. First of all, most tracking algorithms relies on measures from the image which is unavailable in our case. Second, as it is intended to be used to perform complex navigation tasks, the estimated visual signals must be provided sufficiently rapidly with respect to the control law sampling period. Finally, in our application, the initial value of the visual features to be estimated is always known, until the image become unavailable. Thus, designing an observer or a filter is not necessary, as this kind of tools is mainly interesting when estimating the state of a dynamic system whose initial value is unknown. Another idea is to use a 3D model of the object together with projective geometry in order to deduce the lacking data. However, this choice would lead to depend on the considered landmark type and would require to localize the robot. This was unacceptable for us, as we do not want to make any assumption on the visual feature model. Therefore, we have chosen to solve numerically equation (1) on the base of the visual signals previous measurements and of the camera kinematic screw $\mathcal{T}_c$.

As a consequence, our method will lead to closed loop control schemes for any task where the camera motion is defined independently from the image features. This will be the case for example when executing a vision-based task in a cluttered environment if the occlusion occurs in the obstacle neighborhood, as shown in section 3. However, for "true" vision-based tasks where $\mathcal{T}_c$ is directly deduced from the estimated visual signals, the obtained control law remains an open-loop scheme. Therefore, it will be restricted to occlusions of short duration and when there is small perturbation on the camera motion. Nonetheless, in the context of a sole visual servoing navigation task, this approach remains interesting as it appears as an *emergency* solution when there is no other mean to recover from a complete loss of the visual data. This method can also be used to predict the image features between two image acquisitions. It is then possible to compute the control law with a higher sampling period, and it is also quite interesting in our context.

Now, let us state our problem. As equation (1) depends on depth $\mathbf{z}$, it is necessary to evaluate this parameter together with the visual data s. Therefore, our method requires to be able to express analytically the variation $\mathbf{z}$ with respect to the camera motion. Let us suppose that $\dot{\mathbf{z}} = \mathcal{L}_{\mathbf{z}}\mathcal{T}_c$. Using relation (1) and denoting by $\mathbf{X} = \left(s^T, \mathbf{z}^T\right)^T$, the differential equations to be solved can be expressed as:

$$\left\{\begin{array}{lcl} \dot{\mathbf{X}} & = & \left(\begin{array}{cc} \mathcal{L}^T & \mathcal{L}_{\mathbf{z}}^T \end{array}\right)^T \mathcal{T}_c = \mathbf{F}(\mathbf{X}, t) \\ \mathbf{X}(t_0) & = & \mathbf{X}_0 = \left(s_0^T, \mathbf{z}_0^T\right)^T \end{array}\right. \tag{2}$$

where $\mathbf{X}_0$ is the initial value of $\mathbf{X}$, which can be considered known as s_0 is directly given by the feature extraction processing and $\mathbf{z}_0$ can be usually characterized off-line. Finally, for the problem to be well stated, we assume that $\mathcal{T}_c$ has a very small variation during each integration step $h = t_{k+1} - t_k$ of (2).

We propose to use numerical schemes to solve the Ordinary Differential Equations (ODE) (2). A large overview of such methods is proposed for example in (Shampine and Gordon, 1975). In this work, our objective is to compare several numerical schemes (Euler, Runge-Kutta, Adams-Bashforth-Moulton (ABM) and Backward Differentiation Formulas (BDF)) to select the most efficient technique.

3 APPLICATION

We have chosen to apply the considered numerical algorithms in a visual servoing context to compute the visual features when they are lost or unavailable during a navigation task. We have considered two kinds of missions: the first one is a positioning vision-based task during which a camera failure occurs; the second one is a more complex mission consisting in realizing a visually guided navigation task amidst obstacles despite possible occlusions and collisions.

After describing the robotic system, we present the two missions and the obtained results.

3.1 The Robotic System

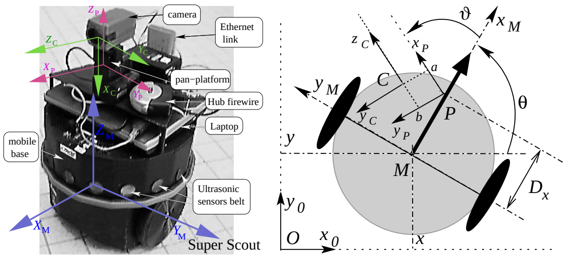
We consider the mobile robot SuperScout II[1] equipped with a camera mounted on a pan-platform (see figure 2(a)). It is a small cylindric cart-like vehicle, dedicated to indoor navigation. A DFW-VL500 Sony color digital IEEE1394 camera captures pictures in YUV 4:2:2 format with 640480 resolution. An image processing module allows to extract points from the image. The robot is controlled by an on-board laptop computer running under Linux on which is installed a specific control architecture called $G^{en}oM$ (Generator of Module) (Fleury and Herrb, 2001). When working on the robot, three different sampling periods are involved:

1. T_E : the integration step defined by $h = t_{k+1} - t_k$,
2. $T_{Ctrl} \simeq 44$ms : the control law sampling period,
3. $T_{Sens} \simeq 100$ms : the camera sampling period.

As Linux is not a real-time OS, these values are only approximately known.

First, let us model our system to express the camera kinematic screw. To this aim, consider figure 2(b). (x, y) are the coordinates of the robot reference point M with respect to the world frame $\mathcal{F}_O$.

[1]The mobile robot SuperScout II is provided by the AIP-PRIMECA.



(a) Nomadic SuperScout II. (b) Modelisation.

Figure 2: The robotic system.

θ and ϑ are respectively the direction of the vehicle and the pan-platform with respect to the x_M-axis. P is the pan-platform center of rotation, D_x the distance between M and P. We consider the successive frames: $\mathcal{F}_M$ (M, x_M, y_M, z_M) linked to the robot, $\mathcal{F}_P$ (P, x_P, y_P, z_P) attached to the pan-platform, and $\mathcal{F}_c$ (C, x_c, y_c, z_c) linked to the camera. The control input is defined by the vector $\dot{q} = (v, \omega, \varpi)^T$, where v and ω are the cart linear and angular velocities, and ϖ is the pan-platform angular velocity with respect to $\mathcal{F}_M$. For this specific mechanical system, the kinematic screw is related to the joint velocity vector by the robot jacobian $\mathbf{J}$: $\mathcal{T}_c = \mathbf{J}\dot{q}$. As the camera is constrained to move horizontally, it is sufficient to consider a reduced kinematic screw $\mathcal{T}_r = (V_{y_c}, V_{z_c}, \Omega_{x_c})^T$, and a reduced jacobian matrix $\mathbf{J}_r$ as follows:

$$\mathcal{T}_r = \begin{pmatrix} -\sin(\vartheta) & D_x\cos(\vartheta) + a & a \\ \cos(\vartheta) & D_x\sin(\vartheta) - b & -b \\ 0 & -1 & -1 \end{pmatrix} \begin{pmatrix} v \\ \omega \\ \varpi \end{pmatrix} = \mathbf{J}_r\dot{q} \quad (3)$$

As $\det(\mathbf{J}_r) = D_x \neq 0$, so the jacobian $\mathbf{J}_r$ is always invertible. Moreover, as the camera is mounted on a pan-platform, its motion is holonomic (see remark 1).

3.2 Execution of a Vision-Based Task Despite Camera Failure

Our objective is to perform a vision-based task despite a camera failure. We first describe the considered mission and state the estimation problem for this particular task before presenting the obtained results.

The Considered Vision-based Task. Our goal is here to position the embedded camera with respect to a landmark made of n points. To this aim, we have applied the visual servoing technique given in (Espiau et al., 1992) to mobile robots as in (Pissard-Gibollet and Rives, 1995). In this approach which relies on the task function formalism (Samson et al., 1991), the visual servoing task is defined as the regulation to zero of the following error function:

$$e_{VS}(q,t) = C\left(s(q,t) - s^*\right) \quad (4)$$

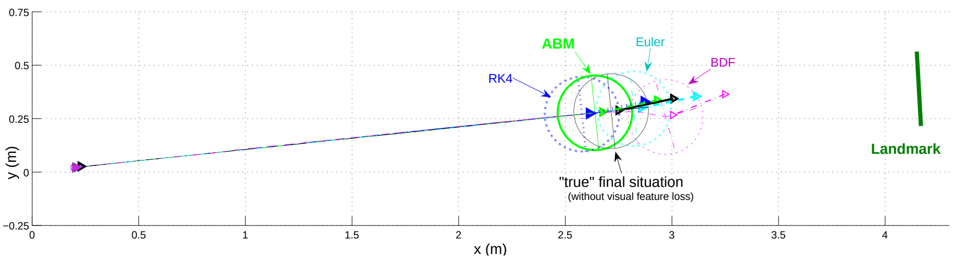


Figure 3: Robot trajectories obtained for the different schemes.

where s is a $2n$-dimensional vector made of the coordinates (U_i, V_i) of each 3D projected point P_i in the image plane (see figure 1). s^* is the desired value of the visual signal, while $\mathcal{C}$ is a full-rank combination matrix which allows to take into account more visual features than available DOF (Espiau et al., 1992). Classically, a kinematic controller, $\dot{q}_{\text{VS}}$ can be determined by imposing an exponential convergence of e_{VS} to zero: $\dot{e}_{\text{VS}} = \mathcal{C}\mathcal{L}\mathbf{J}_{\text{r}}\dot{q}_{\text{VS}} = -\lambda_{\text{VS}}e_{\text{VS}}$, where λ_{VS} is a positive scalar or a positive definite matrix. Fixing $\mathcal{C} = \mathcal{L}^+$ as in (Espiau et al., 1992), we get:

$$\dot{q}_{\text{VS}} = \mathbf{J}_{\text{r}}^{-1}(-\lambda_{\text{VS}})\mathcal{L}^+(s(q,t) - s^*) \quad (5)$$

where $\mathcal{L}$ is a $2n3$ matrix deduced from the classical optic flow equations (Espiau et al., 1992) as follows:

$$\mathcal{L}_i(P_i, z_i) = \begin{bmatrix} 0 & \frac{U}{z_i} & \frac{U_iV_i}{f} \\ -\frac{f}{z_i} & \frac{V_i}{z_i} & f + \frac{V_i^2}{f} \end{bmatrix} \text{ with } i = 1..n \quad (6)$$

where f is the camera focal.

Estimation Problem Statement. Let us state our estimation problem by defining the expression of the ODE (2) in the case of the considered task. As we consider a target made of n points, we need first to determine the depth variation of each of these points. It can be easily shown that, for one 3D point p_i of coordinates $(x_i, y_i, z_i)^T$ in $\mathcal{F}_c$ projected into a point $P_i(U_i, V_i)$ in the image plane as shown in figure 1, the depth variation $\dot{z}_i$ is related to the camera motion according to: $\dot{z}_i = \mathcal{L}_{z_i}\mathcal{T}_{\text{r}}$, with $\mathcal{L}_{z_i} = [0 \quad -1 \quad \frac{z_i}{f}V_i]$. Thus, for the considered task, the ODE to be solved for one point P_i are given by:

$$\begin{cases} \dot{U}_i = \frac{U_i}{z_i}V_{Z_c} + \frac{U_iV_i}{f}\Omega_{X_c} \\ \dot{V}_i = -\frac{f}{z_i}V_{Y_c} + \frac{V_i}{z_i}V_{Z_c} + \left(f + \frac{V_i^2}{f}\right)\Omega_{X_c} \\ \dot{z}_i = -V_{Z_c} - \frac{z_i}{f}V_i\Omega_{X_c} \end{cases} \quad (7)$$

Finally, for the considered landmark made of n points, the ODE (2) are deduced from the above relation (7) by defining: $\mathbf{X} = [U_1 \; V_1 \; \ldots \; U_n \; V_n, \; z_1 \; \ldots \; z_n]^T$.

Experimental Results. We have experimented the considered vision-based task on our mobile robot SuperScout II. For each numerical scheme, we have performed the same navigation task: start from the same configuration using the same $s^\star$. At the beginning of the task, $\dot{q}_{\text{VS}}$ uses the visual features available from the camera and the robot starts converging towards the target. At the same time, the numerical algorithms are initialized and launched. After 10 steps, the landmark is artificially occluded to simulate a camera failure and, if nothing is done, it is impossible to perform the task. Controller (5) is then evaluated using the computed values provided by each of our numerical algorithms and the robot is controlled by an open-loop scheme. For each considered numerical scheme figure 3 shows the robot trajectories and table 1 summarizes the whole results. These errors remain small, which means that there are few perturbations on the system and, in this case, our “emergency” open-loop control scheme allows to reach a neighborhood of the desired goal despite the camera failure. Moreover, *for the proposed task*, the ABM scheme is the most efficient method, as it leads to the least standard deviation error (std) and to the smallest maximal error. The RK4 algorithm gives also correct performances, while Euler method remains the less accurate scheme as expected. As T_{E} is rather small, the BDF technique provides correct results but has been proven to be much more efficient when there are sudden variations in the kinematic screw as it will be shown in the next part.

Table 1: Results synthesis.

Schemes	s / z	std error	max error
Euler	s (pix)	1.0021	9.6842
	z (m)	0.088256	0.72326
RK4	s (pix)	0.90919	7.0202
	z (m)	0.07207	0.63849
ABM	s (pix)	**0.90034**	**5.9256**
	z (m)	**0.05721**	**0.50644**
BDF	s (pix)	1.1172	7.6969
	z (m)	0.10157	0.5989

3.3 Realizing a Navigation Task Amidst Possibly Occluding Obstacles

Our goal is to realize a positioning vision-based task amidst possibly occluding obstacles. Thus, two problems must be addressed: the visual data loss and the risk of collision. The first one will be treated using the above estimation technique and the second one thanks to a rotative potential field method. We describe the control strategy before presenting the results.

Collision and Occlusion Detection. Our control strategy relies on the detection of the risks of collision and occlusion. The danger of collision is evaluated from the distance d_{coll} and the relative orientation α between the robot and the obstacle deduced from the US sensors mounted on the robot. We define three envelopes around each obstacle ξ_+, ξ_0, ξ_-, located at $d_+ > d_0 > d_-$ (see figure 4). We propose to model the *risk of collision* by parameter μ_{coll} which smoothly increases from 0 when the robot is far from the obstacle ($d_{\text{coll}} > d_0$) to 1 when it is close to it ($d_{\text{coll}} < d_-$).

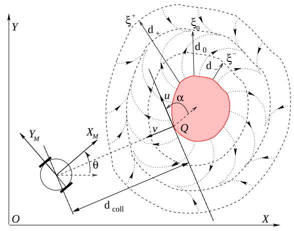
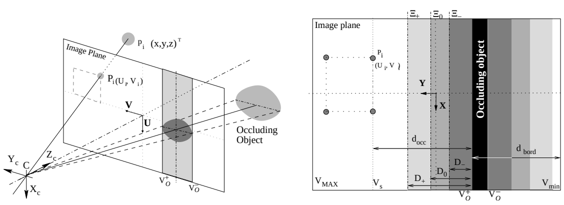


Figure 4: Obstacle avoidance.

The occlusion risk is evaluated from the detection of the occluding object left and right borders extracted by our image processing algorithm (see figure 5(a)). From them, we can deduce the shortest distance d_{occ} between the image features and the occluding object $\mathcal{O}$, and the distance d_{bord} between $\mathcal{O}$ and the opposite image side to the visual features. Defining three envelopes Ξ_+, Ξ_0, Ξ_- around the occluding object located at $D_+ > D_0 > D_-$ from it, we propose to model the risk of occlusion by parameter μ_{occ} which smoothly increases from 0 when $\mathcal{O}$ is far from the visual features ($d_{\text{occ}} > D_0$) to 1 when it is close to them ($d_{\text{occ}} < D_-$). A possible choice for μ_{coll} and μ_{occ} can be found in (Folio and Cadenat, 2005a).



(a) Occluding object projection in the image plane. (b) Definition of the relevant distances in the image.

Figure 5: Occlusion detection.

Global Control Law Design. Our global control strategy relies on μ_{coll} and μ_{occ}. It consists in two steps. First we define two controllers allowing respectively to realize the sole vision-based task and to guarantee non collision while dealing with occlusions in the obstacle vicinity. Second, we switch between these two controllers depending on the risk of occlusion and collision. We propose the following global controller:

$$\dot{q} = (1 - \mu_{\text{coll}})\dot{q}_{VS} + \mu_{\text{coll}}\dot{q}_{\text{coll}} \tag{8}$$

where $\dot{q}_{VS}$ is the visual servoing controller previously defined (5), while $\dot{q}_{\text{coll}} = (v_{\text{coll}}\ \omega_{\text{coll}}\ \varpi_{\text{coll}})^T$ handles obstacle avoidance and visual signal estimation if necessary. Thus, when there is no risk of collision, the robot is driven using only $\dot{q}_{VS}$ and executes the vision-based task. When the vehicle enters the obstacle neighborhood, μ_{coll} increases to reach 1 and the robot moves using only $\dot{q}_{\text{coll}}$. This controller is designed so that the vehicle avoids the obstacle while tracking the target, treating the occlusions if any. It is then possible to switch back to the vision-based task once the obstacle is overcome. The avoidance phase ends when both visual servoing and collision avoidance controllers point out the same direction: $sign(\dot{q}_{VS}) = sign(\dot{q}_{\text{coll}})$, and if the target is not occluded ($\mu_{\text{occ}} = 0$). In this way, we benefit from the avoidance motion to make the occluding object leave the image.

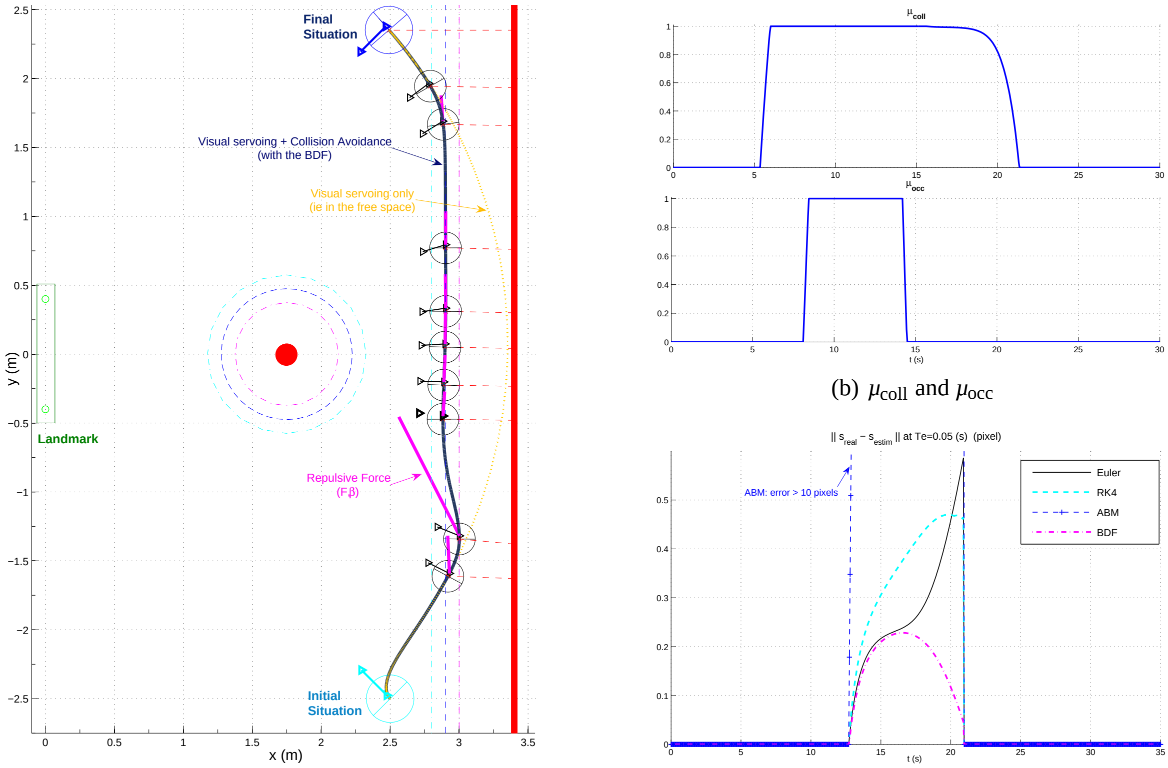
Remark 2 *Controller* (8) *allows to treat occlusions which occur during the avoidance phase. However, obstacles may also occlude the camera field of view without inducing a collision risk. In such cases, we may apply to the robot either another controller allowing to avoid occlusions as done in (Folio and Cadenat, 2005a; Folio and Cadenat, 2005b) for instance, or the open-loop scheme based on the computed visual features given in section 3.2.*

Now, it remains to design $\dot{q}_{\text{coll}}$. We propose to use a similar approach to the one used in (Cadenat et al., 1999). The idea is to define around each obstacle a rotative potential field so that the repulsive force is orthogonal to the obstacle when the robot is close to it ($d_{\text{coll}} < d_+$), parallel to the obstacle when the vehicle is at a distance d_0 from it, and progressively directed towards the obstacle between d_0 and d^+ (as shown on figure 4). The interest of such a potential is that it can make the robot move around the obstacle without requiring any attractive force, reducing local minima problems. We use the same potential function as in (Cadenat et al., 1999):

$$\begin{cases} U(d_{\text{coll}}) = \frac{1}{2}k_1\left(\frac{1}{d_{\text{coll}}} - \frac{1}{d^+}\right)^2 + \frac{1}{2}k_2(d_{\text{coll}} - d^+)^2 & \text{if } d_{\text{coll}} \leq d^+ \\ U(d_{\text{coll}}) = 0 & \text{otherwise} \end{cases} \tag{9}$$

where k_1 and k_2 are positive gains to be chosen. v_{coll} and ω_{coll} are then given by (Cadenat et al., 1999):

$$\dot{q}_b = \begin{pmatrix} v_{\text{coll}} & \omega_{\text{coll}} \end{pmatrix}^T = \begin{pmatrix} k_v F\cos\beta & \frac{k_\omega}{D_x} F\sin\beta \end{pmatrix}^T \tag{10}$$



(a) Robot trajectory (using BDF)

(b) μ_{coll} and μ_{occ}

(c) Error $\|s-\tilde{s}\|$ (pixel) for the different schemes

Figure 6: Simulation results.

where $F = -\frac{\partial U}{\partial d_{\text{coll}}}$ is the modulus of the virtual repulsive force and $\beta = \alpha - \frac{\pi}{2d_0}d_{\text{coll}} + \frac{\pi}{2}$ its direction with respect to $\mathcal{F}_M$. k_v and k_ω are positive gains to be chosen. Equation (10) drives only the mobile base in the obstacle neighborhood. However, if the pan-platform remains uncontrolled, it will be impossible to switch back to the execution of the vision-based task at the end of the avoidance phase. Therefore, we have to address the ϖ_{coll} design problem. Two cases may occur in the obstacle vicinity: either the visual data are available or not. In the first case, the proposed approach is similar to (Cadenat et al., 1999) and the pan-platform is controlled to compensate the avoidance motion while centering the target in the image. As the camera is constrained to move within an horizontal plane, it is sufficient to regulate to zero the error $e_{\text{gc}} = V_{\text{gc}} - V_{\text{gc}}^*$ where V_{gc} and $V_{\text{gc}}^\star$ are the current and desired ordinates of the target gravity center. Rewriting equation (3) as $\mathcal{T}_{\text{r}} = J_{\text{b}}\dot{q}_{\text{b}} + J_\varpi \varpi_{\text{coll}}$ and imposing an exponential decrease to regulate e_{gc} to zero ($\dot{e}_{\text{gc}} = \mathcal{L}_{V_{\text{gc}}}\mathcal{T}_{\text{r}} = -\lambda_{\text{gc}}e_{\text{gc}}$, $\lambda_{\text{gc}} > 0$), we finally obtain (see (Cadenat et al., 1999) for more details):

$$\varpi_{\text{coll}} = \frac{-1}{\mathcal{L}_{V_{\text{gc}}}J_\varpi}(\lambda_{\text{gc}}e_{\text{gc}} + \mathcal{L}_{V_{\text{gc}}}J_{\text{b}}\dot{q}_{\text{b}}) \qquad (11)$$

where $\mathcal{L}_{V_{\text{gc}}}$ is the 2^{nd} row of $\mathcal{L}_i$ evaluated for V_{gc} (see equation (6)). However, if the obstacle occludes the camera field of view, s is no more available and the pan-platform cannot be controlled anymore using (11). At this time, we compute the visual features by integrating the ODE (2) using one of proposed numerical schemes. It is then possible to keep on executing the previous task e_{gc}, even if the visual features are temporary occluded by the encountered obstacle. The pan-platform controller during an occlusion phase will then be deduced by replacing the real target gravity center ordinate V_{gc} by the computed one $\widetilde{V}_{\text{gc}}$ in (11). We get:

$$\widetilde{\varpi}_{\text{coll}} = \frac{-1}{\widetilde{\mathcal{L}}_{V_{\text{gc}}}J_\varpi}(\lambda_{\text{gc}}\tilde{e}_{\text{gc}} + \widetilde{\mathcal{L}}_{V_{\text{gc}}}J_{\text{b}}\dot{q}_{\text{b}}), \qquad (12)$$

where $\tilde{e}_{\text{gc}} = \widetilde{V}_{\text{gc}} - V_{\text{gc}}^*$ and $\widetilde{\mathcal{L}}_{V_{\text{gc}}}$ is deduced from (6). Now, it remains to apply the suitable controller to the pan-platform depending on the context. Recalling that the parameter $\mu_{\text{occ}} \in [0;1]$ allows to detect occlusions, we propose the following avoidance controller:

$$\dot{q}_{\text{coll}} = \begin{pmatrix} v_{\text{coll}}, & \omega_{\text{coll}}, & (1-\mu_{\text{occ}})\varpi_{\text{coll}} + \mu_{\text{occ}}\widetilde{\varpi}_{\text{coll}} \end{pmatrix}^T \qquad (13)$$

Simulation Results. The proposed method has been simulated using Matlab software. We aim at positioning the camera with respect to a given landmark despite two obstacles. D_-, D_0 and D_+ have been fixed to

40, 60 and 115 pixels, and d_-, d_0, d_+ to 0.3m, 0.4m, and 0.5m. For each numerical scheme (Euler, RK4, ABM and BDF), we have performed the same navigation task, starting from the same situation and using the same $s^\star$. Figure 6(c) shows that the BDF are the most efficient scheme, while ABM is the worst, RK4 and Euler giving correct results for this task. Indeed, as the obstacle avoidance induces important variations in the camera motion, ODE (2) becomes stiff, and the BDF have been proven to be more suitable in such cases. Figures 6(a) and 6(b) show the simulation results obtained using this last scheme. The task is perfectly performed despite the wall and the circular obstacle. The different phases of the motion can be seen on the evolution of $\mu_{\rm coll}$ and $\mu_{\rm occ}$. At the beginning of the task, there is no risk of collision, nor occlusion, and the robot is driven by $\dot{q}_{\rm VS}$. When it enters the wall neighborhood, $\mu_{\rm coll}$ increases and $\dot{q}_{\rm coll}$ is applied to the robot which follows the security envelope ξ_0 while centering the landmark. When the circular obstacle enters the camera field of view, $\mu_{\rm occ}$ increases and the pan-platform control smoothly switches from $\varpi_{\rm coll}$ to $\widetilde{\varpi}_{\rm coll}$. It is then possible to move along the security envelope ξ_0 while tracking a "virtual" target until the end of the occlusion. When there is no more danger, the control switches back to $\dot{q}_{\rm VS}$ and the robot perfectly realizes the desired task.

4 CONCLUSIONS

In this paper, we have proposed to apply classical numerical integration algorithms to determine visual features whenever unavailable during a vision-based task. The obtained algorithms have been validated both in simulation and experimentation with interesting results. A comparative analysis has been performed and has shown that the BDF is particularly efficient when ODE (2) becomes stiff while giving correct results in more common use. Therefore, it appears to be the most interesting scheme.

REFERENCES

Benhimane, S. and Malis, E. (2003). Vision-based control with respect to planar and non-planar objects using a zooming camera. In *Int. Conf. on Advanced Robotics*, Coimbra, Portugal.

Cadenat, V., Souères, P., Swain, R., and Devy, M. (1999). A controller to perform a visually guided tracking task in a cluttered environment. In *Int. Conf. on Intelligent Robots and Systems*, Korea.

Chaumette, F. (2004). Image moments: a general and useful set of features for visual servoing. *Trans. on Robotics and Automation*, 20(4):713–723.

Corke, P. (1996). *Visual control of robots : High performance visual servoing*. Research Studies Press LTD.

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

Espiau, B., Chaumette, F., and Rives, P. (1992). A new approach to visual servoing in robotics. *Trans. on Robotics and Automation*.

Fleury, S. and Herrb, M. (2001). *$G^{en}oM$: User Manual*. LAAS-CNRS.

Folio, D. and Cadenat, V. (2005a). A controller to avoid both occlusions and obstacles during a vision-based navigation task in a cluttered environment. In *European Control Conference(ECC-CDC'05)*.

Folio, D. and Cadenat, V. (2005b). Using redundancy to avoid simultaneously occlusions and collisions while performing a vision-based task amidst obstacles. In *European Conference on Mobile Robots*, Ancona, Italy.

Garcia-Aracil, N., Malis, E., Aracil-Santonja, R., and Perez-Vidal, C. (2005). Continuous visual servoing despite the changes of visibility in image features. *Trans. on Robotics and Automation*, 21.

Hutchinson, S., Hager, G., and Corke, P. (1996). A tutorial on visual servo control. *Trans. on Robotics and Automation*.

Kyrki, V., Kragic, D., and Christensen, H. (2004). New shortest-path approaches to visual servoing. In *Int. Conf. on Intelligent Robots and Systems*.

Malis, E., Chaumette, F., and Boudet, S. (1999). 2 1/2d visual servoing. *Trans. on Robotics and Automation*, 15:238–250.

Mansard, N. and Chaumette, F. (2005). A new redundancy formalism for avoidance in visual servoing. In *Int. Conf. on Intelligent Robots and Systems*, volume 2, pages 1694–1700, Edmonton, Canada.

Marchand, E. and Hager, G. (1998). Dynamic sensor planning in visual servoing. In *Int. Conf. on Robotics and Automation*, Leuven, Belgium.

Mezouar, Y. and Chaumette, F. (2002). Avoiding self-occlusions and preserving visibility by path planning in the image. *Robotics and Autonomous Systems*.

Pissard-Gibollet, R. and Rives, P. (1995). Applying visual servoing techniques to control a mobile hand-eye system. In *Int. Conf. on Robotics and Automation*, Nagoya, Japan.

Remazeilles, A., Mansard, N., and Chaumette, F. (2006). Qualitative visual servoing: application to the visibility constraint. In *Int. Conf. on Intelligent Robots and Systems*, Beijing, China.

Samson, C., Leborgne, M., and Espiau, B. (1991). *Robot Control. The Task Function Approach*, volume 22 of *Oxford Engineering Series*. Oxford University Press.

Shampine, L. F. and Gordon, M. K. (1975). *Computer Solution of Ordinary Differential Equations*. W. H. Freeman, San Francisco.

ROBUST AND ACTIVE TRAJECTORY TRACKING FOR AN AUTONOMOUS HELICOPTER UNDER WIND GUST

Adnan Martini, François Léonard and Gabriel Abba
Industrial Engineering and Mechanical Production Lab (LGIPM), ENIM, Ile du Saulcy, 57045 Metz cedex 1, France
a.martini@enim.fr, leonard@enim.fr, abba@enim.fr

Keywords: Helicopter, Nonlinear systems, Robust nonlinear control, Active disturbance rejection control, Observer.

Abstract: The helicopter manoeuvres naturally in an environment where the execution of the task can easily be affected by atmospheric turbulences, which lead to variations of its model parameters.The originality of this work relies on the nature of the disturbances acting on the helicopter and the way to compensate them. Here, a nonlinear simple model with 3-DOF of a helicopter with unknown disturbances is used. Two approaches of robust control are compared via simulations: a nonlinear feedback and an active disturbance rejection control based on a nonlinear extended state observer(*ADRC*).

1 INTRODUCTION

The control of nonlinear systems under disturbance is an active sector of research in the last decades especially in aeronautics where several elegant approaches were presented. We consider here the problem of control of a Lagrangian model with 3- DOF of a helicopter assembled on a platform (VARIO 23cc). It is subjected to a wind gust and it carries out a vertical flight (takeoff, slope, flight, descent and landing). The mathematical model of the system is very simple but its dynamic is not trivial (nonlinear in state, and underactuated).

Basically, the methods of control which adress the attenuation of the disturbance, can be classified according to the different kinds of disturbances. A possible approach is to model the disturbances by a stochastic process, which leads to the theory of nonlinear stochastic control (Gokçek et al., 2000). Another approach is the nonlinear control (Marten et al., 2005) where it is supposed that the energy of the disturbances is limited. A third approach is to treat the disturbances produced by a neutral stable exogenous system using the nonlinear theory of output regulation (Isidori, 1995) (Byrnes et al., 1997) and (Marconi and Isidori, 2000). (Wei, 2001) showed the control of the nonlinear systems with unknown disturbances, where an approach based on the disturbance observer based control (*DOBC*) is carried out: a nonlinear observer of disturbance is presented to estimate the unknown disturbances. This is integrated with a conventional controller by using techniques based on the observation of the disturbance. (Hou et al., 2001) proposed a method of active disturbance rejection control (*ADRC*) which estimates the disturbance with an extended state observer. Many industrial applications use this method (Gao et al., 2001) (Zeller et al., 2001) (Jiang and Gao, 2001) and (Hamdan and Gao, 2000).

In this paper, an observer methodology is proposed to control a disturbed drone helicopter. It is based on the concept of active disturbance rejection control (*ADRC*). In this approach the disturbances are estimated by using an extended state observer (*ESO*) and are compensated for each sampling period.

In section 2, a model of a disturbed helicopter is presented. Details of the section of *ADRC* control are given in section 3. Section 4 presents an application of this method on our problem. Section 5 is dedicated to the zero-dynamics analysis. In section 6, several simulations of the helicopter under wind gust show the relevance of the two controls which are described in this work.

2 MODEL OF THE DISTURBED HELICOPTER

This section presents the nonlinear model of the disturbed helicopter (Martini et al., 2005) starting from a non disturbed model (Vilchis et al., 2003). The Lagrange equation, which describes the system of the helicopter-platform with the disturbance (see figure1), is given by:

$$M(q)\ddot{q} + C(q,\dot{q}) + G(q) = Q(q,\dot{q},u,v_{raf}) \quad (1)$$

The input vector of the control and the state vector are respectively $u = [u_1 \quad u_2]^T$, $x = [z \quad \dot{z} \quad \phi \quad \dot{\phi} \quad \gamma \quad \dot{\gamma}]^T$. The induced gust velocity is noted v_{raf}. Moreover, $q = [z \quad \phi \quad \gamma]^T$,where z represents the helicopter altitude, ϕ is the yaw angle and γ represents the main rotor azimuth angle, $M \in R^{3\times3}$ is the inertia matrix, $C \in R^{3\times3}$ is the Coriolis and centrifugal forces matrix, $G \in R^3$ represents the vector of conservative forces, $Q(q,\dot{q},u,v_{raf}) = [f_z \quad \tau_z \quad \tau_\gamma]^T$ is the vector of generalized forces. The variables f_z, τ_z and τ_γ represent respectively, the vertical force, the yaw torque and the main rotor torque. Finally, the representation of the reduced system of the helicopter, which is subjected to a wind gust, can be written in the following state form (Martini et al., 2005):

$$\begin{aligned}
\dot{x_1} =& x_2 = \dot{z} \\
\dot{x_2} =& \frac{1}{c_0}[c_8\dot{\gamma}^2 u_1 + c_9\dot{\gamma} + c_{10} - c_7] + \frac{1}{c_0}c_{16}\dot{\gamma}v_{raf} \\
\dot{x_3} =& x_4 = \dot{\phi} \\
\dot{x_4} =& \frac{1}{c_1c_5 - c_4^2}[c_5c_{11}\dot{\gamma}^2u_2 - c_4((c_{12}\dot{\gamma} + c_{13} + \\
& c_8\dot{\gamma}v_{raf})u_1 + c_{14}\dot{\gamma}^2 + c_{15})] \\
& - \frac{c_4}{c_1c_5 - c_4^2}[2c_9v_{raf} + c_{17}v_{raf}^2] = \ddot{\phi} \\
\dot{x_5} =& x_6 = \dot{\gamma} \\
\dot{x_6} =& \frac{1}{c_1c_5 - c_4^2}[c_{11}c_4\dot{\gamma}^2u_2 + c_1c_4((c_{12}\dot{\gamma} + c_{13} \\
& c_8\dot{\gamma}v_{raf})u_1 + c_{14}\dot{\gamma}^2 + c_{15})] \\
& + \frac{1}{c_1c_5 - c_4^2}[2c_9v_{raf} + c_{17}v_{raf}^2] = \ddot{\gamma}
\end{aligned} \quad (2)$$

where c_i(i =0,...,17) are the physical constants of the model.

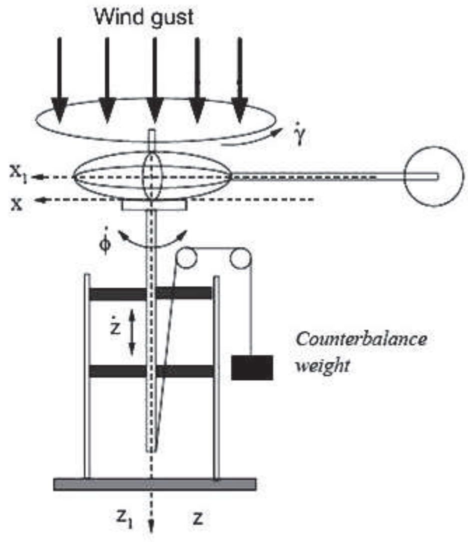


Figure 1: Helicopter-platform (Vilchis et al., 2003).

3 NONLINEAR EXTENDED STATE OBSERVER (*NESO*)

The primary reason to use the control in closed loop is that it can treat the variations and uncertainties of model dynamics and the outside unknown forces which exert influences on the behavior of the model. In this work, a methodology of generic design is proposed to treat the combination of two quantities, denoted as disturbance.

A second order system described by the following equation is considered (Gao et al., 2001)(Hou et al., 2001):

$$\ddot{y} = f(y,\dot{y},w) + bu \quad (3)$$

where $f(.)$ represents the dynamics of the model and the disturbance, w is the input of unknown disturbance, *u* is the input of control, and *y* is the measured output. It is assumed that the value of the parameter *b* is given. Here $f(.)$ is a nonlinear function.

An alternative method is presented by (Han, 1999)(Han, 1995) as follows. The system in (3) is initially increased:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_3 + bu \\ \dot{x}_3 = \dot{f} \end{cases} \quad (4)$$

where $x_1 = y$, $x_2 = \dot{y}$, $x_3 = f(y,\dot{y},w)$. $f(.)$ is treated as an increased state. Here f and $\dot{f}$ are unknown. By considering $f(y,\dot{y},w)$ as a state, it can be estimated with a state estimator. Han in (Han, 1999) proposed a nonlinear observer for (4):

$$\begin{aligned} \dot{\hat{x}} =& \quad A\hat{x} + Bu + Lg(e,\alpha,\delta) \\ \hat{y} =& \quad C\hat{x} \end{aligned} \quad (5)$$

where:

$$A=\begin{bmatrix}0&1&0\\0&0&1\\0&0&0\end{bmatrix},B=\begin{bmatrix}0\\b\\0\end{bmatrix},C=\begin{bmatrix}1&0&0\end{bmatrix} \tag{6}$$

and $L = \begin{bmatrix} L_1 & L_2 & L_3 \end{bmatrix}$. The observer error is $e = y - \hat{x}_1$ and:

$$g_i(e,\alpha_i,\delta)_{|i=1,2,3}=\begin{cases}|e|^{\alpha_i}sign(e) & |e|>\delta \\ \frac{e}{\delta^{1-\alpha_i}} & |e|\le\delta\end{cases} \quad \delta>0$$

The observer is reduced to the following set of state equations, and is called extended state observer (*ESO*):

$$\begin{cases}\dot{\hat{x}}_1=\hat{x}_2+L_1g_1(e,\alpha,\delta)\\ \dot{\hat{x}}_2=\hat{x}_3+L_2g_2(e,\alpha,\delta)+bu\\ \dot{\hat{x}}_3=L_3g_3(e,\alpha,\delta)\end{cases} \tag{7}$$

The active disturbance rejection control (*ADRC*) is then defined as a method of control where the value of $f(y, \dot{y}, w)$ is estimated in real time and is compensated by the control signal *u*. Since $\hat{x}_3 \rightarrow f$, it is used to cancel actively *f* by the application of:

$$u=(u_0-\hat{x}_3)/b \tag{8}$$

This expression reduces the system to:

$$\ddot{y}=(f-\hat{x}_3)+u_0\approx u_0 \tag{9}$$

The process is now a double integrator with a unity gain, which can be controlled with a PD controller:

$$u_0=k_p(r-\hat{x}_1)-k_d\hat{x}_2 \tag{10}$$

where *r* is the reference input. The observer gains L_i and the controller gains k_p and k_d can be calculated by a pole placement. The configuration of *ADRC* is presented in figure2 :

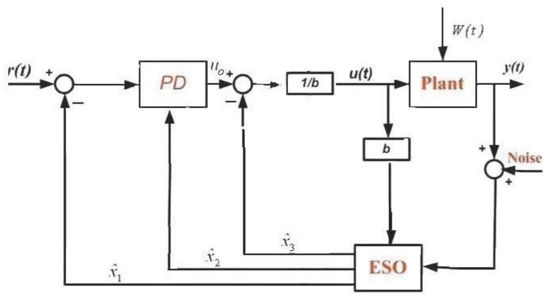


Figure 2: *ADRC* structure.

4 CONTROL OF DISTURBED HELICOPTER

4.1 Control by Nonlinear Feedback

Firstly, the nonlinear terms of the non disturbed model ($v_{raf} = 0$) are compensated by introducing two new controls V_1 and V_2 such as(see Fig.3):

$$\begin{aligned}u_1 &= \tfrac{1}{c_8\dot{\gamma}^2}[c_0V_1-c_9\dot{\gamma}-c_10+c_7]\\ u_2 &= \tfrac{1}{c_5c_{11}\dot{\gamma}^2}[(c_1c_5-c_4^2)V_2+c_4((c_{12}\dot{\gamma}+\\ &\quad c_{13})u_1+c_{14}\dot{\gamma}^2+c_{15})].\end{aligned} \tag{11}$$

By using the above controls V_1 and V_2 , for $v_{raf} = 0$, an uncoupled linear system is obtained which is represented by two equations: $\ddot{z} = V_1$, $\ddot{\phi} = V_2$. Stabilization is carried out by a pole placement. To regulate altitude *z* and the yaw angle ϕ, a *PID* controller is proposed:

$$\begin{aligned}V_1 &= -a_1\dot{z}-a_2(z-z_d)-a_3\int_0^t(z-z_d)dt\\ V_2 &= -a_4\dot{\phi}-a_5(\phi-\phi_d)-a_6\int_0^t(\phi-\phi_d)dt\end{aligned} \tag{12}$$

where z_d and ϕ_d are the desired trajectories. The parameters of regulation were calculated using two dominant poles in closed loop such as:

$$z\begin{cases}\omega_z=2\text{rad/s}\\ \xi_z=1\end{cases} \quad \text{and} \quad \phi\begin{cases}\omega_\phi=5\text{rad/s}\\ \xi_\phi=1\end{cases} \tag{13}$$

where ω_z, ω_ϕ are the natural frequencies, and $\xi_{z,\phi}$ are the damping ratios for the pole placement. These integral controllers are used to eliminate the effect of low frequency disturbance.

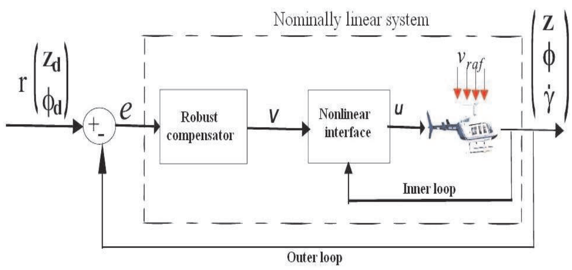


Figure 3: Architecture of nonlinear feedback control.

4.2 Active Disturbance Rejection Control (ADRC)

Since $v_{raf} \neq 0$, a nonlinear system of equations is obtained:

$$\begin{aligned}\ddot{z} &= V_1+\tfrac{1}{c_0}c_{16}\dot{\gamma}v_{raf}\\ \ddot{\phi} &= V_2-\tfrac{c_4c_0v_{raf}}{(c_1c_5-c_4^2)\dot{\gamma}}V_1-\tfrac{c_4v_{raf}}{c_1c_5-c_4^2}\left[\tfrac{c_7-c_{10}}{\dot{\gamma}}+c_9+c_{17}v_{raf}\right]\end{aligned} \tag{14}$$

The stabilization is always done by pole placement. To regulate altitude z and the yaw angle ϕ, we can notice that (14) represent two second order systems which can be written as in (3):

$$\ddot{y} = f(y, \dot{y}, w) + bu \tag{15}$$

with $b = 1$,$u = V_1$ or V_2 and:

$$f_z(y, \dot{y}, w) = \frac{1}{c_0} c_{16} \dot{\gamma} v_{raf}$$
$$f_\phi(y, \dot{y}, w) = -\frac{c_4 c_0 v_{raf}}{(c_1 c_5 - c_4^2)\dot{\gamma}} V_1 - \frac{c_4 v_{raf}}{c_1 c_5 - c_4^2} \left[\frac{c_7 - c_{10}}{\dot{\gamma}} + \right. \\ \left. + c_9 + c_{17} v_{raf} \right]$$

For each control, an observer is built using (7):

- for altitude z :

$$\begin{cases} \dot{\hat{x}}_1 = \hat{x}_2 + L_1 g_1(e_z, \alpha_1, \delta_1) \\ \dot{\hat{x}}_2 = \hat{x}_3 + L_2 g_2(e_z, \alpha_2, \delta_2) + bV_1 \\ \dot{\hat{x}}_3 = L_3 g_3(e_z, \alpha_3, \delta_3) \end{cases} \tag{16}$$

where $e_z = z - \hat{x}_1$ is the observer error, $g_i(e_i, \alpha_i, \delta_i)$ is defined as exponential function of modified gain.

$$g_i(e_z, \alpha_{iz}, \delta_i)_{|i=1,2,3} = \begin{cases} |e_z|^{\alpha_{iz}} sign(e_z), & |e_z| > \delta_i \\ \frac{e_z}{\delta_i^{1-\alpha_{iz}}}, & |e_z| \leq \delta_i \end{cases}$$

with $0 < \alpha_i < 1$ and $0 < \delta_i$, a *PID* controller is used in stead of *PD*(10) to attenuate the effects of disturbance:

$$V_1 = -k_1 \hat{x}_2 - k_2(\hat{x}_1 - z_d) - k_3 \int_0^t (\hat{x}_1 - z_d) dt - \hat{x}_3 \tag{17}$$

The control signal V_1 takes into account the terms which depend on the observer $(\hat{x}_1, \hat{x}_2)$. The fourth part, which also comes from the observer, is added to eliminate the effect of disturbance in this system.

- for the yaw angle ϕ :

$$\begin{cases} \dot{\hat{x}}_4 = \hat{x}_5 + L_4 g_4(e_\phi, \alpha_4, \delta_4) \\ \dot{\hat{x}}_5 = \hat{x}_6 + L_5 g_5(e_\phi, \alpha_5, \delta_5) + bV_2 \\ \dot{\hat{x}}_6 = L_6 g_6(e_\phi, \alpha_6, \delta_6) \end{cases} \tag{18}$$

where $e_\phi = \phi - \hat{x}_4$ is the observer error, with $g_i(e_\phi, \alpha_{i\phi}, \delta_i)$ is defined as exponential function of modified gain:

$$g_i(e_\phi, \alpha_{i\phi}, \delta_i)_{|i=4,5,6} = \begin{cases} |e_\phi|^{\alpha_{i\phi}} sign(e_\phi), & |e_\phi| > \delta_i \\ \frac{e_\phi}{\delta_i^{1-\alpha_{i\phi}}}, & |e_\phi| \leq \delta_i \end{cases}$$

$$V_2 = -k_5 \hat{x}_4 - k_4(\hat{x}_5 - \phi_d) - k_6 \int_0^t (\hat{x}_4 - \phi_d) dt - \hat{x}_6 \tag{19}$$

z_d and ϕ_d are the desired trajectories. *PID* parameters are designed to obtain two dominant poles in closed-loop:

$$\text{for } z \begin{cases} \omega_{c1} = 2rad/s \\ \xi_1 = 1 \end{cases} \quad \text{and for } \phi \begin{cases} \omega_{c2} = 5rad/s \\ \xi_2 = 1 \end{cases} \tag{20}$$

5 ZERO DYNAMICS PROBLEM

The zero dynamics of a nonlinear system are its internal dynamics subject to the constraint that the outputs (and, therefore, all their derivatives) are set to zero for all times (Isidori, 1995)(Slotine and Li, 1991). Nonlinear systems with nonasymptotically stable zero-dynamics are called strictly (or weakly, if the zero dynamics are marginally stable) nonminimum phase system. The output of our system is $q = \begin{bmatrix} z & \phi \end{bmatrix}^T$ and its control input $u = \begin{bmatrix} u_1 & u_2 \end{bmatrix}^T$. The calculation of the relative degrees gives: $r_1 = r_2 = 2$. The dimension of our model $n = 5$ so that: $r_1 + r_2 < n$ what implies the existence of an internal dynamics. If a linearizable feedback is used, it is necessary to check the stability of this internal dynamics. In fact the $\dot{\gamma}$ dynamics represents the zeros-dynamics of (2). Moreover the nonlinear terms of the non disturbed model ($v_{raf} = 0$) can be compensated by introducing two new controls V_1 and V_2. Since $v_{raf} = 0$, a nonlinear system of equations is then obtained:

$$\begin{array}{l} \ddot{z} = V_1; \quad \ddot{\phi} = V_2 \\ \ddot{\gamma} = \frac{1}{c_1 c_5 - c_4^2} [b_1 V_1 + b_2 V_2 + b_3] \end{array} \tag{21}$$

Where:

$$\begin{array}{l} b_1 = \frac{c_4 c_0}{c_5 c_8 \dot{\gamma}^2} (c_{12}\dot{\gamma} + c_{13})(c_1 c_5 + c_4) \\ b_2 = \frac{c_4}{c_5} (c_1 c_5 - c_4^2) \\ b_3 = \dfrac{c_4}{c_5} (c_1 c_5 + c_4)[(c_{12}\dot{\gamma} + c_{13})(-c_9 \dot{\gamma} - \\ \qquad c_{10} + c_7) \times \dfrac{1}{c_8 \dot{\gamma}^2} + c_{14}\dot{\gamma}^2 + c_{15})] \end{array} \tag{22}$$

Zero dynamics of nondisturbed model can then obviously be got by putting $z = \phi = 0 \Rightarrow \dot{z} = \dot{\phi} = 0 \Rightarrow \ddot{z} = \ddot{\phi} = 0 \Rightarrow V_1 = V_2 = 0$:

$$\ddot{\gamma} = \frac{1}{c_1 c_5 - c_4^2} b_3 \tag{23}$$

Simplified in:

$$\ddot{\gamma} = b_4 \dot{\gamma}^2 + \frac{b_5}{\dot{\gamma}^2} + \frac{b_6}{\dot{\gamma}} + b_7 \tag{24}$$

With: $b_4 = 4.1425 \times 10^{-5}$, $b_5 = -778300$, $b_6 = -6142$ and $b_7 = 0.1814$. To get possible equilibrium points dynamics of (24), the following equation is solved:

$$b_4 \dot{\gamma}^4 + b_7 \dot{\gamma}^2 + b_6 \dot{\gamma} + b_5 = 0 \tag{25}$$

The four solutions of (25) are $\dot{\gamma}^* = -219.5 \pm 468.2i$, 563.71 and $-124.6\,rad/s$. Only the two last values of $\dot{\gamma}^*$ have physical meaning for the system. On the other hand, the value $\dot{\gamma}^* = 563.7rad/s$ is too high regarding the blade fragility. As a result, the only equilibrium point to consider is $\dot{\gamma}^* = -124.6rad/s$. The $\dot{\gamma}-$dynamics(24), linearized around the equilibrium point of interest $\dot{\gamma}^* = -124.634\,rad/s$, has a real eigenvalue equal to -0.419. As a consequence, all trajectories starting sufficiently near $\dot{\gamma}^*$ converge to the latter. Then it follows that the zeros-dynamics of (21) has a stable behavior. Simulation results show that $\dot{\gamma}$ remains bounded away from zero during the flight. For the chosen trajectories and gains $\dot{\gamma}$ converges rapidly to a constant value(see Figure6). This is an interesting point to note since it shows that the dynamics and feedback control yield flight conditions close to the ones of real helicopters which fly with a constant $\dot{\gamma}$ thanks to a local regulation feedback of the main rotor speed (which does not exist on the *VARIO* scale model helicopter).

6 RESULTS IN SIMULATION

To show the efficiency of active disturbance rejection control (*ADRC*), it is compared to the nonlinear control, which uses a *PID* controller. The various numerical values are the following:

6.1 Control by Nonlinear Feedback

We have $a_1 = 24$, $a_2 = 84$, $a_3 = 80$, $a_4 = 60$, $a_5 = 525$ and $a_6 = 1250$, the numerical values are calculating by pole placement as defined in (13).

6.2 Active Disturbance Rejection Control (ADRC)

- For z:

  $k_1 = 24$, $k_2 = 84$, $k_3 = 80$ (the numerical values are calculating by pole placement as defined in (20)). Choosing a triple pole located in ω_{0z} such as $\omega_{0z} = (3 \sim 5)\omega_{c_1}$, one can choose $\omega_{0z} = 10rad/s$, $\alpha_1 = 0.5$, $\delta_1 = 0.1$. Using pole placement method, the gains of the observer for the case $|e| \leq \delta$ (i.e linear observer) can be evaluated:

$$\begin{array}{l} \frac{L_1}{\delta_1^{1-\alpha_1}} = 3\,\omega_{0z} \\ \frac{L_2}{\delta_1^{1-\alpha_1}} = 3\,\omega_{0z}^2 \\ \frac{L_3}{\delta_1^{1-\alpha_1}} = \omega_{0z}^3 \end{array} \tag{26}$$

  which leads to: $L_i = \{9.5,\ 95,\ 316\}, i \in [1,2,3]$

- For ϕ:

  $k_4 = 60$, $k_5 = 525$, $k_6 = 1250$ and $\omega_{0\phi} = 25rad/s$, $\alpha_2 = 0.5, \delta_2 = 0.025$. And by the same method in (26) one can find the observer gains: $L_i = \{11.9,\ 296.5,\ 2470\}, i \in [4,5,6]$

For the (*ADRC*) control, one can show that if the gains of the observer $L_{i=1,2,3,4,5,6}$ are too large, the convergence of $\hat{x}_{i=1,2,3,4,5,6}$ to the following values $(z,\dot{z},\ddot{z},\phi,\dot{\phi},\ddot{\phi})$ is very fast but the robustness against the noises quickly deteriorated. By choosing $L_{i+1} \gg L_i(i = 1,2,\cdots,6)$, higher order observer state $\hat{x}_{i+1}$ will converge to the actual value more quickly than lower order state $\hat{x}_i$. Therefore, the stability of *ADRC* system will be guaranteed. δ is the width of linear area in the nonlinear function *ADRC*. It plays an important role to the dynamic performance of *ADRC*. The larger δ is, the wider the linear area. But if δ is too large, the benefit of nonlinear characteristics would be lost. On the other hand, if δ is too small, then high frequency chattering will happen just the same as in the sliding mode control. Generally, in *ADRC*, δ is set to be approximately 10% of the variation range of its input signal. α is the exponent of tracking error. The smaller α is, the faster the tracking speed is, but the more calculation time is needed. In addition, very small α will cause chattering. In reality, selecting $\alpha = 0.5$ will provide a satisfactory result. The induced gust velocity operating on the main rotor is chosen as (G.D.Padfield, 1996):

$$v_{raf} = v_{gm}\sin\left(\frac{2\pi Vt}{L_u}\right) \tag{27}$$

where V in *m/s* is the rise speed of the helicopter and $v_{gm} = 0.68m/s$ is the gust density. This density corresponds to an average wind gust, and $L_u = 1.5m$ is its length (see Figure8). In simulation the gust is applied at t=80s.

A band limited white noise of covariance $2 \times 10^{-8}\mathrm{m}^2$ for *z* and $2 \times 10^{-9}\mathrm{rad}^2$ for ϕ, has been added equally to the measurements of *z* and ϕ for the two controls. The compensation of this noise is done by using a Butterworth second-order low-pass filter. Its crossover frequency for *z* is $\omega_{cz} = 10$rad/s and for ϕ is $\omega_{c\phi} = 25$rad/s.

The parameters used for 3DOF standard helicopter model are based on a VARIO 23cc small helicopter(see figure 1).

Figure4 shows the desired trajectories. Figure7 illustrates the variations of control inputs, where from initial conditions when $\|\dot{\gamma}\|$ increases quickly, the control output u_1 and u_2 saturates. Nevertheless the stability of the closed-loop system is not destroyed.

One can observe that $\dot{\gamma} \rightarrow -124.6rad/s$ as expected from the previous zero dynamics analysis. One can also notice that the main rotor angular speed is similar for the two controls as illustrated in Figure6.

The difference between the two controls appears in Figure5 where the tracking errors are less significant by using the *PID* (*ADRC*) control than *PID* controller. One can see in Figure8 that the main rotor thrust converges to values that compensate the helicopter weight, the drag force and the effect of the disturbance on the helicopter.

The simulations show that the control by nonlinear feedback *PID* (*ADRC*) is more effective than nonlinear *PID* controller, i.e. the tracking errors are less significant by using the first control. But the *PID* (*ADRC*) control is a little more sensitive to noise than *PID* controller. Moreover, under the effect of noise, the second control allows the main rotor thrust T_m to be less away from its balance position than the first control (Figure8). Figure9 represent the effectiveness of the observer: $\hat{x}_3$ and $f_z(y, \dot{y}, w)$, are very close and also $\hat{x}_6$ and $f_\phi(y, \dot{y}, w)$. Observer errors are presented in the Figure10. By tacking a large disturbance ($v_{raf} = 3m/s$), the *ADRC* control shows a robust behavior compared to the nonlinear *PID* control as illustrated in Figure11.

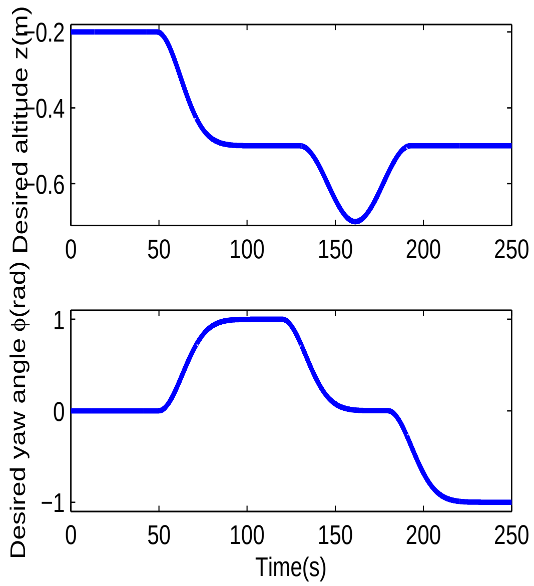


Figure 4: The desired trajectories in z and ϕ.

7 CONCLUSION

In this paper, the active disturbance rejection control (*ADRC*) has been applied to the drone helicopter control. The basis of *ADRC* is the extended state observer. The state estimation and compensation of the change of helicopter parameters and disturbance variations are implemented by *ESO* and *NESO*. By using

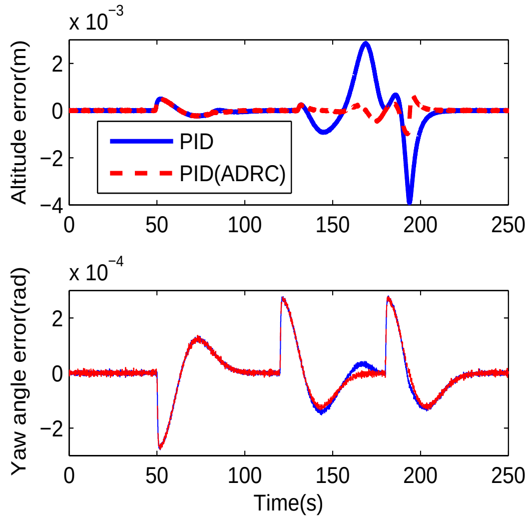


Figure 5: Tracking error in z and ϕ.

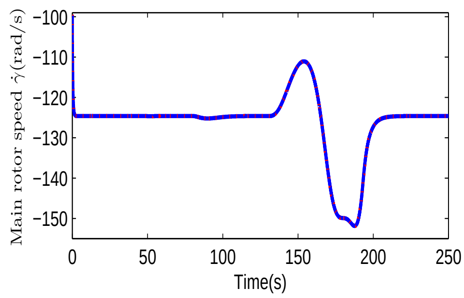


Figure 6: Variations of the main rotor angular speed $\dot{\gamma}$.

ESO, the complete decoupling of the helicopter is obtained. The major advantage of the proposed method is that the closed loop characteristics of the helicopter system do not depend on the exact mathematical model of the system. Comparisons were made in detail between *ADRC* and conventional nonlinear *PID* controller. It is concluded that the proposed control algorithm produces better dynamic performance than the nonlinear *PID* controller. Even for large disturbance $v_{raf} = 3m/s$(figure11), the proposed *ADRC* control system is robust against the modeling uncertainty and the external disturbance in various operating conditions. It is indicated that such scheme can be applicable to aeronautical applications where high dynamic performance is required. We note that the next step will be the validation of this study on the real helicopter model VARIO 23cc.

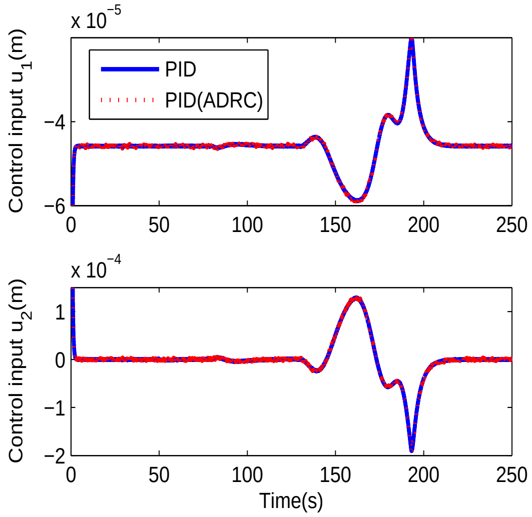


Figure 7: The control inputs u_1 and u_2.

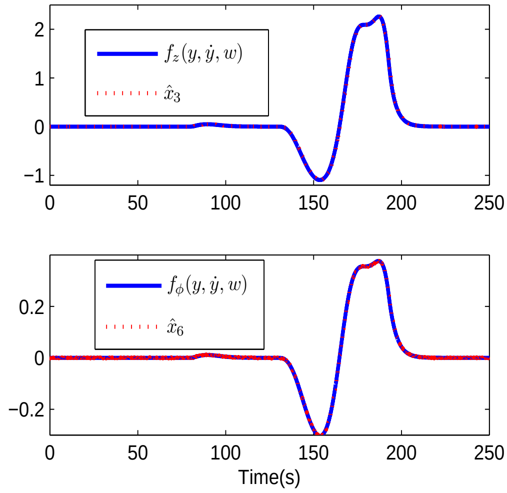


Figure 9: Estimation of f_z and f_ϕ.

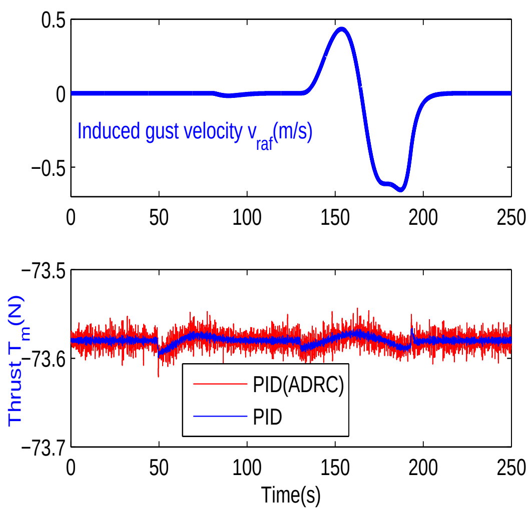


Figure 8: The induced gust velocity v_{raf} and the variations of the main rotor thrust T_M.

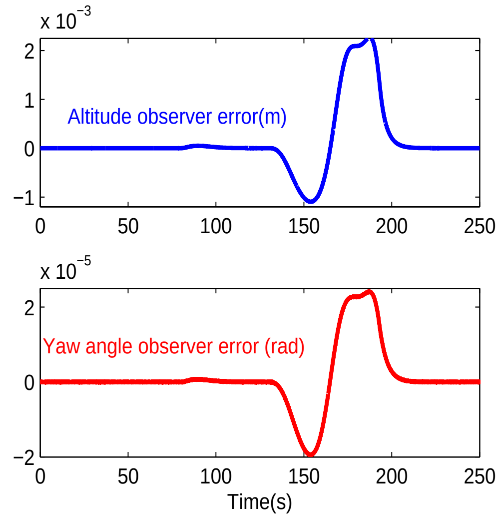


Figure 10: Observer error in z and ϕ.

REFERENCES

Byrnes, C., Priscoli, F., Isidori, A., and Kang, W. (1997). Structurally stable output regulation of nonlinear systems. *Automatica*, 33(2):369–385.

Gao, Z., Hu, S., and Jiang, F. (2001). A novel motion control design approach based on active disturbance rejection. pages 4877–4882. Orlando, Florida USA, Proceedings of the 40th IEEE Conference on Decision and Control.

G.D.Padfield (1996). *Helicopter Flight Dynamics: The Theory and Application of Flying Qualities and Simulation Modeling*. Blackwell Science LTD.

Gokçek, C., Kabamba, P., and Meerkov, S. (2000). Disturbance rejection in control systems with saturating actuators. *Nonlinear Analysis*, 40:213–226.

Hamdan, M. and Gao, Z. (2000). A novel pid controller for pneumatic proportional valves with hysteresis. *IEEE Industrial Application Society 2000 Annual Meeting and World Conference on Industrial Applications of Electrical Energy*, 2:1198 – 1201.

Han, J. (1995). A class of extended state observers for uncertain systems. *Control and Decision*, 10(1):85–88.

Han, J. (1999). Nonlinear design methods for control systems. Beijing, China, The Proc of the 14th IFAC World Congress.

Hou, Y., Z. Gao, F. J., and Boulter, B. (2001). Active disturbance rejection control for web tension regulation. *Proceedings of the 40th IEEE Conference on Decision and Control, Orlando, Florida USA*, pages 4974–4979.

Isidori, A. (1995). *Nonlinear Control Systems*. Springer-Verlag, Berlin, 3rd edition edition.

Jiang, F. and Gao, Z. (2001). An application of nonlinear pid control to a class of truck abs problems. *IEEE Conference on Decision and Control.Orlando, FL, USA*, 1:516–521.

Marconi, L. and Isidori, A. (2000). Robust global stabilization of a class of uncertain feedforward nonlinear systems. *Systems and Control Letters*, 41:281–290.

Marten, M., Ludwick, S., and Trumper, D. (2005). A loop shaping perspective for tuning controllers with adaptive feedforward cancellation. *Precision Engineering*, 29:27 –40.

Martini, A., Léonard, F., and Abba, G. (2005). Suivi de trajectoire d'un hélicoptère drone sous rafale de vent[in french]. *CFM 17ème Congrès Français de Mécanique. Troyes, France, CD ROM.*$N^0.467$.

Slotine, J. and Li, W. (1991). *Applied nonlinear control*. Prentic-Hall, Inc.

Vilchis, A., Brogliato, B., Dzul, L., and Lozano, R. (2003). Nonlinear modeling and control of helicopters. *Automatica*, 39:1583 –1596.

Wei, W. (2001). Approximate output regulation of a class of nonlinear systems. *Journal of Process Control*, 11:69–80.

Zeller, J., Zhu, M., T.Stimac, and Gao, Z. (2001). Nonlinear digital control implementation for a dc-dc power converter. *Proc. of* 36^{th} *Intersociety Energy Conversion Engineering Conference IECEĆ01, Savannah, Georgia*.

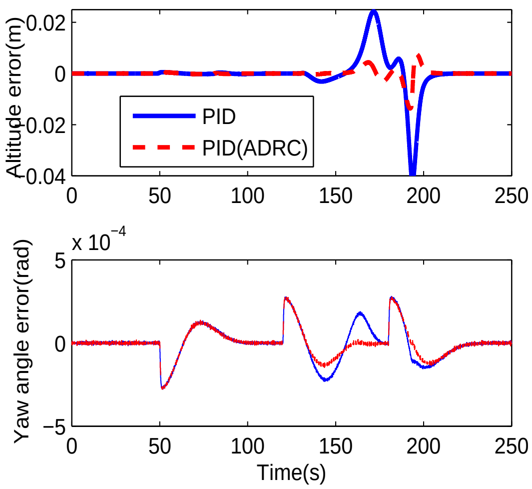


Figure 11: Test of a large disturbance $v_{raf} = 3m/s$.

BEHAVIOR BASED DESCRIPTION OF DEPENDABILITY
Defining a Minium Set of Attributes for a Behavioral Description of Dependability

Jan Rüdiger, Achim Wagner and Essam Badreddin
Automation Laboratory, University of Mannheim, B6, 23-29, Building B, EG, 68131 Mannheim, Germany
ruediger@uni-mannheim.de, a.wagner@ti.uni-mannheim.de, badreddin@ti.uni-mannheim.de

Keywords: Fault-tolerant systems, Autonomous systems, Behavioral systems.

Abstract: Dependability is widely understood as an integrated concept that consists of different attributes. The set of attributes and requirements of each attribute varies from application to application thus making it very challenging to define dependability for a broad amount of application. The dependability, however, is of great importance when dealing with autonomous or semi-autonomous systems, thus defining dependability for those kind of system is vital. Such autonomous mobile system are usually described by their behavior. In this paper a minimum set of attributes for the dependability of autonomous mobile systems is proposed based on a behavioral definition of dependability.

1 INTRODUCTION

Complex computing systems, such as network computers, computer controlled plants or flight controll systems need not only to fulfill their functional but also their non-functional properties like availability, reliability, safety, performance, dependability etc. Non-functional properties reflect the overall quality of a system. Besides performance the dependability is getting a more important non-functional requirement of a system.

The dependability is usually understood as an integrated concept (Avizienis et al., 2004b; Avizienis et al., 2004a; Randell, 2000; Candea, 2003; Dewsbury et al., 2003) that further consists of attributes that affect the dependability of the system. The set of attributes and the requirements on each attribute vary from application to application. This makes it hard to define dependability for a broad amount of applications.

The dependability of a system is particularly important when dealing with autonomous or semi-autonomous systems. With an increasing degree of autonomy and safety requirements the requirements for dependability increase hence being able to measure and compare the dependability of these system is getting more and more important.

In this paper a minimum set of attributes for the dependability of autonomous mobile systems is proposed.

This paper is outlined as follows: In Section 2 a description for systems on which dependability is usually defined is presented. Since the dependability definition used throughout this paper is based on a different definition of a system the equivalence of the two system definitions is shown. In Section 3 the different definitions used in the literature are used and again compared to the behavior based definition used throughout this paper. Section 4 summarizes the attributes of dependability and a minimum set of those attributes is proposed based on the behavioral definition of dependability and of the attributes. The paper ends with the discussion of the set in Section 5 and the conclusion.

2 SYSTEM

According to (Randell, 1999; Avizienis et al., 2004b; Avizienis et al., 2004a; Jones, 2003) the system for which dependability will be discussed is described by its

- functional and non-functional properties,

- the boundaries of the system,
- the environment the system is designed for,
- the system behavior,
- the service the system delivers, and
- its structure.

In Wikipedia a **System** (from the Latin (systēma), and this from the Greek συστημα (sustēma)) is defined as an assemblage of entities/objects, real or abstract, comprising a whole with each and every component/element interacting with or related to at least one other component/element. Any object which has no relationship with any other element of the system, is not a component of that system. A subsystem is then a set of elements, which is a system itself, and a part of the whole system.

In this view it is equal wether a system is connected to another system or to a user, who is again treated as a system.

A system is usually defined by its functional and non-functional properties. The functional properties define specific behaviors of the system or subsystem while the non-functional properties define overall characteristics of the system. Thus, the non-functional properties define properties the system must satisfy while performing its functional properties. Among other things the non-functional properties of a system are: functionality, performance, availability, dependability, stability, cost, extensibility, scalability, manageability, application maintainability, portability, interface, usability and safety. This list is non-exhaustive since the non-functional properties of a system are highly system specific (Torres-Pomales, 2000; Sutcliffe and Minocha, 1998; Franch and Botella, 1998). When systems or sub-systems interact with each other or with their environment the common boundaries of those systems as well as the environment itself must be defined. A system acting well in the specified environment may fail in an environment its was not designed for. The system boundary defines the scope of what the system will be and as such defines the limits of the system.

The behavior of the system is how the system implements its intended function. The behavior of a dynamic system as defined (Willems, 1991) is a time trajectory of the legal states of the system. The legal states of the system are further divided into external and internal states. External states of a system are those which are perceivable by the user or another system. The external states thus define the interface of the (sub-)system. The remaining states are internal.

The service the system delivers is its visible behavior to the user or another system. According to the above definition of behavior this is the time trajectory of its external states.

Last but not least the structure of the system defines how the system is partitioned into sub-systems and how those sub-systems are connected to each other and how the system is „connected" to the environment. The structure of the system also defines how the communication of the sub-systems is organized.

When dealing with autonomous mobile robots the system is often viewed as a black box and described by its behavior. The behavioral approach is very common when dealing with autonomous mobile robots (Brooks, 1986; Michaud, ; Jaeger, 1996). The framework of Willems (Willems, 1991) is used for describing a system by its behavior. In this framework a dynamical system is defined to be „living" in an universe $\mathbb{U}$.

Definition 2.1 *A* dynamical system $\sum$ *is a triple* $\sum = (\mathbb{T}, \mathbb{W}, \mathfrak{B})$ *with* $\mathbb{T} \subseteq \mathbb{R}$ *the time axis,* $\mathbb{W}$ *the signal space, and* $\mathfrak{B} \subseteq \mathbb{W}^{\mathbb{T}}$ *the behavior.*

A mathematical model of a system claims that certain outcomes are possible, while others are not. This subset is called the *behavior* of the system. The behavior $\mathfrak{B}$ is thus the set of all admissible trajectories. The universe $\mathbb{U}$ is the equivalence to the environment as described above and the behavior $\mathfrak{B}$ is the equivalence to function of the system. In (Rüdiger et al., 2007) the definition of a dynamical system is extended by a set of *basic and fused behaviors* $\mathbb{B}$ and by a mission w_m of the system which is the equivalence of the service the system is intended to deliver. Such a system is defined as:

Definition 2.2 *Let* $\Sigma = (\mathbb{T}, \mathbb{W}, \mathfrak{B})$ *be a time-invariant dynamical system then* $B \subseteq \mathbb{W}^{\mathbb{T}}$ *is called the set of* basic behaviors $w_i(t) : \mathbb{T} \to \mathbb{W}$, $i = 1...n$ *and* $\mathbb{B}$ *the set of fused behaviors.*

B is a set of trajectories in the signal space $\mathbb{W}$. The set of basic behaviors B of an autonomous system, in contrast to the behaviors $\mathfrak{B}$ of a dynamical system as defined in (Willems, 1991), is not the set of admissible behaviors, but solely those behaviors which are given to the system by the system engineer (programmer).

The mission of such a system is defined as:

Definition 2.3 *Let* $\Sigma = (\mathbb{T}, \mathbb{W}, \mathfrak{B})$ *be a time-invariant dynamical system. We say the* mission w_m *of this system is the map* $w_m : \mathbb{T} \to \mathbb{W}$ *with* $w_m \in \mathfrak{B}$.

A dynamical system can, like the system described above, be divided into subsystem having their own behavior. This definition of system and behavior is used throughout this paper.

3 DEFINITION OF DEPENDABILITY

Beside the other mentioned non-functional properties of a system the dependability is getting a more important non-functional property.
The general, qualitative, definitions for *dependability* used in the literature so far are:

> **Carter** (Carter, 1982): A system is dependable if it is trustworthy enough that reliance can be placed on the service it delivers.
>
> **Laprie** (Laprie, 1992): Dependability is that property of a computing system which allows reliance to be justifiably placed on the service it delivers.
>
> **Badreddin** (Badreddin, 1999): Dependability in general is the capability of a system to successfully and safely fulfill its mission.
>
> **Dubrova** (Dubrova, 2006): Dependability is the ability of a system to deliver its intended level of service to its users.

All four definitions have in common that they define dependability on the service a system delivers and the trust that can be placed on that service. As mentioned before the service a system delivers is the behavior as it is perceived by the user, which in our case is also called the mission of the system.
A more quantitative definition for dependability used in (Avizienis et al., 2004a) is:

> **Dependability** of a system is the ability to avoid service failures that are more frequent and more severe than is acceptable by the user(s).

This definition, however, does not directly include the service the system is intended to deliver nor does it include the time up to which the system has to deliver the intended service.
Derived from the above definitions and the behavioral definition of a system a behavior-based definition for dependability for autonomous mobile robots was introduced in (Rüdiger et al., 2007). This includes the definition of a mission which coresponds with the service mentioned above.

4 ATTRIBUTES OF DEPENDABILITY

According to (Avizienis et al., 2004b; Avizienis et al., 2004a; Randell, 2000) the dependability is an inte-

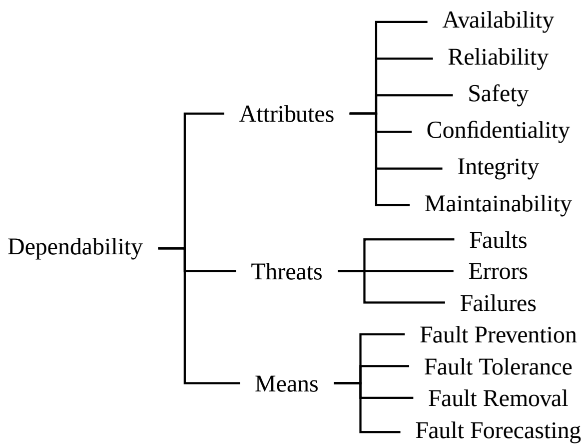


Figure 1: The dependability tree.

grated concept that further consists of the attributes (see also Figure 1)

- **Availability** readiness for correct service,
- **Reliability** continuity of correct service,
- **Safety** absence of catastrophic consequences for the user(s) and the environment,
- **Confidentiality** absence of unauthorized disclosure of information,
- **Integrity** absence of improper system state alteration and
- **Maintanability** ability to undergo modifications and repairs.

In (Candea, 2003) only reliability, availability and safety together with security is listed; however, security is seen as an additional concept as described below.
In (Dewsbury et al., 2003) the dependability attributes for home systems are defined as:

- **Trustworthiness** the system behaves as the users expects,
- **Acceptability** a system that is not acceptable will not be used,
- **Fitness for its purpose** the system must fit the purpose it was designed for and
- **Adaptability** the system must evolve over time and react to changes in the environment and the user.

The dependability specifications of a system must set requirements for the above attributes. Based on a specific system the dependability of the system depends on those requirements for a subset or all of the above attributes. Since the (sub-)systems are designed in a behavioral context it is common to also describe the attributes of dependability in a behavioral context or the other way round to describe the requirements for the attributes on the behavior of the (sub-)system. Before further describing the attributes it is, however, important to define a priority for the attributes.

4.1 Safety

For autonomous mobile robots the main attribute is, or should be, safety. The attribute safety is not to be mistaken with the attribute security which is a combination of the attributes confidentiality, integrity and availability and as thus an additional concept (Samarati and Jajodia, 2000; Cotroneo et al., 2003; Cera et al., 2004). For a comparison of security and dependability see (Meadows and McLean, 1999). Even if the the intended service of the system cannot be fullfilled the safety requirements of the system are not allowed to be violated. Thus, the requirement on the behavior of the system, as defined in section 2, is that it must always fullfill its safety requirements.
From a reliability point of view, all failures are equal. In case of safety, those failures are further divided into *fail-safe* and *fail-unsafe* ones. Safety is reliability with respect to failures that may cause catastrophic consequences. Therefore, safety is unformaly defined as (see e.g. (Dubrova, 2006)):

> Safety $S(t)$ of a system is the probability that the system will either perform its function correctly or will discontinue its operation in a fail-safe manner.

In (Rüdiger et al., 2007) an area $\mathfrak{S}$ around the behavior of the system $\mathfrak{B}$ is introduced, which leads to catastrophic consequences when left. Safety of a system Σ is then defined as:

Definition 4.1 *Let* $\Sigma = (\mathbb{T}, \mathbb{W}, \mathfrak{B})$, $\mathbb{T} = \mathbb{Z}$ *or* $\mathbb{R}$, *be a time-invariant dynamical system with a safe area* $\mathfrak{S} \supseteq \mathfrak{B}$. *The system is said to be* safe *if for all* $t \in \mathbb{T}$ *the system state* $w(t) \in \mathfrak{S}$.

The definition is illustrated in Figure 2. This definition is consistent with the idea that a safe system is either operable or not operable but in a safe state.

4.2 Availability Vs Realiability

Reliability means (Dubrova, 2006):

> Reliability $R|_t$ is the probability that the system will operate correctly in a specified operating environment in the interval $[0,t]$, given that it worked at time 0.

An autonomous system is, thus, said to be reliable if the system state does not leave the set of admissible trajectories $\mathfrak{B}$. In contrast to reliability the availability is defined at a time instant t while the reliability is defined in a time interval.

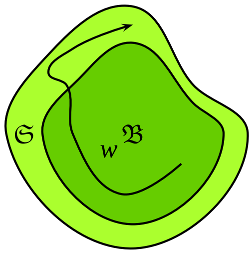


Figure 2: Safety: The system trajectory w leaves the set of admissible trajectories $\mathfrak{B}$ but is still considered to be safe since it remains inside $\mathfrak{S}$.

> Availability $A|_t$ is the probability that a system is operational at the instant of time t.

Availability is typically important for real-time systems where a short interrupt can be tolerated if the deadline is not missed. This also holds for autonomous mobile systems. In (Rüdiger et al., 2007) the availability is defined as:

Definition 4.2 *Let* $\Sigma = (\mathbb{T}, \mathbb{W}, \mathfrak{B})$, $\mathbb{T} = \mathbb{Z}$ *or* $\mathbb{R}$, *be a time-invariant dynamical system. The system is said to be* available *at time t if* $w(t) \in \mathfrak{B}$. *Correspondingly, the availability of the system is the probability that the system is available.*

For dependable autonomous mobile systems as defined above requirements for reliability are redundant and can be omitted. In case of reliability and availability it is sufficient to define requirements for the availability.

4.3 Maintainability

A maintainable system is „able to react“ either autonomously or by human interaction to changes in the system and the environment.

> Maintainability is the ability of a system to undergo modification and repairs.

While the requirements for the first two attributes rather passively define the dependability of a system, the maintainability gives the system the ability to react to changes. An event that would reduce or violate the dependability of the system can counteract to recover the dependability. In (Rüdiger et al., 2007) the maintainability is defined as:

Definition 4.3 *A dynamical system* $\Sigma = (\mathbb{T}, \mathbb{W}, \mathfrak{B})$ *with the behaviors* $\mathbb{B}$ *is said to be* maintainable *if for all* $w_1 \in \mathbb{W}$ *a* $w_2 \in \mathfrak{B}$ *and a* $w : \mathbb{T} \cap [0,t] \to \mathbb{W}$ *exist,*

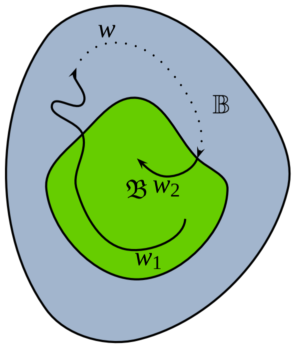


Figure 3: Maintainability: The system trajectory w_1 leaves the set of admissible trajectories $\mathfrak{B}$ and is steered back to $\mathfrak{B}$ with the trajectory $w \in \mathbb{B}$.

with $w' : \mathbb{T} \to \mathbb{W}$ *defined by:*

$$w'_{(t')} = \begin{cases} w_{1(t')} & \text{for } t' < 0 \\ w_{(t')} & \text{for } 0 \leq t' \leq t \\ w_{2(t'-t)} & \text{for } t' > t \end{cases}$$

The definition is illustrated in Figure 3. An autonomous mobile system is said to be maintainable if it is able to steer the system from any trajectory $w \notin \mathfrak{B}$ back to the set of admissible trajectories $\mathfrak{B}$ in time $[0,t]$.

4.4 Confidentiality and Integrity

Confidentiality has been defined by the International Organization for Standardization (ISO) as ”ensuring that information is accessible only to those authorized to have access”. This attribute is very important for systems like operating systems or transaction systems. For autonomous mobile robots, however, this attribute is underpart. If informations of the system will be available un-authorized then this will not reduce the dependability of the autonomous mobile system. For this attribute the functions of the underlaying operating system are used.
When a program is executed on a system it is usually checked whether the program is allow to be runned by the user. *Integrity* ensures that the program flow and the information of the program will not be altered during the execution. Even if a change, wether it was on purpose, by an external or by soft- or hardware failure, in the program flow could be severe this aspect is already covered by the safety attribute.

5 DISCUSSION

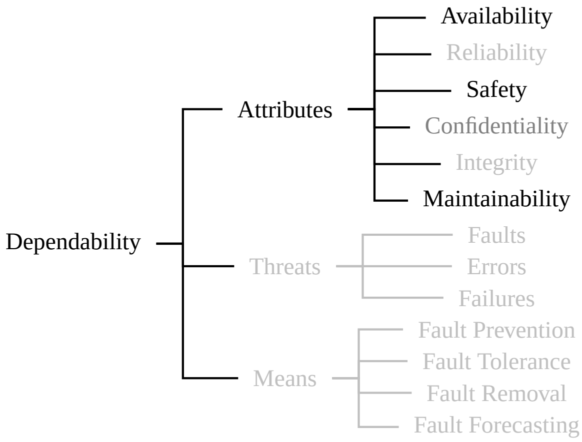


Figure 4: The resulting dependability tree.

The resulting dependability tree for autonomous mobile systems is shown in Figure 4. The requirements for the safety assures that failures in the system will not lead to catastrophic consequences. The requirements for the availability assures that the system is operational at the desired time instances t and finally the maintainability requirements assures that even in case of changes of the system or the environment the system is able to react and modify itself to maintain the dependability of the system.

6 CONCLUSION

Dependability is part of the non-functional properties of a system which reflect the overall quality of a system. Qualitative definitions for dependability like in (Carter, 1982; Laprie, 1992; Badreddin, 1999; Dubrova, 2006) further divide the dependability into attributes. Those attributes are again rather qualitative and also not distinct. Autonomous mobile systems are often described by their behavior. This aspect was utilized in this paper to propose a minimum subset of the attributes of dependability, as defined in (Rüdiger et al., 2007), which are defined quantitative and can still ensure the dependability of the autonomous mobile system.

REFERENCES

Avizienis, A., Laprie, J.-C., and Randell, B. (2004a). Dependability and its threats: A taxonomy.

Avizienis, A., Laprie, J.-C., Randell, B., and Landwehr, C. (2004b). Basic concepts and taxonomy of dependable and secure computing. *IEEE Trans. on Dependable and Secure Computing*, 1(1):11–33.

Badreddin, E. (1999). Safety and dependability of mechatronics systems. In *Lecture Notes*. ETH Zürich.

Brooks, R. A. (1986). A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1):14–23.

Candea, G. (2003). The basics of dependability.

Carter, W. (1982). A time for reflection. In *Proc. 12th Int. Symp. on Fault Tolerant Computing (FTCS-12)*. FTCS-12) IEEE Computer Society Press Santa Monica.

Cera, C. D., Kim, T., Han, J., and Regli, W. C. (2004). Role-based viewing envelopes for information protection in collaborative modeling.

Cotroneo, D., Mazzeo, A., Romano, L., and Russo, S. (2003). An architecture for security-oriented perfective maintenance of legacy software.

Dewsbury, G., Sommerville, I., Clarke, K., and Rouncefield, M. (2003). A dependability model for domestic systems. In *SAFECOMP*, pages 103–115.

Dubrova, E. (2006). Fault tolerant design: An introduction. Draft.

Franch, X. and Botella, P. (1998). Putting non-functional requirements into software architecture.

Jaeger, H. (1996). Brains on wheels: Mobile robots for brain research.

Jones, C. (2003). A formal basis for some dependability notions.

Laprie, J. C. (1992). *Dependable computing: Basic concepts and terminology*. Ed. Springer Verlag, 1992.

Meadows, C. and McLean, J. (1999). Security and dependability: then and now. In *Computer Security, Dependability, and Assurance: From Needs to Solutions, 7-9 July 1998 & 11-13 November 1998, York, UK & Williamsburg, VA, USA*, pages p.166–70. Los Alamitos, CA, USA : IEEE Comput. Soc, 1999.

Michaud, F. Adaptability by behavior selection and observation for mobile robots.

Randell, B. (1999). Dependability - a unifying concept.

Randell, B. (2000). Turing Memorial Lecture: Facing up to faults. *j-COMP-J*, 43(2):95–106.

Rüdiger, J., Wagner, A., and Badreddin, E. (2007). Behavior based definition of dependability for autonomous mobile systems. European Control Conference.

Samarati, P. and Jajodia, S. (2000). Data security.

Sutcliffe, A. G. and Minocha, S. (1998). Scenario-based analysis of non-functional requirements.

Torres-Pomales, W. (2000). *Software Fault Tolerance: A Tutorial*.

Willems, J. (1991). Paradigms and puzzles in the theory of dynamical systems. *Automatic Control, IEEE Transactions on*, 36(3):259–294.

A HUMAN AIDED LEARNING PROCESS FOR AN ARTIFICIAL IMMUNE SYSTEM BASED ROBOT CONTROL

An Implementation on an Autonomous Mobile Robot

Jan Illerhues and Nils Goerke
Division of Neural Computation, Department of Computer Science
University of Bonn, Bonn, Germany
illerhues@nero.uni-bonn.de, goerke@nero.uni-bonn.de

Keywords: Autonomous robot, systemic architecture, learning, artificial immune system, rule-like association (RLA).

Abstract: In this paper we introduce a pre-structured learning process that enables a teacher to implement a robot controller easily. The controller maps sensory data of a real, autonomous robot to motor values. The kind of mapping is defined by a teacher. The learning process is divided into four phases and leads the agent from a two stage supervised learning system via reinforcement learning to unsupervised autonomous learning. In the beginning, the controller starts completely without knowledge and learns the new behaviours presented from the teacher. In second phase is dedicated to improve the results from phase one. In the third phase of the learning process the teacher gives an evaluation of the states zhat the robot has reached performing the behaviours taught in phase one and two. In the fourth phase the robot gains experience by evaluating the transitions of the different behavioral states. The result of learning is stored in a rule-like association system (RLA), which is inspired from the artificial immune system approach. The experience gained throughout the whole learning process serves as knowledge base for planning actions to complete a task given by the teacher. This paper presents the learning process, its implementation, and first results.

1 INTRODUCTION

Nowadays the development of a robot control is mostly reserved for professionals. To integrate robots into peoples' everyday lives, it would be necessary to give the user direct access to the robot's behaviours. A learning process would therefore be helpful, in which the user acts as a teacher, showing the robot how to act according to a special behaviour, without any programming skills. By repeating this learning process, the robot would gain more and more experience, enabling it to learn different behaviours. If the repertoire of behaviours is large enough, the robot would become capable of using its past experiences to choose the best behaviour to fulfill a given task.

In this paper we introduce the implementation and first results of such a learning process on a real robot. The presented implementation is based on rule-like associations (RLA) (Hart et al., 2003), as derived from an artificial immune system approach. During the learning process, a system of RLAs is created as a knowledge-base, storing the learned results. This knowledge base is used to choose robot actions when faced with previously learned situations. The robot is trained in real time, on-line. The teacher is providing training input while the robot is performing actions. To give a maximal flexibility and mobility during training, the interface between the teacher is implemented as a PDA with wireless connection to the moving robot. The learning process is divided into four phases, which influence the RLA system following different learning paradigms: supervised, reinforcement, unsupervised. The presented implementation extends and accelerates an approach (Rattenberger et al., 2004), which provides an unsupervised learning RLA system without planning and task fulfilling skills.

2 THE COMPLETE SYSTEM

The learning program was implemented on the autonomous, mobile robot "KURT2" (KTO, 1995). A sensory upstream prepares the sensory data of the

Figure 1: The robot ”KURT2”.

robot for the learnable robot controller. A motor downstream converts the controller output to motor values, which could be directly addressed to the robot. Both data streams match the requirements of the Systemic Architecture (Goerke, 2001). The learning controller is responsible for the scheduling of the teacher's/robot's communication, the management of the knowledge-base (RLAs), the mapping of sensory data to motor values and for choosing the right behaviour to fulfill a task.

2.1 The Agent

The robot ”Kurt2” (see figure 1) is an autonomous, mobile agent. A notebook is fixed onto the robot's chassis, running the robot's control software which has a direct CAN-Bus connection to the robot.

The agent has a row with six supersonic sensors and eight infrared sensors 14 cm above ground. A nother row of 10 infrared sensors is installed 20 cm above ground. The supersonic sensors are able to detect objects nearly parallel to the robot's chassis from a distance of 10 cm to 70 cm. The values of the infrared sensors range from 0 to 550. They represent distances from 10 cm (value 550) to 80 cm (value 0), but may be very noisy. The mapping of the sensor values to the distance of the object is non-linear (see figure 2).

There are three wheels on each side of the robot which are hard connected, thus doing exactly the same; each side is controlled by one motor. Thes two motors are able to move the robot forward and backward, with speeds of up to $0.9\frac{m}{sec}$ in either direction. Setting the motor speed of each side individually enables the robot to not only drive straight ahead, but also to make curves, u-turns and up to 360° rotations.

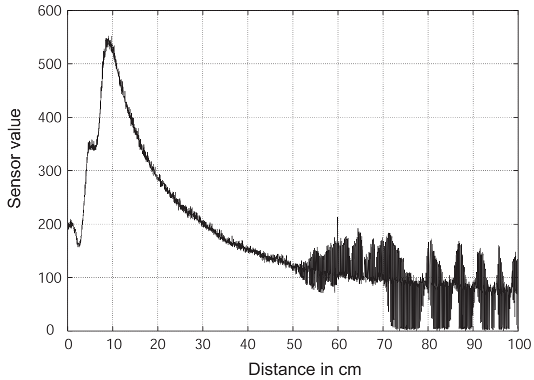


Figure 2: Mapping of distance to infrared sensor values.

2.2 The Rule-Like Association System (RLA-System)

A rule-like association (Hart et al., 2003)(Webb et al., 2003) maps a situation (presented by sensor values) to motor values. Accordingly each RLA consists of three sections: The first section **C** contains a (partial) sensory description of a situation (condition **C**). The second section is a robot action command **A**. The third section of the used RLA-implementation contains a description of the expected result **E** after action **A** was performed in the situation described by **C**. Therefore, a RLA describes a transition of states. The RLA X whose condition part **C** describes the current state of the robot the best, is chosen to control the robot. Its corresponding action **A** is then used to command the robot in its current state. After that, the robot is in a new situation, and once again the RLA with the closest matching **C** condition is chosen to control the robot next. In this way the RLA controller performs a reactive behaviour (Arkins, 1998).

Taking this concept one step further, a network of RLAs could be created (Ross, 2003). Thus making it possible to store relations between different RLAs. We built a RLA network in which each RLA is represented by a different node. An edge in the network represents a successive relation: If RLA Y is chosen after a RLA X was chosen, the network creates an edge from node X to node Y. In that way the RLA network is built (see figure 3).

2.3 The User Interface

A PDA is used as interface between the robot and the teacher. The PDA communicates with the notebook installed on the robot ”KURT2” per WLAN. W ith this interface the user may send different information to the robot:

- motor commands

The user may send the robot direct motor commands.

- behaviours
  The user can choose the behaviour that the robot should perform.
- reinforcements
  The user may give the robot reinforcements to help evaluate situations or actions.
- organising commands
  These are commands that organise learning, e.g. cause an output of the RLA structure into a file.

2.4 Systemic Architecture

The Systemic Architecture contains the sensory upstream and the motor downstream.

The upstream collects the actual sensory data from the robot as a vector of all sensor values and user inputs. Afterwards, it performs a preprocessing of the sensory values. The sensory values from the ultrasonic sensors have shown to be not reliable, because the beam of these ultrasonic sensors might be reflected in an unintended way. Therefore we have only used the infrared sensor values.

The front-side of the robot is resolved with five sensors. On the left side of the robot however, we use only two sensors. The difference of these two sensor values is used to detect the position to a wall (turned towards, turned away, or parallel). Additionally, we use their maximum value to identify if any object is detected on the left side. Thus from this information we gain two virtual sensor values: one for positioning purposes, and the other for object detection. The right side is resolved the same way.

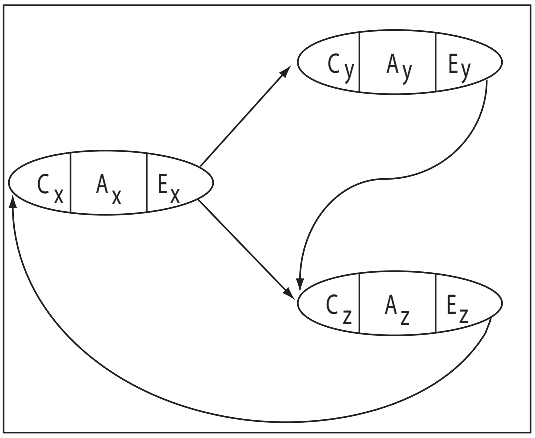


Figure 3: Example of a RLA system with 3 RLAs. Choosing RLA X and performing its action $\mathbf{A}_x$ could lead to the situation described in $\mathbf{C}_y$, or to the situation described in $\mathbf{C}_z$. Choosing RLA Y leads to the situation $\mathbf{C}_z$. RLA Z is antecessor of RLA X.

The maximum of the backwards sensors is used to receive object detection on the back side. All in all we gain a vector of 10 (virtual) sensor values. We receive further virtual sensor values by logging a history, which contains the average of the last five values of each of these 10 sensors. These are added to the sensor stream, and as a result we obtain a 20 dimensional vector.

The values of this 20 dimensional vector are discretised in the following way: The front sensors and their history values are translated into five discrete values: 0 for ”Out-Of-Sight” (distance $>$ 70 cm), 1 for ”Far Away” (70 cm $\geq$ distance $>$ 40 cm), 2 for ”Middle-distance” (40 cm $\geq$ distance $>$ 20 cm), 3 for ”Close” (20 cm $\geq$ distance $>$ 12 cm) and 4 for ”Emergency” (12 cm $\leq$ distance).

The values from the virtual sensor describing the positioning are discretised as follows: 0 for ”Hard towards the wall”, 1 for ”Slightly towards the wall”, 2 for ”Parallel to the wall”, 3 for ”Slightly away from the wall” and 4 for ”Hard away from the wall”.

The object detection is binary coded with ”0” for no object, and ”1” for an object detected.

Using this vector to describe a situation we receive a state space with $(5^5 \cdot 5^2 \cdot 2^3)^2 \cdot N = 390.625.000.000 \cdot N$ states, where N is the amount of behaviours.

The upstream also analyses the user input. Each behaviour has a characteristic number, which extends the 20 dimensional vector by one component. The resulting 21 dimensional vector serves as input to the RLA system in the controller. All other user inputs are forwarded directly to the controller (see figure 5 and 6).

3 PHASE 1- BASIC TRAINING

In this section the robot learns the basic movements of a behaviour B by reactively planning its action. In section 2.2 we described a RLA system, and the way it could help the robot plan its action. The robot learns a new behaviour B by creating new RLAs associated with this behaviour. These RLAs describe situations reached by performing the appropriate actions. At the beginning the robot is completely unaware of this behaviour, because it has no RLAs and therefore all situations are unknown.

In this phase of the implementation, our aim is to create adequate RLAs for the behaviour B. For each new RLA we have to generate a situation-describing condition part $\mathbf{C}$, an action command $\mathbf{A}$, and an expectation $\mathbf{E}$. The controller works round-based, comparing the current situation of the robot with the condition **C** of each RLA. If a RLA was found that describes the situation adequately, its action **A** controls the robot. If none is found, the scheme creates a new RLA that matching the current situation **C**, and performing its (new) action command **A**.

3.1 Find an Adequate RLA

In section 2.4 we introduced a vector which is used to represent the robot's current situation and current behaviour. To aid in the comparison between the **C** parts of the RLAs and the current situation, the description in part **C** is created in the same form as the input for the current situation: a 21 dimensional vector, K.

It is now possible to measure the similarity between the current situation S and the **C** part of a RLA containing vector K. This is done by building the difference $s_i - k_i$ between each i components of both vectors. The more different the vectors are, the more penalty points P the corresponding RLA gets. Great differences should be rated worse than small differences. Therefore we use the quadric difference $(s_i - k_i)^2$ to calculate the penalty points. In respect to this, penalty points P of the RLA ID are calculated as follows:

$$P_{ID} = \sum_{i=0}^{20} (s_i - k_i)^2$$

The RLA with the smallest P value is therefore the RLA with the best description of the current situation, and could be used to control the robot. If the penalty points of this RLA are too high however, its description of the current situation is not satisfactory. To implement this we define a similarity radius. If the penalty points are within the similarity radius, the "winning" RLA could be used to control the robot. If the minimum of the penalty points is outside of the similarity radius, the current situation is declared as unknown, the motors are stopped, and the controller needs to create a new RLA.

For some behaviours, the importance of different sensor-types may vary. For example, the front sensors are very important for a collision avoidance behaviour, if the main moving direction is forward. In this case the position to the wall is not important. In contrast, the positioning sensors are necessary when performing a "wall-following". To account for these differences, we define an attention focus D. This variable D is a 21 dimensional vector of integer numbers. The higher the number d_i, the more attention the corresponding sensor value s_i gets. With respect to the attention focus, the penalty points are calculated as follows:

$$P_{ID} = \sum_{i=0}^{20} d_i \cdot (s_i - k_i)^2$$

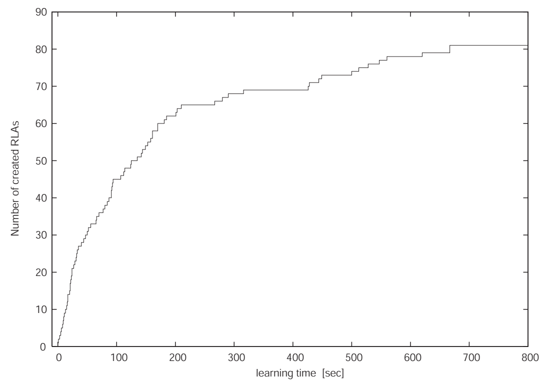


Figure 4: This diagram shows the learning of Wall-Following behaviour. During learning 93 RLAs were created in phase 1 to adequately perform the desired behaviour.

If d_i is large, just a small difference of $s_i - k_i$ could lead to a penalty point value outside of the similarity radius. If $d_i = 0$ however, then the corresponding sensor to s_i has no influence on the penalty points. This sensor is not necessary to perform the specified behaviour. To distinguish between RLAs of the different behaviours, d_{20} is chosen disproportionally high, so that the penalty points of another behaviour's RLA are always outside the similarity range. That means a RLA could only gain control when the robot performs the specific behaviour that the RLA was created for.

3.2 Creating New RLAs

When the robot is stopped because of an unknown situation, we have to find a condition **C** for the new RLA, that describes the current situation S of the robot. Because the vector S and the new condition **C** vector K have the same structure, we can take the present situation S as the new vector K, and get the best possible description of the situation S therefore. After that we need an adequate action **A** for the new RLA. Because the system is in a supervised learning modus, it waits for a teacher's command. The teacher is able to see the robot in its environment, assess the situation, and set the correct motor commands. These motor commands can then be copied into the action part **A** of the new RLA. After that, this new RLA automatically becomes the "winning" RLA, and the new motor command **A** can be performed. In the next round, the subsequent situation can taken to be the expectation **E** of the new RLA. Therefore section **E** of the RLA has the same structure as the situation S and the condition **C** part.

In addition to this, a RLA network is created, which contains all RLAs as nodes. An edge (X,Y) represents a relationship between RLA X and RLA Y. RLA X is an antecessor of RLA Y and RLA Y a

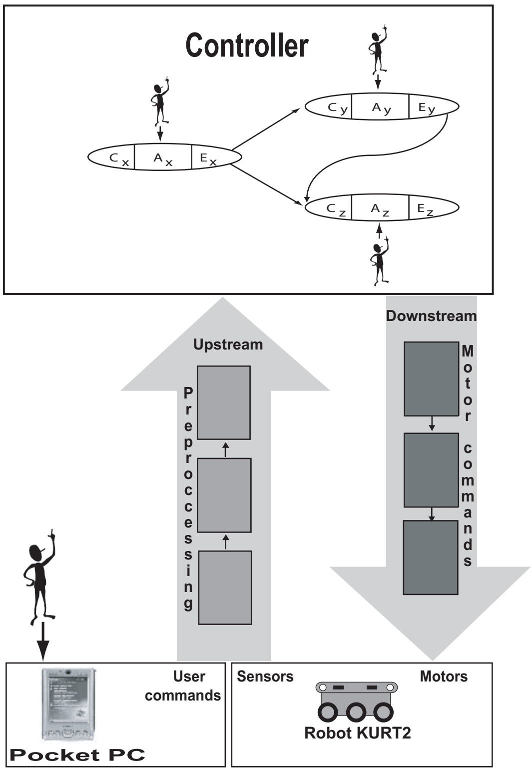


Figure 5: This diagram shows the system used in phase 1 and 2. The controller manages the RLA system and a RLA network with probabilities of transitions between the states.

successor of RLA X. The edges also have weights. A weight of an edge (X,Y) represents the probability of Y being successor of X. Self references (meaning RLA X follows RLA X) are not considered. Thus the sum of the weights of a node's output edges is 1.

In this training phase, the rate of creating new RLAs is very high. The robot reaches unknown situations very often, because it must build up all of the knowledge needed to perform a behaviour. The rate steadily decreases however, until the robot knows most of the situations reached by a behaviour (see figure 4. The end of each training phase is determined by the teacher.

4 PHASE 2- BUILD-UP TRAINING

During the first phase, it is possible that the teacher may have made mistakes, such as sending the incorrect motor commands, or misinterpreting the robot's situation. The purpose of the second phase is to improve the RLA system by correcting these mistakes, as well as getting to know some special situations not encountered in phase 1. At the end of this learning section the robot is able to perform the learned behaviour, and to plan its actions reactively to the teacher's satisfaction. After the teacher finishes this phase, the learning of this behaviour is finished as well. There are three ways in which the teacher may influence the robot's behaviour:

1. Changing a RLA's action command A
   The teacher may interrupt the robot when he notices that a RLA has a false action command. The robots stops and asks for a new action command A, which overrides the old action command of the current RLA. This modification should be made, if the RLA's action is ever inadequate.
2. Creation of a RLA in a known situation
   The teacher can instruct the robot to create a new RLA, forcing the robot improves its movement skills. This helps the robot to distinguish between more situations, thus enabling it to act more precisely.
3. Creation of a RLA in an unknown situation
   This is the same modification of the RLA system as described in the first learning phase.

The two phases 1 and 2, can be seen as two parts of one single supervised training phase. It is not generic to divide the supervised learning into the two phases like we did. During the experiments with the real robot we made the observation that the training changes its character after some initial training steps (pase 1), and the subsequent training (phase 2) was different. Within phase 1 the robot stops very often, demanding for new RLAs to be created. After a while, the robot performs well, performing a rudimentary version of the given task, and stops rather seldom for getting a new RLA. This observation is the motivation behind the two learning phases 1 and 2. Further investigations are necessary to clarify the observed effect.

5 PHASE 3- REINFORCEMENT LEARNING

In this phase the robot learns how to evaluate a situation, based on reinforcements provided by the teacher ((Sutton and Barto, 1998)). The robot begins by performing one of its various behaviours just as it learned it in phases 1 and 2. If the robot reaches a situation that the teacher regards as valuable, the teacher will then send a reinforcement of either ”Good”, or ”Bad”. When the teacher assesses a situation as being good in terms of the associated behaviour, he should reinforce it as ”Good”. Critical situations or situations which were handled less effectively by the robot should be reinforced as ”Bad”. The robot’s task in this phase is to perform a behaviour and learn which situations were reinforced (as ”Good” or as ”Bad”) so that it could provide its own reinforcements in the fourth learning phase.

5.1 Reinforcement-RLAs

In order to learn reinforcements, the robot must be able to memorise situations and to associate them with evaluations. Therefore we use a modified form of a RLA: a Reinforcement-RLA. The Reinforcement-RLA associates a situation with a reinforcement, and consists of only two parts: A situation describing part **C** and an action part implemented as reinforcement-part **R**.

Just as in the previously mentioned RLAs, the condition **C** part is represented by a 21 dimensional vector. The reinforcement part **R** consists of a reinforcement signal (”Good” or ”Bad”).

5.2 Creating a New Reinforcement-RLA

When the teacher gives a reinforcement signal, the PDA sends it to the robot, and the robot creates a new Reinforcement-RLA. The content of the new Reinforcement-RLA’s **C** part (in the form of a 21 dim vector) is the upstream output (vector *S*). This represents the current situation, which has been reinforced. The reinforcement signal received from the PDA is stored in the reinforcement part **R** of the new Reinforcement-RLA. With this procedure, the robot creates a set of Reinforcement-RLAs, one for each given reinforcement.

6 PHASE 4 - UNSUPERVISED REINFORCEMENT

In this phase the robot should generate a reinforcement signal for its current situation by itself (unsupervised). At this point it utilizes of a set of RLAs to perform reactive action planning, and a set of Reinforcement-RLAs to evaluate its own situations accoring to phase 3.

6.1 Reinforce a Situation

The robot performs its reactive action planning according to a behaviour, and every subsequent situation it reaches is then compared to the conditions **C** of the Reinforcement-RLAs. This comparison is executed

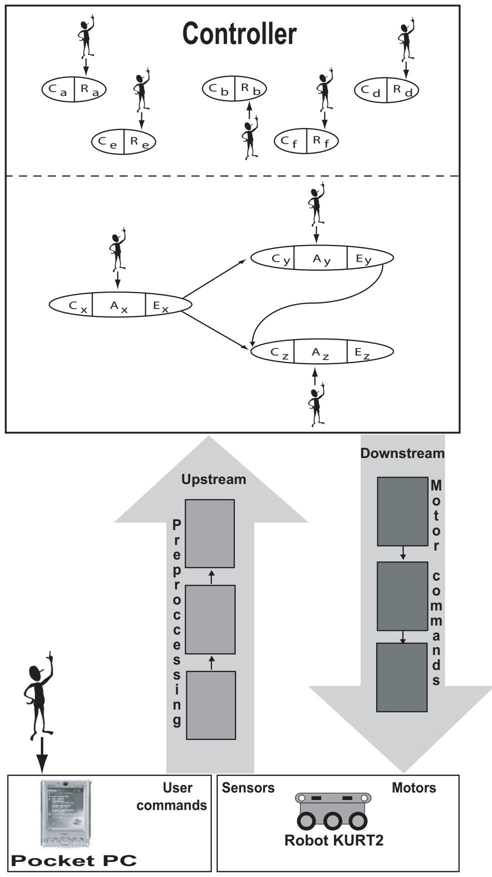


Figure 6: This diagram shows the complete system. The controller manages the RLA system, the Reinforcement-RLAs, and a RLA network with transition’s probabilities and evaluations.

in the same way as described in section 3.1. If any **C** section of a Reinforcement-RLA is similar enough to the current situation, the respective reinforcement signal in the reinforcement part **R** is used to evaluate this situation. The procedure is as follows:

1. The penalty points between the current situation and all Reinforcement-RLAs’ **C** parts are built. The attention focus of this behaviour is the same used in phase 1 and 2.
2. The Reinforcement-RLA with the lowest penalty points is deemed the best.
3. A reinforcement radius is set to express a measure of similarity. If the penalty points are lower than the reinforcement radius, then the situation could be evaluated with the reinforcement signal saved in the part R of the best Reinforcement-RLA. If the penalty points are higher than the Reinforcement-radius, no reinforcement signal is given and the robot could continue reactively planning its actions.

6.2 Consequences of a Reinforcement Signal

With the reinforcement signal, the robot evaluates its previous actions and transitions of states, and therefore creates a new RLA network. This network has the same structure as the one described in section 2.2: Nodes represent RLAs (not Reinforcement-RLAs) and the edges represent transitions of RLAs. These transitions could be interpreted as transitions between states. However, instead of attaching probabilities of these transitions, we log a weight w for their value. Every time the robot extracts a reinforcement signal, it evaluates the most recently used transitions. Therefore a RLA network is initialised where all edge's weights $w_{(x,y)} = 100$. To reinforce a state's transition, we define reinforcement factors rf_j for "Good" and "Bad" reinforcements, where j is an natural number that represents the distance to the reinforced state. After a reinforcement signal is given, we can update the weights $w_{(x,y)}$ of the previous transitions as follows:

$w_{(x,y),i+1} = w_{(x,y),i} \cdot rf_j$

For example, an episode of RLA W, RLA X, RLA Y and RLA Z with a "Good" reinforcement signal is given while RLA Z was performed. The reinforcement-factors rf_j for a "Good" reinforcement should be $rf_1 = 0.9$, $rf_2 = 0.95$ and $rf_3 = 0.97$. The values in the RLA network are then updated as follows:

$$w_{(y,z),i+1} = w_{(y,z),i} \cdot 0.9$$
$$w_{(x,y),i+1} = w_{(x,y),i} \cdot 0.95$$
$$w_{(w,x),i+1} = w_{(w,x),i} \cdot 0.97$$

Therefore we receive a RLA network whose weights represent the effectiveness of a transition.

If $w_{(x,y),i} < 100$, the transition between RLA X and RLA Y is preferred. The lesser the value of $w_{(x,y),i}$ the better the transition. If $w_{(x,y),i} = 100$, the transition between RLA X and RLA Y is neutral. If $w_{(x,y),i} > 100$, the transition between RLA X and RLA Y is avoidable. The higher the value of $w_{(x,y),i}$ the worse the transition.

We call this RLA network "effectiveness network", and the RLA network described in section 3 as containing the probabilities is called "probability network".

The probability network is built during all four phases of learning, but the effectiveness network is only built in the fourth phase. The robot performs reactive action planning, and evaluates the situations it reaches. This fourth phase could last life-long, because at this point the robot is autonomous and no longer requires a teacher.

7 RESULTS OF EXPERIMENTS

We trained three behaviours to show the practicability and the effectiveness of our learning process. We used three different types of attention focus:

1. $(3,3,3,3,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,50)$
2. $(2,2,2,2,2,5,5,1,1,1,1,1,1,1,1,1,1,1,1,1,50)$
3. $(1,1,1,1,1,7,7,5,5,5,1,1,1,1,1,7,7,5,5,5,50)$

7.1 Collision Avoidance

In this behaviour we used attention focus number 1 which considers only front sensors. With this we can simulate a braitenberg vehicle (Braitenberg, 1986) to achieve a collision avoidance behaviour. All other sensors are suppressed.

The robot achieved a satisfying collision avoidance behaviour with 21 RLAs, 13 of them were created during the first phase of learning. The robot took 8:20 minutes to learn this behaviour. (see table 7.4).

The teacher created 11 Reinforcement-RLAs in the third phase, which lasted for five minutes. These Reinforcement-RLAs were used by the robot to evaluate situations by itself for another 15 minutes. After this fourth phase of learning, the best transition in the robot's effectiveness network had a value 0.01 and the worst 159.

7.2 Wall-Following

This behaviour is a left-handed Wall-Follower and the robot learned it with attention focus number 2. This behavior considers the position sensors the most, and the front sensors were necessary to perform in concave edges.

The robot achieved a satisfying wall-following behaviour with 92 RLAs, 81 of which were created during the first phase of learning, which lasted 11:07 minutes. There were only a few new RLAs created in extraordinary situations in the second phase of learning. (see table 7.4).

The teacher created 12 Reinforcement-RLAs in the third phase, which lasted for five minutes. These Reinforcement-RLAs were used by the robot to evaluate the situations it reached by itself for another 10 minutes (see table 7.4). After this fourth phase of learning, the best transition in the robot's effectiveness network had the value 80.26 and the worst 826. This high negative value is caused by a situation in which the robot is too close to a wall, which it corrects in the next step. This situation occurred very often.

7.3 Drawing an 8

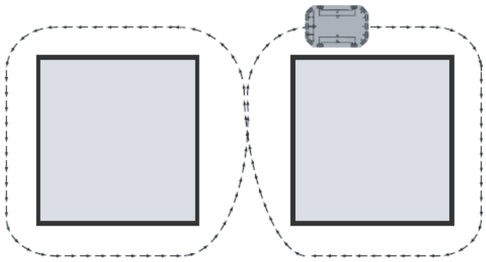


Figure 7: This figure shows the 8-drawing behaviour. The robot must perform two different kinds of wall-following behaviours.

The 8-drawing behaviour combines a left-handed and a right-handed Wall-follower. To teach this behaviour, we used attention focus number 3, which considers the positioning sensors, the object-detection sensors and the history as very important. The robot achieved a satisfying 8-drawing behaviour with 127 RLAs in 12 minutes (see table 7.4). The teacher created 10 Reinforcement-RLAs in the third phase, which lasted five minutes. The robot used these Reinforcement-RLAs to evaluate the situations it reached by itself for another 10 minutes. After this fourth phase of learning, the best transition in the robot's effectiveness network had the value 35.35 and the worst 100. This means that either a negative reinforcement was not given, or it was neutralised by a positive reinforcement.

7.4 Performance

Table 1: Results of the first and second phase learning for the three different tasks: Collision Avoidance, Wall-Following, Draw an 8. The time is the sum of learning time for phase 1 and for phase 2.

Task	Att. focus	Sim. radius	No of RLAs	Time 1+2
Avoid	1	16	21	8:20 min
Wall-Follow	2	16	92	26 min
Draw an 8	3	16	127	12 min

All behaviours were learned to the teacher's satisfaction. The robot performed in real-time and was very effective in learning its behaviours. All behaviours could be taught by an amateur user in reasonably short time. After these learning phases, the robot was able to deliberately and actively plan its next actions, based on the trained RLA network with the probability and/or the effectiveness network (Sutton and Barto, 1998).

8 CONCLUSIONS

Within this paper we presented a novel approach of creating bahaviour based robot controllers. Based on an artificial immune sytem inspired RLA system, the learning process starts with supervised learning via reinforcement learning and reaches unsupervised, autonomous learning capabilities. We have demonstrated, that the presented paradigm is capable of teaching tasks with different complexity with a real robot in reasonably short training time, even for a non-expert robot end-user. Several aspects of the presented works are still open for further interesting investigations: find an universal attention focus, investigate the 2 phases of supervised learning, planning with the use of the RLA network.

The authors are convinced that the presented works is an powerfull alternative to design and train behaviour based robot controllers. The final goal to get rid of the nasty ”low-level” robot programming has come one step further into reach.

We do no longer programm our robot:

We teach the tasks by showing the actions and judge the performed behaviour.

REFERENCES

Arkins, R. C. (1998). *An Behavior-based Robotics*. The MIT Press.

Braitenberg, V. (1986). *Vehicles- Experiments in Synthetic Psychology*. The MIT Press.

Goerke, D. N. (2001). Perspectives for the next decade of neural computation. In *Proc. of NATO Advanced Research Workshop on: Limitations and Future Trends in Neural Computation (LFTNC'01)*, pages 1–9.

Hart, E., Ross, P., Webb, A., and Lawson, A. (2003). A role for immunology in '”next generation”' robot controllers. In *ICARIS03, Artificial Immune Systems, Second International Conference*, pages 46–56.

KTO (1995). KTO- Kommunikation und Technologietransfer Odenthal. http://www.kurt2.de/.

Rattenberger, J., Poelz, P. M., Prem, E., Hart, E., Webb, A., and Ross, P. (2004). Artifical immune networks for robot control. In *Proc Fourth Int Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems*, volume 117, pages 151–154.

Ross, P. (2003). Notes on the RLA network implementation. Technical report, Napier University, Scotland.

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. The MIT Press.

Webb, A., Ross, P., Hart, E., and Lawson, A. (2003). Generating robot behaviours by evolving immune networks. Technical report, Napier University, Scotland.

ADDRESSING COMPLEXITY ISSUES IN A REAL-TIME PARTICLE FILTER FOR ROBOT LOCALIZATION

Dario Lodi Rizzini, Francesco Monica, Stefano Caselli
Dipartimento di Ingegneria dell'Informazione, Università di Parma
Viale G. P. Usberti, 181/A, 43100 Parma, Italy
{*rizzini, fmonica, caselli*}*@ce.unipr.it*

Monica Reggiani
Dipartimento di Informatica, Università di Verona
Ca' Vignal 2, Strada Le Grazie 15, 37134 Verona, Italy
reggiani@metropolis.sci.univr.it

Keywords: Robot Localization, Real-Time, Particle Filters.

Abstract: Exploiting a particle filter for robot localization requires expensive filter computations to be performed at the rate of incoming sensor data. These high computational requirements prevent exploitation of advanced localization techniques in many robot navigation settings. The Real-Time Particle Filter (RTPF) provides a tradeoff between sensor management and filter performance by adopting a mixture representation for the set of samples. In this paper, we propose two main improvements in the design of a RTPF for robot localization. First, we describe a novel solution for computing mixture parameters relying on the notion of effective sample size. Second, we illustrate a library for RTPF design based on generic programming and providing both flexibility in the customization of RTPF modules and efficiency in filter computation. In the paper, we also report results comparing the localization performance of the proposed extension and of the original RTPF algorithm.

1 INTRODUCTION

Robot localization is the problem of estimating robot coordinates with respect to an external reference frame. In the common formulation of the localization problem, the robot is given a map of its environment, and to localize itself relative to this map it needs to consult its sensor data. A particularly effective approach to solve this problem is the *probabilistic* one, due to the uncertainty affecting sensor data and movement execution. For this reason, bayesian filtering has become the prevailing paradigm in recent works on localization (Elinas and Little, 2005; Sridharan et al., 2005).

A stochastic estimator provides a result expressed in the form of a *probability density function* (PDF) represented like a continuous function by Gaussian filters (Kalman Filter, EKF) (Leonard and Durrant-Whyte, 1991; Arras et al., 2002) or a discrete decomposition of the state posterior by nonparametric filters. The main nonparametric algorithm is called *Particle Filter* (Fox et al., 1999) and relies on *importance sampling* (Doucet et al., 2001). With importance sampling, the probability density of the robot pose is approximated by a set of samples drawn from a proposal distribution, and an importance weight measures the distance of each sample from the correct estimation.

The nonparametric approach has the advantage of providing a better approximation of the posterior when a parametric model does not exist or changes during iteration, e.g. in initialization or when environment symmetries determine a multi-modal PDF. Even if techniques like *Multi-Hypothesis Tracking* (Arras et al., 2002) attempt to manage multi-modal distributions, particle filters are more efficient and can represent all kinds of PDFs, including uniform distributions. Moreover, particle filters avoid linearizations that can lead to poor performance and divergence of the filter for highly nonlinear problems.

Unfortunately, particle filters suffer from computational complexity due to the large number of discrete samples of the posterior: for each sample a pose update, a correction and a resample step are performed. Since localization can be performed slowly with respect to the usual movement and tasks of the robot, it would be conceivable to perform localization over a large time interval. Therefore, there have been attempts to adapt the number of samples (Fox, 2003). However, during an excessive time interval uncertainty increases and many useful observations are dropped; a proper interval to complete a particle filter iteration should be approximately equal to the rate of incoming data. A trade-off must therefore be reached between time constraints imposed by the need of collecting sensor data incoming with a given rate and the number of samples determining localization performance.

The *Real-Time Particle Filter* (RTPF) (Kwok et al., 2004) is a variant of a standard particle filter addressing this problem. This algorithm relies on the decomposition of the posterior by partitioning the set of samples in smaller subsets computed taking into account the sensor data received in different time intervals. By choosing the proper size for partitions, a particle filter iteration can be performed aligned with the sensor acquisition cycle.

While RTPF represents a remarkable step toward a viable particle filter-based localizer, there are a few issues to be addressed in developing an effective implementation. RTPF convergence is prone to some numerical instability in the computation of important parameters of the algorithm, namely the coefficients of the mixture constituting the posterior. Furthermore, even adopting RTPF as the basic architecture, the design of a flexible and customizable particle filter remains a challenging task. For example, life cycle of samples extends beyond a single iteration and covers an estimation windows in which mixture posterior computation is completed. This extended life cycle of samples impacts over software design. Moreover, RTPF addresses observations management and derived constraints. A good implementation should be adaptable to a variety of sensors.

This paper proposes improvements in both the algorithmic solution and the implementation of RTPF. In section 2, a novel approach in the computation of mixture weights based on the effective number of samples is proposed. This approach simplifies RTPF and tries to avoid spurious numeric convergence of gradient descent methods. In section 3, a localization library implementing a highly configurable particle filter localizer is described. The library takes care of efficient life cycle of samples and control data, which is different in RTPF and standard particle filter, and supports multiple motion and sensor models. This flexibility is achieved by applying *generic programming techniques* and a *policy* pattern. Moreover, differing from other particle filter implementations (e.g., CARMEN (Montemerlo et al., 2003)), the library is independent from specific control frameworks and toolkits. In section 4, simulation and experimental results are reported and compared with the original RTPF performance. These results confirm the effectiveness and viability of the proposed algorithm and its implementation.

2 ADDRESSING ALGORITHMIC COMPLEXITY

In particle filters, updating the particles used to represent the probability density function (potentially a large number) usually requires a time which is a multiple of the cycle of sensor information arrival. For example, range scanners return hundreds of values per scan, at a rate of several scans per second; vision sensors often require advanced algorithms to identify visual landmarks (Se et al., 2002; Sridharan et al., 2005) draining computational resources from the process of localization.

Naive approaches, yet often adopted, include discarding observations arriving during the update of the sample set, aggregating multiple observations into a single one, and halting the generation of new samples upon a new observation arrival (Kwok et al., 2004). These approaches can affect filter convergence, as either they loose valuable sensor information, or they result in inefficient choices in algorithm parameters.

An advanced approach dealing with such situations is the Real-Time Particle Filters (RTPF), proposed in (Kwok et al., 2004), which will be briefly described in the following.

2.1 Real-time Particle Filter

Assume that the system received k observations within an *estimation window*, i.e. the time required to update the particles. The key idea of the Real-Time Particle Filter is to distribute the samples in sets, each one associated with one of the k observations. The distribution representing the system state within an estimation window will be defined as a *mixture* of the k sample sets as shown in Figure 1. At the end of each

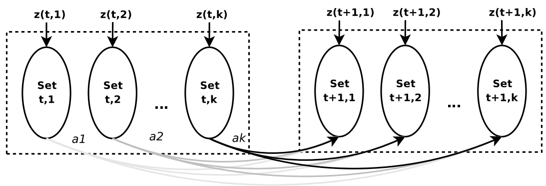


Figure 1: RTPF operation: samples are distributed in sets, associated with the observations. The distribution is a mixture of the sample sets based on weights α_i (shown as a_i in figure).

estimation window, the weights of the mixture belief are determined by RTPF based on the associated observations in order to minimize the approximation error relative to the optimal filter process. The *optimal belief* could be obtained with enough computational resource by computing the whole set of samples for each observation. Formally:

$$Bel_{opt}(x_{t_k}) \propto \int \cdots \int \prod_{i=1}^{k} p(z_{t_i}|x_{t_i}) \cdot p(x_{t_i}|x_{t_{i-1}}, u_{t_{i-1}}) \cdot Bel(x_{t_0}) dx_{t_0} \cdots dx_{t_{k-1}} \quad (1)$$

where $Bel(x_{t_0})$ is the belief generated in the previous estimation window, and z_{t_i}, u_{t_i}, x_{t_i} are, respectively, the observation, the control information, and the state for the $i-th$ interval.

Within the RTPF framework, the *belief* for the $i-th$ set can be expressed, similarly, as:

$$Bel_i(x_{t_k}) \propto \int \dots \int p(z_{t_i}|x_{t_i}) \cdot \prod_{j=1}^{k} p(x_{t_j}|x_{t_{j-1}}, u_{t_{j-1}}) \cdot Bel(x_{t_0}) dx_{t_0} \dots dx_{t_{k-1}} \quad (2)$$

containing only observation-free trajectories, since the only feedback is based on the observation z_{t_i}, sensor data available at time t_i.

The weighted sum of the k believes belonging to an estimation windows results in an approximation of the optimal belief:

$$Bel_{mix}(x_{t_k}|\alpha) \propto \sum_{i=1}^{k} \alpha_i Bel_i(x_{t_k}) \quad (3)$$

An open problem is how to define the optimal mixture weights minimizing the difference between the $Bel_{opt}(x_{t_k})$ and $Bel_{mix}(x_{t_k}|\alpha)$. In (Kwok et al., 2004), the authors propose to minimize their Kullback-Leibler distance (KLD). This measure of the difference between probability distributions is largely used in information theory (Cover and Thomas, 1991) and can be expressed as:

$$J(\alpha) = \int Bel_{mix}(x_{t_k}|\alpha) \log \frac{Bel_{mix}(x_{t_k}|\alpha)}{Bel_{opt}(x_{t_k})} dx_{t_k} \quad (4)$$

To optimize the weights of mixture approximation, a gradient descent method is proposed in (Kwok et al., 2004). Since gradient computation is not possible without knowing the optimal belief, which requires the integration of all observations, the gradient is obtained by Monte Carlo approximation: believes Bel_i share the same trajectories over the estimation windows, so we can use the weights to evaluate both Bel_i (each weight corresponds to an observation) and Bel_{opt} (the weight of a trajectory is the product of the weights associated to this trajectory in each partition). Hence, the gradient is given by the following formula:

$$\frac{\partial J}{\partial \alpha_i} \simeq 1 + Bel_i \log \frac{\sum_{i=1}^{k} \alpha_i Bel_i}{Bel_{opt}} \quad (5)$$

where Bel_i is substituted by the sum of the weights of partition set $i-th$ and Bel_{opt} by the sum of the weights of each trajectory.

Unfortunately, (5) suffers from a *bias problem*, which (Kwok et al., 2004) solve by clustering samples and computing separately the contribution of each cluster to the gradient (5). In the next section, an alternative solution is proposed.

2.2 Alternative Computation of Mixture Weights

This section proposes an alternative criterion to compute the values of the weights for the mixture belief. Instead of trying to reduce the Kullback-Leibler divergence, our approach focuses on evaluating the believes by synthetic values depending on the sample weights of each partition. The concrete approximation given by (Kwok et al., 2004) for the gradient of KL-divergence is a function of weights.

Real-time particle filter prior distribution is the result of two main steps: resampling of samples and propagation of trajectories along the estimation window. The effect of resampling is the concentration of previous estimation window samples in a unique distribution carrying information from each observation. Conversely, the trajectories update given by odometry and observation spreads the particles on partition sets.

Our attempt is to build synthetic values for each element of the resampled distribution and of the partition trajectory; this could be done using weights. Let w_{ij} be the weight of the $i-th$ sample (or trajectory) of the $j-th$ partition set. Then the *weight partition matrix* is given by

$$W = \begin{bmatrix} w_{11} & \dots & w_{1k} \\ \dots & & \dots \\ w_{N_p 1} & \dots & w_{N_p k} \end{bmatrix} \quad (6)$$

The weights on a row of this matrix trace the history of a trajectory on the estimation window; a group of values along a column depicts a partition handling sensor data in a given time. Resampling and trajectory propagation steps can be shaped using matrix W and mixture weights α.

- *Resampling*. The effect of resampling is the concentration of each trajectory in a unique sample whose weight is the weighted mean of the weights of the trajectory. In formula, the vector of trajectory weights is given by $t = W \cdot \alpha$.

- *Propagation*. Projecting a sample along a trajectory is equivalent to the computation of the weight of the sample (i.e., the posterior) for each set given the proper sensor information. Again, matrix W gives an estimation of the weight. Trajectories projection can thus be done with a simple matrix product

$$\hat{\alpha} = W^T \cdot t = W^T\, W \cdot \alpha \tag{7}$$

Vector $\hat{\alpha}$ is a measure of the relative amount of importance of each partition set after resampling and propagation depending on the choice of coefficient α. Hence, $\hat{\alpha}$ is the new coefficient vector for the new mixture of believes.

Some remarks can be made about the matrix $V = W^T\, W$ in (7). First, since we assume $w_{ij} > 0$, V is a symmetric and positive definite matrix. Moreover, each element j on the main diagonal is the inverse of the effective sample size (see (Liu, 1996)) of set j

$$n_{eff_j} = \frac{1}{\sum_{i=1}^{N_p} w_{ij}^2} \tag{8}$$

The effective sample size is a measure of the efficiency of the importance sampling on each of the partition sets. Therefore, the off-diagonal elements of V correspond to a sort of importance covariances among two partition sets. Thus we will refer to this matrix as *weights matrix*.

Hence, a criterion to compute the mixture weights consists of achieving a balance in the mixture forcing $\hat{\alpha}$ in (7) to be equal to α except for scale. The vector is thus obtained by searching an eigenvector of matrix V

$$V\,\alpha = \lambda\, I\, \alpha \tag{9}$$

The eigenvector can be computed using the power method or the inverse power method. This criterion can be interpreted as an effort to balance the effective number of samples keeping the proportion among different partition sets.

3 ADDRESSING SOFTWARE COMPLEXITY

While the choice of the algorithm is a key step, integration of a localization subsystem in a real mobile robot requires a number of practical issues and tradeoffs to be addressed. Real-time execution is the result of different aspects like communication and integration of the localizer with the robot control architecture, careful analysis in object creation/destruction cycles, and tradeoffs between abstraction level management and efficiency.

This section describes a library designed to efficiently support the implementation of particle filter localization algorithms, and specifically of RTPF. The library aims at providing an efficient yet open infrastructure allowing users to take advantage of the provided genericity to integrate their own algorithms. The library has been designed to be easily exploited in different control systems for autonomous mobile robots. In a functional layer, or controller, with the basic computational threads for robot action and perception, the localization task can be simply configured as a computational demanding, low priority thread.

3.1 Design of the Library

Advanced localization algorithms like RTPF address restrictions on real-time execution of localization due to limited computational resources. However, strictly speaking, real-time execution relies on the scheduling and communication capabilities of the robot control system which hosts the localizer. A localization subsystem should therefore be properly integrated in the control architecture. Nonetheless, localizer independence from the underlying low level control layer is a highly desirable property for a localization library. The adaptable aspects of the localization library are the data format and the models of the information provided by the physical devices of a mobile robots.

The functional analysis of the localization problem led to the identification of four main components: the localizer, the dynamic model of the system, the sensor data model, and the map. Each of these components is implemented by a class, which provides a general interface to handle prediction, correction and resampling phases of particle filters. However, there are different ways of modelling details like sensor and motion uncertainty, data formats of system state, control commands and observations, or implementation-specific like map storage and access. In our library, classes `Localizer`, `SystemModel`, `SensorModel` and `LocalizeMap` consist of a general interface which can be adapted.

The strategy pattern (Gamma et al., 1995) is the well-known design solution for decoupling algorithms from system architecture: with this pattern, algorithms can be modified and integrated in the application without modifying the internal code. Thus, using external abstract strategy classes for each changeable aspect of the localization subsystem, adaptation to the robotic platform hosting the localizer can be obtained. While this implementation avoids the hardwiring of user's choices inside the localizer code, it causes inefficiency due to the abstraction levels introduced to support the polymorphic behavior. Furthermore, the strategy pattern cannot handle changes of data types lacking an abstract base class; i.e., observation or control command types given by a generic robot control architecture cannot be used directly. These remarks, together with the observation that choices are immutable at runtime, suggested the use of *static polymorphism*. In the current library implementation, static polymorphism is effectively guaranteed by a *generic programming* approach, and in particular with *policy and policy classes* (Alexandrescu, 2001). This programming technique supports developer's choices at compile time, together with type checking and code optimization, by using templates.

```
template< State ,
          SensorData ,
          Control ,
          SampleCounter ,
          SampleManager ,
          Fusion >
class Localizer : public SampleManager<State>
{
  Fusion<SensorData> fusion_;

public:
  template< ... >     // StateConverter
  Localizer(Pdf &pdf,
            StateConverter<...> &converter);

  ~Localizer();

  template< ... >     // SystemModel Params
  void update(Control &u,
              SystemModel<...> &sys);

  template< ... >     // SensorModel Params
  void update(SensorData &z,
              SensorModel<...> &sen);
};
```

Listing 1: The `Localizer` class.

The main component of the library, the class `Localizer`, exemplifies how policies allow management of different aspects of localization problem. Listing 1 shows a simplified interface of the `Localizer` class, including only two `update()` methods. Note the template parameters of the class: there are both data types (`State`, `SensorData` and `Control`) and policies related to particle filter execution. Methods `update()` allow the execution of prediction and correction phases by using generic sensor and motion models, `SensorModel` and `SystemModel`, which are fully customizable interface classes with their own polices too.

Policies in `Localizer` provide the required flexibility in RTPF implementation. The library supports different versions of particle filters, and template parameters determine the actual algorithm implemented in the system. `SampleCounter` allows choosing the number of samples, that can be fixed or adaptable, e.g. with KL-distance (Fox, 2003). `SampleManager` is the class implementing sample management and creation/destruction of data involved in computation. This class plays an important role, since RTPF determines a complex life cycle for particles, as shown in figure 2. While in standard particle filters samples and control commands survive only during a single iteration (prediction, correction, resampling), RTPF needs the storage of data over the period of an estimation window.

4 RESULTS

In this section, we describe RTPF performance evaluation both in a simulated environment and using experimental data collected by navigating a robot in a known environment. One of the purposes of this evaluation is the comparison of the two RTPF versions differing in their method for computing mixture weights: the original method based on steepest descent and the method described in this paper and relying on the eigenvalues of the weights matrix. Simulations allow comparison of the two methods in a fully-controlled environment, whereas experiments show the actual effectiveness of the proposed technique.

4.1 Simulation

Several tests were performed in the simulated environment shown in figure 3, which corresponds to the main ground floor hallway in the Computer Engineering Department of the University of Parma. This environment allows verification of RTPF correctness while coping with several symmetric features, which may cause ambiguities in the choice of correct localization hypotheses. Real experiments with a mobile robot were carried out in the same environment and are described later in the paper: simulations have helped with the setup of experiments and viceversa. Two simulated paths exploited in simulation are also shown in figure 3. These paths, labeled as Path 1 and Path 2, correspond to lengths of approximately 7 *m* and 5 *m*.

In simulation, the map is stored as a grid with a given resolution (0.20 m) and is used both to create simulated observations and to compute importance weights in correction steps. Data provided to the localizer consist of a sequence of laser scans and measurements: scanned ranges are obtained by ray trac-

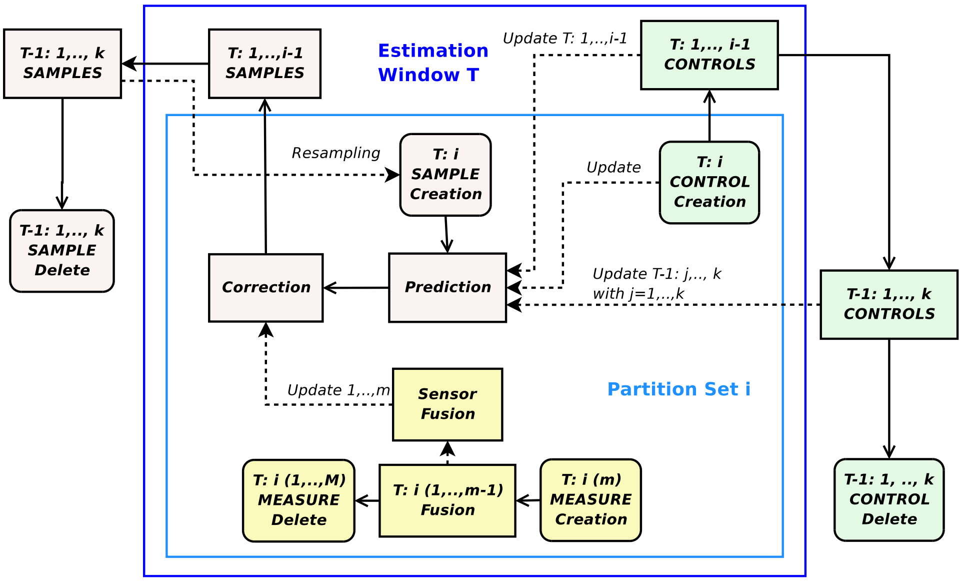


Figure 2: Life cycle for particle, measurement, and control objects within a single step in a real-time particle filter.

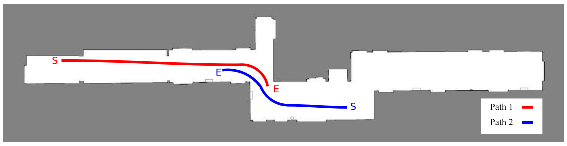


Figure 3: Hallway and simulated paths in the Computer Engineering Department, University of Parma (S=Start, E=End).

ing a beam on the discretized map. The measurement model is also based on ray tracing and follows standard beam models for laser scanners (Thrun et al., 2005). In our tests, we have used only three laser beams measuring distances to left, right and frontal obstacles; such poor sensor data stress the role of algorithm instead of sensor data. A gaussian additive noise is added to both range beams and robot movements representing environment inputs and robot state in simulation. The task of the robot is to achieve localization while moving in the map of figure 3 along different trajectories.

Localization algorithms investigated are RTPFs in the two versions: the original steepest descent-based one (RTPF-Grad) and the proposed one based on the effective number of samples (RTPF-Eig). During the tests the partition set size is 1000 samples.

A summary of simulation results is reported in figure 5. In the figure, curves show the localization error for the two algorithms at each iteration by considering convergence to the nearest hypothesis. For both curves, each value is obtained by averaging the distances of the estimated pose from the real pose over 10 trials where localization converged to the correct hypothesis. For both algorithms there were also a few simulation instances where localization did not converge to the correct hypothesis within the length of the path, although the correct hypothesis was the second best. These unsuccessful cases, mostly occurring on Path 2, were approximately 10% of all simulated localization trials. We did not verify whether the robot would recover its correct pose in the environment with further navigation.

On the average, the two versions of the RTPF-based localizer converged to some few hypotheses after three iterations: the common samples distribution is multi-modal, as shown in figure 4 where there are two local maxima. Hence, cluster search leads to few hypotheses with different weight (an example is shown in figure 4). In our tests a hypothesis close to the correct robot pose always exists, and when this hypothesis prevails there is a sudden change in localization error, as shown in figure 5. Convergence is helped by recognizable features, e.g. the shape of scans, but when the environment is symmetric it can be difficult to reach, especially with limited or noisy sensoriality. Of course, the mean error trend in figure 5 does not correspond to any of the simulated trials; rather, it is the result of averaging trials with quick convergence and trials where the correct hypothesis could only be recovered after many more iterations.

Figure 6 shows the percentage of simulation trials converging to the correct hypothesis (i.e. with localization error less than 1.5 m) at each iteration. Note that for 40 – 50% of simulation tests, convergence is reached after few iterations. In other simulations, the correct robot pose is recovered only after about 20 it-

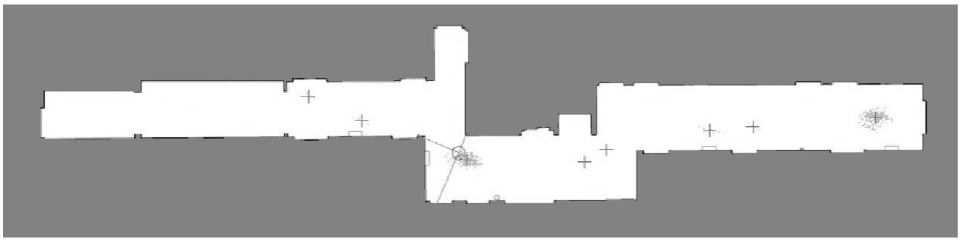


Figure 4: A typical distribution of samples condensed around two prevailing hypotheses (crosses mark hypotheses, the circle is centered in the robot position). When the wrong hypothesis has a higher weight, the localization error is huge.

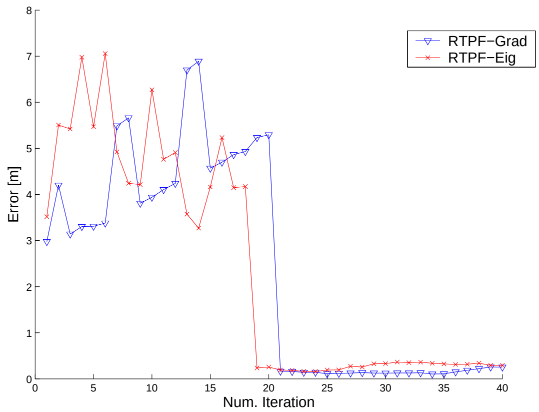


Figure 5: Performance of the two RTPF versions in the simulated environment. The *x*-axis represents the iterations of the algorithm. The *y*-axis shows the average error distance of the estimated pose from robot pose.

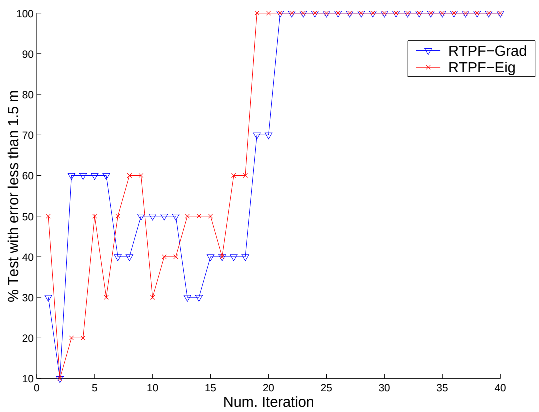


Figure 6: Percentage of simulation trials converged to the correct hypothesis, i.e. with localization error less than 1.5 m, during iterations for Map 1.

erations, i.e. after sensing map features that increase the weight of the correct samples.

Empirically, for the examined environment RTPF-Eig seems to exhibit a slightly faster convergence, on the average, to the correct hypothesis, but its average error after convergence appears somehow larger.

4.2 Experiments

Real experiments took place in the environment of figure 3 collecting data with a Nomad 200 mobile robot equipped with a Sick LMS 200 laser scanner. The robot moved along Path 1 for about 5 m, from the left end of the hallway in steps of about $15-20$ cm and reading three laser beams from each scan in the same way of the simulation tests. In the real environment localization was always successful, i.e. it always converged to the hypothesis closer to the actual pose in less than 10 iterations (remarkably faster than in simulation). Localization error after convergence was measured below 50 cm, comparable or better than in simulation.

To assess the consistency of the localizer's output on a larger set of experiments, we compared the robot pose computed by the localizer (using the RTPF-Eig algorithm) with the one provided by an independent localization methodology. To this purpose, some visual landmarks were placed in the environment and on the mobile robot, and a vision system exploiting the *ARToolKit* framework (Kato and Billinghurst, 1999) was exploited to triangualate the robot position based on these landmarks. The vision system provided an independent, coarse estimate of the robot pose at any step, and hence allowed to establish convergence of the RTPF-based localizer. The two localization estimates were computed concurrently at each location and stored by the robot.

Figure 7 shows the results of 10 tests of RTPF-Eig over about 20 iterations. These results confirm that RTPF-Eig achieves localization to the correct hypothesis very fast in most experiments. After convergence, the maximum distance between RTPF-based and vision based estimates is about 70 cm due to the compound error of the two systems.

5 CONCLUSION

Localizing a mobile robot remains a difficult task: although effective algorithms like particle filters exist,

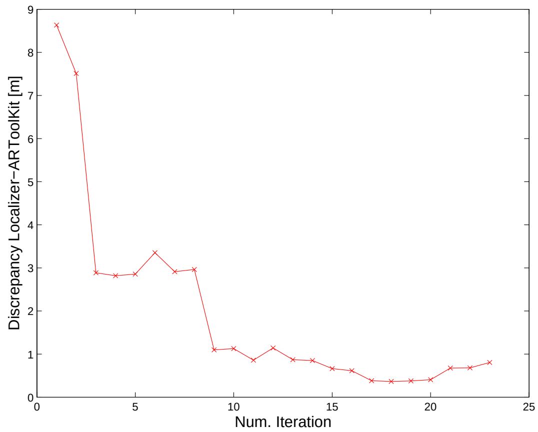


Figure 7: Discrepancy between RTPF-Eig and ARToolKit estimations using real data collected in the hallway of Map 1.

setting up a concrete localization system must deal with several architectural and implementation problems. One of these problems is the tradeoff between sensor data acquisition rate and computational load. Solutions like RTPF have been proposed to achieve this tradeoff, and are open to a number of improvements. Therefore, localization involves configuring customizable features both in the algorithm and in modules for sensor data, motion models, maps and other algorithmic details. A good generic implementation should provide components that the end user can adapt to his needs.

This paper has presented some improvements of the RTPF algorithm. In the proposed enhancement, the weight mixture of sample sets representing the posterior are computed so as to maximize the effective number of samples.

A novel RTPF implementation based on the enhanced algorithm has been developed. Experiments reported in the paper have shown this implementation to work both in simulated environments and in the real world. Assessing its relative merit with respect to the original RTPF proposal requires further investigation.

ACKNOWLEDGEMENTS

This work was supported by ”LARER – Laboratorio di Automazione della Regione Emilia-Romagna.” We thank Bruno Ferrarini for his help with the experimental assessment.

REFERENCES

Alexandrescu, A. (2001). *Modern C++ Design: Generic Programming and Design Pattern Applied*. Addison-Wesley.

Arras, K. O., Castellanos, H. F., and Siegwart, R. (2002). Feature-based multi-hypothesis localization and tracking for mobile robots using geometric constraints. *IEEE Int. Conf. on Robotics and Automation*, 2:1371–1377.

Cover, T. M. and Thomas, J. A. (1991). *Elements of Information Theory*. Wiley.

Doucet, A., de Freitas, J., and Gordon, N. (2001). *Sequential Monte Carlo Methods in Practice*. Springer.

Elinas, P. and Little, J. (2005). σMCL: Monte-carlo localization for mobile robots with stereo vision. *Proc. of Robotics: Science and Systems*.

Fox, D. (2003). Adapting the sample size in particle filters through KLD-sampling. *Int. J. of Robotics Research*, 22(12):985–1003.

Fox, D., Burgard, W., and Thrun, S. (1999). Monte Carlo Localization: Efficient position estimation for mobile robots. *Proc. of the National Conference on Artificial Intelligence*.

Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1995). *Design Patterns: elements of reusable object-oriented software*. Addison-Wesley.

Kato, H. and Billinghurst, M. (1999). Marker tracking and hmd calibration for a video-based augmented reality conferencing system. *Proc. of the Int. Workshop on Augmented Reality*.

Kwok, C., Fox, D., and Meilă, M. (2004). Real-time particle filters. *Proc. of the IEEE*, 92(3):469–484.

Leonard, J. J. and Durrant-Whyte, H. F. (1991). Mobile Robot Localization by Traking Geometric Beacons. *IEEE Int. Conf. on Robotics and Automation*.

Liu, J. (1996). Metropolized independent sampling with comparisons to rejection sampling and importance sampling. *Statistics and Computing*, 6(2):113–119.

Montemerlo, M., Roy, N., and Thrun, S. (2003). Perspectives on standardization in mobile robot programming: The Carnegie Mellon navigation (CARMEN) toolkit. *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*.

Se, S., Lowe, D., and Little, J. (2002). Mobile robot localization and mapping with uncertainty using scale-invariant visual landmark. *Int. J. of Robotics Research*, 21(8):735–758.

Sridharan, M., Kuhlmann, G., and Stone, P. (2005). Practical vision-based monte carlo localization on a legged robot. *IEEE Int. Conf. on Robotics and Automation*, pages 3366–3371.

Thrun, S., Burgard, W., and Fox, D. (2005). *Probabibilistic Robotics*. MIT Press, Cambridge, MA.

DYNAMIC REAL-TIME REDUCTION OF MAPPED FEATURES IN A 3D POINT CLOUD

Marko Reimer and Bernardo Wagner
Institute for Systems Engineering, Leibniz University Hanover, Hanover, Germany
{reimer, wagner}@rts.uni-hannover.de

Keywords: 3D perception, real-time processing, data reduction.

Abstract: This paper presents a method to reduce the data collected by a 3D laser range sensor. The complete point cloud consisting of several thousand points is hard to process on-line and in real-time on a robot. Similar to navigation tasks, the reduction of these points to a meaningful set is needed for further processes of object recognition. This method combines the data from a 3D laser sensor with an existing 2D map in order to reduce mapped feature points from the raw data. The main problem is the computational complexity of considering the different noise sources. The functionality of our approach is demonstrated by experiments for on-line reduction of the 3D data in indoor and outdoor environments.

1 INTRODUCTION

Environmental perception of mobile robots is still a challenging task. Many research groups work on improvements for on-line spatial perception. Almost all possible kinds of sensors are used to collect information. Actually, lots of research in robotics deals with the localization and mapping problem. Partly, this research is based on features. These features are mostly lines, planes or special edges. Few research is done in processing all the raw data in real-time on the robot itself. The main problem for doing it on-line is the less time for each process combined with less computation power for each step to finish. A further field of research, mainly in security context, focuses on object detection. On-line algorithms for object detection must be integrated into the control process to be of any advantage for robot task fulfillment and have to be finished in real-time time. One of the most important steps to speed up the data processing is to reduce the collected data to a meaningful set of data. An example for this is the mapping and localization problem. Therefore feature points in 3D are extracted, mapped into a 2D map and well known 2D algorithms are applied (Wulf et al., 2004b). The intention of this paper is to present a way to reduce the amount of data collected by a 3D range scanner for a following on-line object detection. This reduction uses environmental context knowledge to remove static features like walls from the 3D point cloud and keep only information not present in the given map. This may be used for reducing speed of an autonomous vehicle within range of dynamic obstacles. The reduceable feature points are found by two criteria. First, several sequent points must form a vertical line in 3D space. Second, this line must be matched to the environment map of the robot. As the computation power and the data storage for on-line reduction is limited, only actual data is used without keeping any previous information or calculating averages over time.

Another benefit reducing the raw data, additional to the gain in speed, is the better result of algorithms working on the reduced data set. A following object detection and segmentation algorithm may not be confused by mapped (known) features any more. For an example segmentation see (Talukder et al., 2002).

The main problem handled in this paper is matching the 3D-data in the presence of different kinds of noise. All used sensors and maps have different forms of noise requiring different handling.

The paper is organized in 5 parts. The remaining part of section 1 describes the form of the input data. Section 2 describes similar work. Section 3 shows the steps needed for the reduction and ways to handle uncertainties. After that, the experiments carried out on two robot platforms one indoor and one outdoor are shown in section 4. The conclusion is drawn in the 5. section.

The 3D data to be reduced is measured from a continuous 360 degree 3D laser range scanner. To be able to collect data while moving a movement compensation is applied. For details on the 3D sensor and the movement compensation see (Wulf and Wagner, 2003) or (Reimer et al., 2005). A 3D scan is organized as an ordered sequence of vertical 2D laser scans (see *Figure 1* left) in clockwise rotating directions. One set of 3D data always contains a 360 degree turn of the 3D scanner. Each 2D laser scan itself is an ordered list of points in 3D space. There is a fixed number of points per 2D scan which are ordered from bottom direction towards ceiling direction. The points are given in cartesian coordinates with the origin centered at the laser scanner.

Like a normal 2D map the map used for reduction represents the environment in a top view. The map consists of a set of lines defining surfaces detectable by a range sensor (see *Figure 1*). As the map is designed and also used for localization it only needs to include a 2D representation of the environment. This kind of map is often build from landmark offices or construction plans, which are not including 3D information. It might be build from aerial images given ground truth.

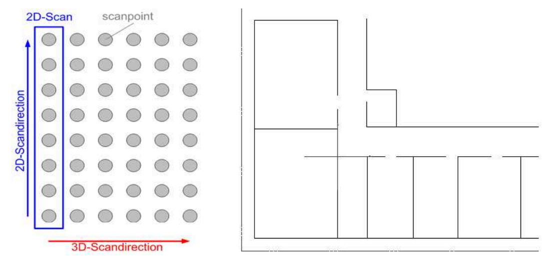


Figure 1: Left: Schematic of a 2D/3D Scan; Right: Sample map of our office environment.

2 RELATED WORK

At the first step, each 2D scan is handled seperately. Each 2D scan is segmented into lines. These line features extracted from range data are used by several successful systems as they are easy to compute and quite compact to handle. There are several papers how to construct line features. Two of the most commonly used ways are either Hough transformation (Pfister, 2003), and the split-and-merge approach (Borges, 2000). For an overview of line segmentation algorithms see (Nguyen et al., 2005).

The closest work to ours is the work extracting walls from 3D points collected by a static mounted but moved 2D scanner. For example (Thrun, 2003) collect these data with a helicopter, and Haehnel et al. do so with a moving robot (Haehnel et al., 2003). The main difference is the way 3D points are built. In their case the whole system is moved along a given path while the sensor is static on the system. They achieve a more dense point cloud and have a reduced angular error. They do not reduce the data on-line and in real-time but they post process the whole point cloud to extract matched walls using the EM-algorithm.

The proposed method should remove as much points as possible of visible matching feature parts. In contrast to the shown wall matching methods with a minimum 2D size, our method needs only a line to be detectable. The overall height of a line or plane is not constrainted. The removed features are not constricted by any minimum size requirement as it depends on the distance to the feature.

Another similar technique is the well researched segmentation of horizontal 2D range data into line features and matching them with a given 2D map. This is mostly done for localization like (Biber, 2005) and (Pfister, 2003). This is at least done on-line while the system operates. In the case of a vertical mounted 2D scanner vertical and not horizontal lines are measured, which differ in handling. Horizontal line matching has to deal with similar sensor noise and errors in the position estimation. But the horizontal neighborship of succeeding points can be used to account for the angular error in the localization. Compared to a horizontal neighbouring point a whole line in the vertical scan is effected by this error. The line cannot be corrected by weighting the single point with neighboring points into a line.

On the first sight, the well researched area of mapping is very similar to the described problem. There is an enormous number of papers about 'SLAM' which are at least partly dealing with the problem of mapping ((Thrun et al., 2005), (Thrun et al., 2004), (Nuechter et al., 2005), (Kuipers and Byun, 1990)). Especially the survey (Thrun, 2002) gives a good overview. The main difference of our approach to the mapping problem is that we assume to know our position and orientation up to a remaining error introduced by the localization. It is not our attempt to improve the quality of the localization in any kind as we want to detect and segment dynamic objects. The SLAM takes a big advantage of the possibility to sense a particular point in space twice or more often. As shown in (Thrun, 2002) they take an EM-algorithm to maximize the probability of a particular cell in a grid map. This is manly done using multiple views of the same cell over time. We want to reduce data immediately. These SLAM algorithms include methods for an estimated match of 2D point positions, mostly positions of feature points, with a given 2D map of the environment for localization (Wulf et al., 2004a).

Another similar topic is the iterative matching of two 3D scans or the iterative matching of features to 3D scans. There exist lots of matching techniques. The most popular technique is the iterative closest point (ICP) algorithm based on (Besl and McKay., 1992). As all these techniques try to give an optimal result in the overall match it takes too much computation time for our application.

3 REAL-TIME REDUCTION OF 3D DATA

Assuming a perfect localization, a perfect map and a noiseless sensor, the mathematical model for the reduction of 3D data to a 2D-map is straightforward. If the sensor is located at position α and the measured points form an exact vertical line at position β both positions can be transformed into positions within the map without any error. The position of the robot in the map is ι and the feature position is called δ. It is a comparison between the measured distance and the distance given by the map assuming the same direction.

$$d_{\alpha\beta} - d_{\iota\delta} = \zeta \tag{1}$$

If ζ is smaller than a threshold, the points are removed.

3.1 Laser Scanner Noise

The most common problem is the measurement noise of the laser range sensor. We show here how to care about the sensor noise without explizitly handling every point but combining the errors into a line representation. Range sensors mostly report data in a polar representation consisting of a distance measurement and a corresponding measurement angle. The distance measurement of commonly used laser scanners underlies a gaussian noise as shown in (Ye and Borenstein, 2002). This noise depends on parameter as temperature, incidence angle and color of the scanned surface. They showed that the influence of the surface properties is less than the influence of the incidence angle. As for example in (Pfister, 2002) the noise of a 2D laser scanner in the measured angle can be modeled as an additive zero-mean gaussian noise with a variance of σ^2. Whereas a common assumption σ^2 is less than one degree. The resulting measured distance vector $\overline{d_{meas}}$ is calculated by

$$\overline{d_{meas}} = (d_{true} + \varepsilon_{d_{sensor}}) \times \begin{bmatrix} \cos(\Theta + \varepsilon_\Theta) \\ \sin(\Theta + \varepsilon_\Theta) \end{bmatrix} \tag{2}$$

with Θ being the assumed angle of the measurement and ε being the corresponding error. For our reduction we do not consider single points and their noise but lines. These lines allow us to handle the sensor noise as an distance interval in the 2D ground distance d_ground_{meas}.

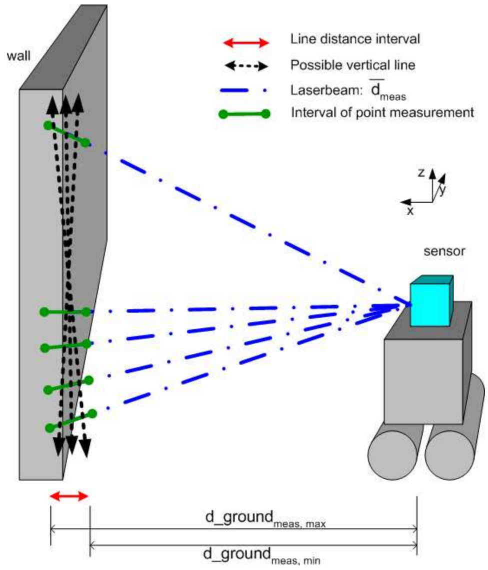


Figure 2: Sensor noise - points forming lines.

Figure 2 shows an example of the possible distributions of some wall points. As all points are taken on the front side of the wall they must be measured at the real distance added gaussian noise. The possible locations of the scanned points are shown by the intervals around the true front point at the wall (green solid lines). The line search algorithm is configured to some maximum distance a measured point may have towards the line. This defines the spread of points around the line. After fitting a line through the points the upper and the lower end point of the line define the maximum and minimum distance of the line to the sensor. There are several orientation possibilities for the found line, depending on the distribution of the noise (dotted orange lines). The angle of the found line against the z-axis is named γ. It's bound by a userdefined treshold η. This line forms the measured ground distance interval at the xy-plane (red line). This interval is given by:

$$d\_ground\_line_{meas,j} = \tag{3}$$

$$[min_i(d\_ground_{meas,i}); max_i(d\_ground_{meas,i})]$$

with i all points forming line j.

The mounting of the scanner does not need to be exactly vertical. So an error in the orientation of the lines is introduced. This angular difference is directly visible in the orientation of the measured line (orange line in *Figure 2*. All lines are not exactly vertical anymore but tilted. The tilting angle adds to the interval of possible line orientation angles (γ). This angular error results in different distances for the upper and lower ending of the line. The error in the distance is dependent on the length or height of the wall and the height of the scanner position. This error has a maximum influence, if both ends of the line strongly differ in the height towards the sensor. As an example a 2 degree mounting error results in a 10 cm distance error, if the line has a height of 3 m, if the scanner is mounted on the ground. This distance difference adds to the distance interval caused by the sensor distance noise.

$$d_{error}(\gamma) = height * \tan(\gamma) \quad (4)$$

As the tilting of the sensor is detectable during the mounting process it can be mechanically bound. To simplifiy calculations this maximum bound is used.

The minimum distance is given by:

$$d\_total_{meas,min} = d\_ground\_line_{meas,min} - d_{error}(\gamma) \quad (5)$$

Similar calculations can be applied for the maximum distance. Together these two values form the measured distance interval.

3.2 Localization Distance Noise

For calculating the expected distance the map and the output of the localization are used. These values are independent of the sensor model. As our algorithm relies on a given localization, we must take the error of the localization into account. A commonly used model for the localization error (Thrun et al., 2005) describes the located position as an ellipse with axis a and b, the orientation as an angle ω added a zero-mean gaussian noise ε_ω. There is no general bound on the localization error which can be applied to all different kinds of localization methods. So we assume these variances to be known for each position. As we want to reduce data very fast, we do not try to upgrade the position data by any means. We use the radii of the ellipses to calculate the possible difference in the ground distance introduced by the localization and build an interval of this size around the reference value taken from the map using the center of the ellipse.

$$r_{position} = \frac{b}{\sqrt{1 - a^2 b^2 \cos^2(\kappa)}} \quad (6)$$

κ is the direction of the scan. It is counted clockwise from the front direction of the scanner.

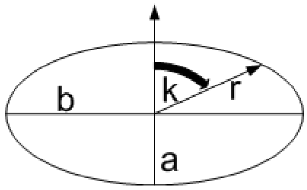


Figure 3: Localization position error.

The condition for the reduction is given by equation 7, when the angular error of the localization is not considered.

$$(d_{map} - r_{position} \leq d\_total_{meas,min} \leq d_{map} + r_{position})$$
$$or \quad (7)$$
$$(d_{map} - r_{position} \leq d\_total_{meas,max} \leq d_{map} + r_{position})$$

3.3 Localization Orientation Noise

The angular difference between the true orientation and the measured orientation has a worse effect than the position error. On a plane object or wall this angular error introduces a difference in the incidence angle ρ between the beam and the object. The resulting error in the distance is dependent on the distance and the actual incident angle.

$$d_{local-err} = d_{map} \times \left(1 - \frac{\sin(\rho)}{\sin(\rho + \varepsilon_\rho)}\right) \quad (8)$$

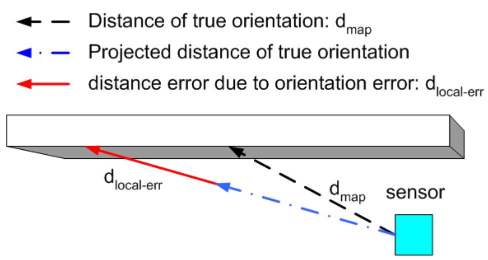


Figure 4: Incidenceangle with orientation error.

This error might be of enormous impact for small incident angles at large distances. If we add the difference $d_{local-err}$ (8) to the already calculated interval by Section 3.2 it is no longer possible to distinguish between objects significant in front of the wall and the wall itself. If we do not consider this error, the points forming this line are not reduced and remain for further processing steps. To account for this we extend the identified lines from the measured points. Our approach is to form a horizontal line from the already found vertical lines. The found vertical lines are projected into the ground plane. They form an interval or line in the xy-plane in contrast to the line in the xyz-system in *Figure 2*. All already matched wall lines are marked like the blue lines in *Figure 5*. The centers of neighboring matched ground lines are tested for forming a horizontal line in the ground plane using the same methode as in the first part. The next vertical line not already matched is tested to be a part of this horizontal line. Doing this the difference between the unmatched vertical line distance and the horizontal line is bounded by the point distance difference and independent of the distance error between the map and the measurement. If the vertical line interval fits into the horizontal line it is reduced as well. This method needs the feature to be matchable at least party. A matched segment must form a line to be extended.

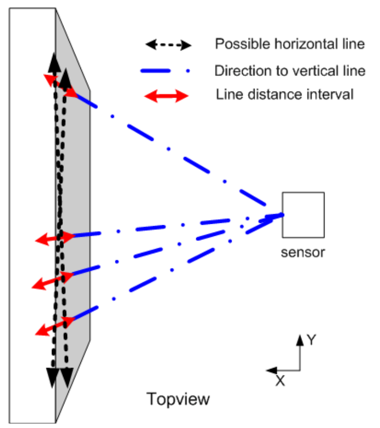


Figure 5: Vertical lines forming horizontal line.

3.4 Remaining Noise Sources

The error in the estimation of the measurement angle introduced by the servo drive cannot be handled in the same way as the orientation error of the localization. This error depends on the sampling frequency of the servo drive position and it is acceleration. The position of the servo drive is linear interpolated between two measurements. If the servo drive accelerates the change in turn velocity leads to an angular error dependent on the acceleration speed and sampling frequency of the position sensor. Both variables are controllable by the user and may be chosen to result in a negligible angular error. This might be reached, if the system is considered to be turning with a constant turn velocity after a startup phase.

The worst problem appears, if the angular difference between map and measured beam results in a hit on a different object or wall due to a corner. In this case the mathematical expression of the differences in the distance are not longer valid. Partly, if the incidence angle is small enough this case is caught by the horizontal line extension. The lines not caught remain as line in the point cloud.

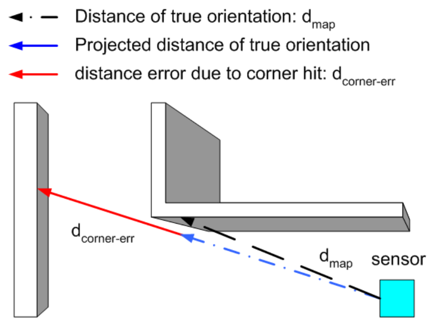


Figure 6: Error hitting a corner.

The mixed pixel problem is not solved by our approach. A measured line of points within a mixed pixel distance to the sensor cannot be matched to any feature. As the remaining vertical lines are orphants within their neighbourhood, they might be removed in a postprocessing step.

Finally, the map itself is probably not perfect but has some error. The error within the map is assumed to be within the same dimension as the noise of the range sensor. And therefore not handled explizit.

4 EXPERIMENTS

The proposed method has been implemented into the perception module of our 3D laser scanner. This 3D laser scanner is mounted on to two different chassis one for indoor use Section 4.1 and one for outdoor use Section 4.2. Both systems differ in various parameter but use the same software environment. For the indoor cases the map is created by hand using manually measured length. The outdoor map is based on material from the land registry office. The map error is in the dimension of centimeter. The same maps are used for localization and reduction at all times. The 3D laser scanner consists of a Pentium III Processor with 256 MB of RAM running a real-time linux (Xenomai). The time needed for the reduction on this processor is well below one second.

4.1 Indoor Experiments

For the indoor experiment we were driving on the floor with a speed of 0.5 m/s when the 3D scan has been taken. Beside the walls and doors there is one person and one cartoon standing on the floor roughly in front of the robot. The doors on the left side are closed, while the doors on the right are open. Through the right door an office is visible. This office is equiped with normal filled desks and cabinets at the wall. On the back side a fire extinguisher is located at the wall.

Figure 7 shows all found vertical lines and the result of the reduction. The dark red lines have been matched to the map while light green lines remain unmatched. At a first glance some lines within the wall appear not vertical. This is a problem of the line segmentation algorithm. As shown in (Borges, 2000) the start and endpoints of a line are not always chosen correctly. This may result in a wrong angle calculated for that line. These single lines (the corresponding points) will be removed after the matching step.

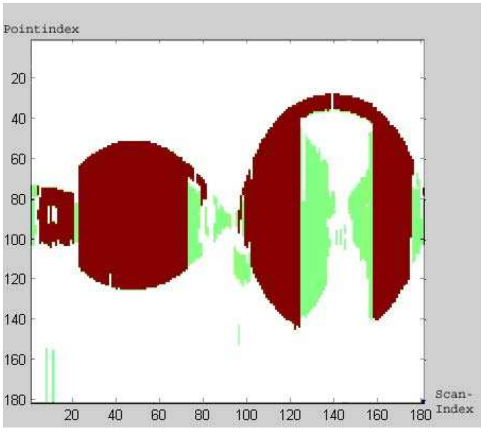


Figure 7: Reduced(red) and remaining lines(green) indoor.

On the left hand side all but one small vertical line on the wall has been removed. The door in the front part has not been modeled in the map and cannot be reduced at all. (see *Figure 1*) Also the person and the carton still remain as detected lines. The open door on the right remains like the lines on the cabinets within the office. The wall on the right side is totally removed even far in front at small incidence angles. This is due to the applied wall extension. The far wall on the left side cannot be removed because the person is totaly blocking the wall extension. On the back side on the left, between Scanindex 0-1, some lines which belong to a wall have not been removed. This is due to the error in the orientation while processing a corner in the map. At this direction no wall is found. The calculated beam hits the right side of the corner with the front wall, while the measured beam passes on the left side to the about 5m far away wall. The small line not removed around scanIndex 22 belongs to an other open door. In total from 12300 points forming vertical lines about 11700 points are removed. The complete 3D scan has 32761 points.

4.2 Outdoor Experiments

For the outdoor experiments we drove around a parking lot with several warehouses around. These buildings form the vertical lines to be matched. All existing features have been found as lines. As the distance to the buildings is big compared to the indoor distances much less points form vertical lines. All found vertical lines are shown in *Figure 8*. The removed vertical lines are dark red and remaining vertical lines are light green. These remaining lines are mostly located at the wirefence on the right side and the bush on the front side. Both are not represented in the map to be removed. The white area shows points not identified as vertical line and so not applictable for our reduction. All existing features have been removed while all dynamic objects are remaining. The computational complexity of further steps is reduced by more than 50 percent as most of the significant points are removed.

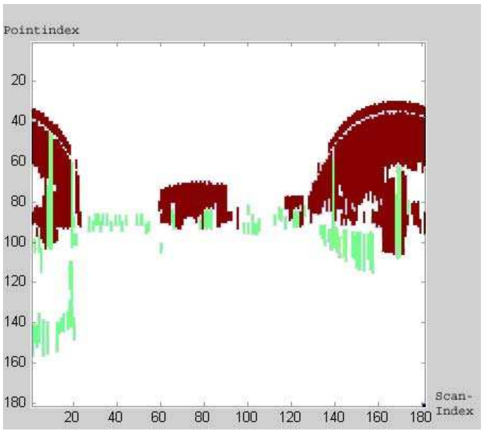


Figure 8: Reduced (dark red) and remaining lines (light green) outdoor.

5 CONCLUSION

In this paper we presented a method for real-time and on-line reduction of a 3D point cloud by given rough 2D-environmental knowledge. The method finds vertical line features in the 3D data and matches them to a given 2D map of these features. The problem for this procedure is the different noise introduced by several measurements. In order to speed up the calculation and simplifiy to noise handling we do not deal with every single point but build higher level line features. These line features are suitable to incorporate the noise added to the distance measurements. The line approach is used a second time on the line features itself, in order to handle the error in the orientation angle. The vertical lines are reduced to points on the ground which form a horizontal line feature to be independent of the relative orientation. A possible extension could be the integration of a non straight feature model as for example curved walls.

The proposed method is well suited for dynamic and complex environments as long as a simple 2D-map of matchable features is given and these features remain visible beside the dynamic objects. It improves a following object detection and recognition in computational speed and result quality.

The best improvement for the reduction quality may be gained by an improvement of the localization. Until the localization error is in the same dimension as the sensor noise, the use of a second or better sensor is not usefull. A possible extension might be to use the intensity value delivered by a range sensor together with the distance value. This intensity value is proportional to the strength of the reflection of the transmitted signal. The uncertainty interval of a measured distance could be calculated corresponding to the measured intensity of the distance. But the intensity is dependent on many factors not only on the incident angle and the other influences have to be canceled out before.

REFERENCES

Besl, P. J. and McKay., N. D. (1992). A method for registration of 3-d shapes. *IEEE Trans. on PAMI*, 14(2):239256.

Biber, P. (2005). 3d modeling of indoor environments for a robotic security guard. *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR05)*.

Borges, G.A; Aldon, M.-J. (2000). A split-and-merge segmentation algorithm for line extraction in 2d range images. *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, pages 441–444.

Haehnel, D., Triebel, R., Burgard, W., and Thrun, S. (2003). Map building with mobile robots in dynamic environments. *ICRA*.

Kuipers, B. and Byun, Y.-T. (1990). A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. Technical Report AI90-120.

Nguyen, V., Martinelli, A., Tomatis, N., and Siegwart, R. (2005). A comparison of line extraction algorithms using 2d laser rangefinder for indoor mobile robotics. *International Conference on Intelligent Robots and Systems, IROS2005,*.

Nuechter, A., Lingemann, K., Hertzberg, J., and Surmann, H. (2005). 6D SLAM with approximate data association. *ICRA*.

Pfister (2002). Weighted range sensor matching algorithms for mobile robot displacement estimation. *International Conference on robotics and automation*, pages 1667–1675.

Pfister, Samuel; S. Rourmelitis, J. B. (2003). Weighted line fitting algorithms for mobile robot map building and efficient data representation. *International conference on robotics and automation*, pages 1304–1312.

Reimer, M., Wulf, O., and Wagner, B. (2005). Continuos 360 degree real-time 3d laser scanner. *1. Range Imaging Day*.

Talukder, A., Manduchi, R., Rankin, A., , and Matthies, L. (2002). Fast and relaiable obstacle detection and segmentation for cross-country navigation. *Intelligent Vehicle Symposium, Versaille, France*.

Thrun, S. (2002). robotic mapping a survey. *Book chapter from "Exploring Artificial Intelligence in the New Millenium"*.

Thrun, S. (2003). Scan alignment and 3-d surface modeling with a helicopter platform. *IC on Field and service robotics*, 4th.

Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic Robotics*. The MIT Press.

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*.

Wulf, O., Arras, K. O., Christensen, H. I., and Wagner, B. (2004a). 2d mapping of cluttered indoor environments by means of 3d perception. *ICRA*.

Wulf, O., Brenneke, C., and Wagner, B. (2004b). Colored 2d maps for robot navigation with 3d sensor data. *IROS*.

Wulf, O. and Wagner, B. (2003). Fast 3d scanning methods for laser measurement systems. *International Conference on Control Systems and Computer Science (CSCS)*, 1.

Ye, C. and Borenstein, J. (2002). Characterization of a 2-d laser scanner for mobile robot obstacle negotiation. *International Conference on Robotics and Automation*.

LOW COST SENSING FOR AUTONOMOUS CAR DRIVING IN HIGHWAYS

André Gonçalves, André Godinho
Instituto Superior Técnico, Institute for Systems and Robotics, Lisbon, Portugal
andre_pvg@hotmail.com, agodes@hotmail.com

João Sequeira
Instituto Superior Técnico, Institute for Systems and Robotics, Lisbon, Portugal
jseq@isr.ist.utl.pt

Keywords: Autonomous car driving, Car-like robot, Behavior-based robot control, Occupancy grid.

Abstract: This paper presents a viability study on autonomous car driving in highways using low cost sensors for environment perception.
The solution is based on a simple behaviour-based architecture implementing the standard perception-to-action scheme. The perception of the surrounding environment is obtained through the construction of an occupancy grid based on the processing of data from a single video camera and a small number of ultrasound sensors. A finite automaton integrates a set of primitive behaviours defined after the typical human highway driving behaviors.
The system was successfully tested in both simulations and in a laboratory environment using a mobile robot to emulate the car-like vehicle. The robot was able to navigate in an autonomous and safe manner, performing trajectories similar to the ones carried out by human drivers. The paper includes results on the perception obtained in a real highway that support the claim that low cost sensing can be effective in this problem.

1 INTRODUCTION

Over the last decade there has been a steady decrease in the number of casualties resulting from automobile accidents in the European Union roads. Still, more than 40000 people die each year (see Figure 1) and hence additional efforts in autonomous car driving are socially relevant.

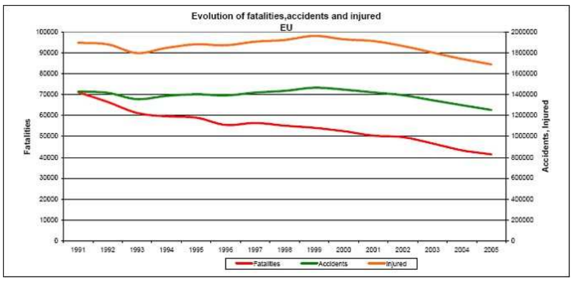


Figure 1: Eurostat data of sinistrality (source (EUROSTAT, 2006)).

Research on autonomous car driving systems shows that such systems may reduce drivers stress and fatigue resulting in a significant decrease in the number of accidents and fatalities.

Two main strategies are possible in the development of autonomous driving systems, one acting on the infrastructures, i.e., the highways, and the other on vehicles. The latter has been the one which has gathered higher acceptance throughout the scientific community. Developing a system that has the ability to navigate in the existing road network requires the capability of perceiving the surrounding environment. This paper describes an architecture for autonomous driving in highways based on low cost sensors, namely standard video cameras and ultrasound sensors.

Nowadays, the most advanced autonomous driving systems, such as the ones which took part in the Darpa Grand Challenge 2006, (Thrun et al., 2006; Whittaker, 2005) embody a large number of sensors with powerful sensing abilities. Despite proving capable of autonomous navigation even in unstructured environments such as deserts, these systems are unattractive from a commercial perspective.

A number of autonomous driving systems designed specifically for highway navigation have been described in the literature. As highways are a more structured and less dynamic environment, solutions can be developed with lower production and operational costs. At an academic level, (Dickmanns, 1999) developed systems that drove autonomously in the German Autobahn since 1985, culminating with a 1758Km trip between Munich and Denmark (95% of distance in autonomous driving). In 1995, Navlab 5 drove 2849 miles (98% of distance in autonomous driving) across the United States, (Jochen et al., 1995). In 1996, ARGO, (Broggi et al., 1999) drove 2000Km (94% of distance in autonomous driving) in Italy. At a commercial level, only recently have autonomous driving systems been introduced. Honda Accord ADAS (at a cost of USD 46.500), for example, is equipped with a radar and a camera, being capable of adapting the speed of the car to traffic conditions and keeping it in the center of the lane.

The system described in this paper, named HANS, is able to perform several tasks in the car driving domain, namely, following the road, keeping the car in the right lane, maintaining safe distances between vehicles, performing overtaking maneuvers when required, and avoiding obstacles. Without loosing generality, it is assumed that there are no cars driving faster than the HANS vehicle, meaning that no cars will appear from behind.

HANS uses a low resolution web camera located in the centre of the vehicle behind the rear-view mirror and a set of sixteen sonars. The camera's objective is to detect the road lanes and vehicles or objects that drive ahead. The sonar is used to detect distances to the vehicles/objects in the surroundings. HANS vehicle is an ATRV robot with a unicycle-like kinematic structure. A straightforward kinematics transformation is used such that the vehicle is controlled as if it has a car-like kinematics. Figure 2 shows the vehicle and the location of the sensors considered (the small spheric device in the front part of the robot is the webcam; other sensors shown are not used in this work) The experiments were conducted in a laboratory environment (in Figure 2) consisting on a section of a highway scaled down to reasonable laboratory dimensions. The total length is around 18 meters.

A behaviour-based architecture was used to model the human reactions when driving a vehicle. It is divided in three main blocks, common to a standard robot control architecture. The first, named Perception, generates the environment description. It is responsible for the sensor data acquisition and processing. An occupancy grid (see for instance (Elfes, 1989)) is used to represent the free space around the robot. The second block, named Behavior, is responsible for mapping perception into actuation. A finite automaton is used to choose from a set of different primitive behaviors defined after the a priori knowledge on typical human driving behaviors. The third block, Actuation, is responsible for sending the commands to the actuators of the robot.

Figure 2: The HANS vehicle (ATRV robot) in the test environment; note the obstacle ahead in the "road".

This paper is organized as follows. Sections 2 to 4 describe the perpection, behavioral and actuation blocks. Section 5 presents the experimental results. Section 6 summarizes the conclusions drawn from this project and points to future developments.

2 PERCEPTION

This section describes the sensor data acquisition and processing, together with the data fusion strategy. The environment representation method and the data fusion scheme were chosen to, in some sense, mimic those used by humans when driving a car.

2.1 Ultrasound Sensors

The space covered by the set of sonars is discretized into an occupancy grid, where the obstacles and free space around the robot are represented. Basic obstacle avoidance is achieved by computing the biggest convex polygon of free space in the area in front of the robot and controlling it such that it stays in this area.

The chosen occupancy grid, shown in Figure 3, divides each sonar cone into a number of zones, i.e., circular sectors, each being defined by its distance to the centre of the robot. Each raw measurement from a sonar is quantized to determine the zone of the grid it belongs to. Each zone, or cell, in the occupancy grid keeps record of the number of measurements that fell in it. The grid is updated by increasing the value in the cell where a measurement occured and decreasing the cell where the previous measurement occured. The zone with the highest number of measurements (votes in a sense) is considered as being occupied by an obstacle. This strategy has a filtering effect that reduces the influence of sonar reflexions.

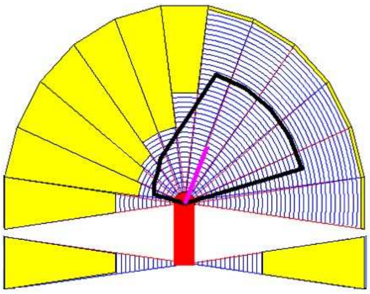


Figure 3: Occupancy grid for the sonar data.

The "three coins" algorithm, (Graham, 1972), is used to find the convex polygon which best suits the free area around the robot, using the information provided by the occupancy grid. This polygon represents a spatial interpretation of the environment, punishing the regions closer to the robot and rewarding the farther ones.

Sonars are also used to detect emergency stopping conditions. Whenever any of the measurements drops bellow a pre-specified threshold the vehicle stops. These thresholds are set differently for each sonar, being larger for the sonars which covering the front area of the vehicle.

2.2 Camera Sensor

The video information is continuously being acquired as low-resolution (320x240 pixels) images. The processing of video data detects (i) the side lines that bound the traffic lanes, (ii) the position and orientation of the robot relative to these lines, and (iii) the vehicles driving ahead and determining their lane and distance to the robot.

The key role of the imaging system is to extract the side lines that bound the lanes that are used for normal vehicle motion. Low visibility due to light reflections and occlusions due to the presence of other vehicles in the road are common problems in the detection of road lanes. In addition, during the overtaking maneuver, one of the lines may be partially or totally absent of the image.

Protection rails tend also to apear in images as lines parallel to road lanes. Assuming that road lines are painted in the usual white or yellow, colour segmentation or edge detection can be used for detection. The colour based detection is heavily influenced by the lighting conditions. Though an acceptable strategy in a fair range of situations, edge detection is more robust to lighting variations and hence this is the technique adopted in this study.

The algorithm developed starts by selecting the part of the image bellow the horizon line (a constant value as the camera is fixed on the top of the robot) converting to gray-scale. Edge detection is performed on the image using the Canny method, (Canny, 1986), which detects, suppresses and connects the edges. Figure 4 shows the result of an edge detection on the road lane built in a laboratory environment.

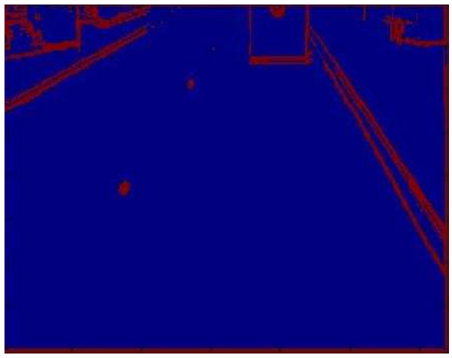


Figure 4: Edge detection using Canny's method.

The line detection consists of horizontal scannings starting from the middle of the road (using the line separating the lanes detected in the previous frame) in both directions until edge values bigger than a threshold are found. This method uses the estimates of the position of the lines obtained from previous frames.

To decide which of the edge points belong to the road lines, they are assembled in different groups. The difference between two consecutive points measured along the horizontal x axis is measured and if it is bigger than a threshold a new group is created.

After this clustering step, the line that best fits each group (slope and y-intersect) is computed by finding the direction of largest variance that matches the direction that corresponds to the smaller eigenvalue of the matrix containing the points of the group. Groups whose lines are very similar are considered to be part of the same line and are regrouped together.

The number of points in each group measures the confidence level for each line. The two lines (one for each side of the road) with the biggest confidence level are considered to be the left and right lines.

The final step in this selection consists in comparing the distance between the two computed lines with the real value (known a priori). If the error is above a predefined threshold, the line with a smaller confidence level is changed to one extrapolated at an adequate distance from the one with the bigger confidence level (recall that the width of the lanes is known a priori). The line separating the road lanes is then determined by finding the line equidistant to the detected side lines. Figure 5 shows the detection of the lines in the laboratory environment using the above procedure

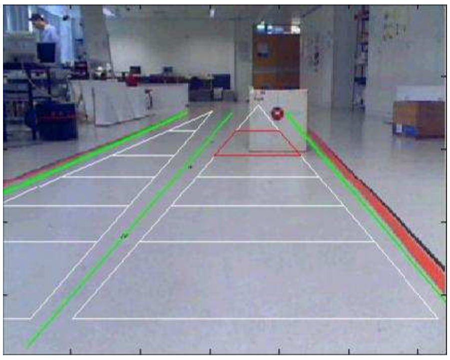


Figure 5: Detection of the road limiting lines.

Vehicles lying ahead in the road are detected through edge detection during the above line searching procedure. The underlying assumption is that any vehicle has significant edges in an image.

The road ahead the vehicle is divided in a fixed number of zones. In the image plane these correspond to the trapezoidal zones, shown in Figure 5, defined after the road boundary lines and two horizontal lines at a predefined distance from the robot. A vehicle is detected in one of the zones when the mean value of the edges and the percentage of the points inside the zone that represent edges lie above a predefined threshold. This value represents a confidence level that the imaging system detected a vehicle in one of the zones.

The two values, mean value and percentage value, represent a simple measure of the intensity and size of the obstacle edges. These are important to distinguish between false obstacles in an image. For instance, arrows painted on the road and the shadow of a bridge, commonly found in highways, could be detected as an obstacle or vehicle when doing the image processing. However, these objects are on the road surface while a vehicle has a volume. Therefore, an object is only considered as a vehicle if it is detected in a group of consecutive zones.

The robot's lateral position and orientation is determined from the position and slope of the road lines in the image.

2.3 Fusing Sonar an Image Data

The occupancy grid defined for the representation of the data acquired by the camera is also used to represent the obstacle information extracted from the ultrasound sensors data. Obstacles lying over a region of the occupancy grid contribute to the voting of the cells therein.

Both the camera and ultrasound sensors systems have associated confidence levels. For the camera this value is equal to the vehicle detection confidence level. The sonar confidence level depends on the time coherence of the detection, meaning that it rises when a sonar detects a vehicle in the same zone in consecutive iterations. The most voted zones in the left and right lanes are considered as the ones where a vehicle is detected.

There are areas around the robot which are not visible by the camera, but need to be monitored before or during some of the robot trajectories. For instance, consider the beginning of an overtaking maneuver, when a vehicle is detected in the right lane. The robot can only begin this movement if another vehicle is not travelling along its left side. Verifying this situation is done using the information on the position of the side lines that bound the road and sonar measurements. If any sonar measurement is smaller than the distance to the side line the system considers that a vehicle is moving on the left lane. This method is also used to check the position of a vehicle that is being overtaken, enabling to determine when the return maneuver can be initiated.

3 BEHAVIORS

Safe navigation in a highway consists in the sequential activation of a sequence of different behaviors, each controlling the vehicle in the situations arising in highway driving. These behaviors are such that the system mimics those of a human driver in similar conditions.

Under reasonably fair conditions, the decisions taken by a human driver are clearly identified and can be modeled using a finite state machine. The events triggering the transition between states are defined after the data perceived by the sensors. Figure 6 shows the main components of the decision mechanism used in this work (note the small number of states). Each state corresponds to a primitive behavior that can be commonly identified in human drivers.

Each of these behaviors indicates the desired lane of navigation. The first behavior, labeled *Normal*, is associated with the motion in the right lane, when the

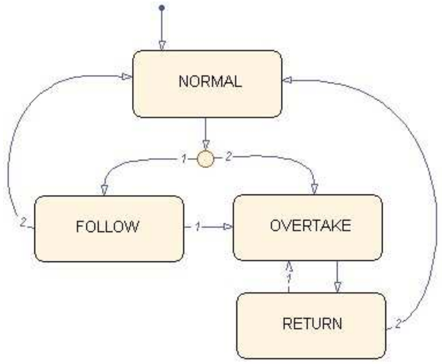


Figure 6: Finite automaton.

road is free. When a vehicle is moving in the right lane at a smaller speed than the HANS vehicle, the robot changes to one of two behaviors depending on the presence of another vehicle in the left lane. If the lane is clear, the behavior labeled *Overtake* is triggered and the robot initiates the overtaking maneuver. Otherwise the behavior labeled *Follow* is triggered and the robot reduces its speed to avoid a collision and follows behind him. The behavior labeled *Return* is associated with the return maneuver to the right lane that concludes the overtaking and is preceded by checking that the right lane is clear. The last behaviour, labeled *Emergency*, not shown in Figure 6, is activated when an emergency or unexpected situation is detected and implies an emergency stop, as this is a safety critical system.

4 ACTUATION

The actuation block receives position references from the currently active behavior. Assuming safe driving conditions, the trajectories commonly performed by vehicles moving in highways are fairly simple.

The control law considered computes the angle of the steering wheels from a lateral position error (the horizontal axis in the image plane), (Hong et al., 2001; Tsugawa, 1999; Coulaud et al., 2006). The rational under the choice of such a control law is that for overtaking, and under safe driving conditions, a human driver sets only a horizontal position reference while smoothly accelerates its vehicle aiming at generating a soft trajectory which allows him to change lane without colliding with the other car. This behavior leads the vehicle to the overtaking lane while increasing the linear velocity relative to the vehicle being overtaken such that, after a certain time, results in the overtaking maneuver completed. The reference for the HANS vehicle is thus a point lying a d look-ahead-distance, in one of the road lanes. Hard changes in direction caused by changes in the desired lane are further reduced through the use of a pre-filter that smooths the variations of the reference signal.

The control law is then,

$$\varphi = -A \tan^{-1}(Ke), \tag{1}$$

where φ is the angle of the directional wheels measured in the vehicle reference frame e is the lateral position error, A a constant parameter that is used to tune the maximum amplitude of φ, and K is a constant parameter used to tune the speed by which the control changes for a given error. Different values for the parameters K and A yield different driving behaviors. In a sense these two parameters allow the tuning of the macro-behavior of the autonomous driving system.

The primary concern when designing such a system must be safety, i.e., the system must operate such that it does not cause any traffic accident. Under reasonable assumptions (e.g., any vehicle circulating does not move on purpose such that is causes an accident), safety amounts to require system stability. The overall system is in fact a hybrid system, with the active behavior representing the discrete part of the hybrid state and the position and velocity of the vehicle the continuous one. From hybrid systems theory it is well known that continuous stability of individual states does not imply stability of the whole system, i.e., it is not enough to ensure that each behavior is properly designed to have the global system stable.

Lyapunov analysis can be used to demonstrate that (1) yields a stable system. Given the HANS vehicle kinematics (the usual reference frame conventions are adopted here)

$$\begin{aligned} \dot{x} &= v \sin(\theta)\cos(\varphi) \\ \dot{y} &= v \cos(\theta)\sin(\varphi) \\ \dot{\theta} &= \frac{v}{L}\sin(\varphi), \end{aligned} \tag{2}$$

where x, y, θ stand for the usual configuration variables, v is the linear velocity and L the distance between the vehicle's rear and front axis, and the lateral position and orientation errors, respectively, $e = x + d\sin(\theta)$, with d the look-ahead distance, and $\varepsilon = \theta - \theta_0$, after some straightforward manipulation yields

$$\begin{aligned} \dot{e} &= V \sin(\theta + \varphi) \\ \dot{\varepsilon} &= \tfrac{V}{L}\sin(\varphi) \end{aligned} \tag{3}$$

Substituting in the Lyapunov function candidate $V(e,\varepsilon) = 1/2\left(e^2 + \varepsilon^2\right)$ yields for $\dot{V}(e,\varepsilon)$

$$\dot{V} = e\dot{e} + \varepsilon\dot{\varepsilon} = ev\sin(\varepsilon + \varphi) + \varepsilon\frac{v}{L}\sin(\varphi) \qquad (4)$$

Given that V is positive definite and radially unbounded, to have the Lyapunov conditions for global assymptotic stability verified amounts to verify that $\dot{V}$ is negative definite. Table 5 shows the conditions to be verified for (4) to be negative.

Case	State variables	1st term	2nd term
1	$e > 0, \varepsilon > 0$	$\varphi < -\varepsilon$	$\varphi < 0$
2	$e > 0, \varepsilon < 0$	$\varphi < -\varepsilon$	$\varphi > 0$
3	$e ><, \varepsilon > 0$	$\varphi > -\varepsilon$	$\varphi < 0$
4	$e ><, \varepsilon < 0$	$\varphi > -\varepsilon$	$\varphi > 0$

(5)

Figure 7 shows the surfaces corresponding to the third and fourth columns in table 5.

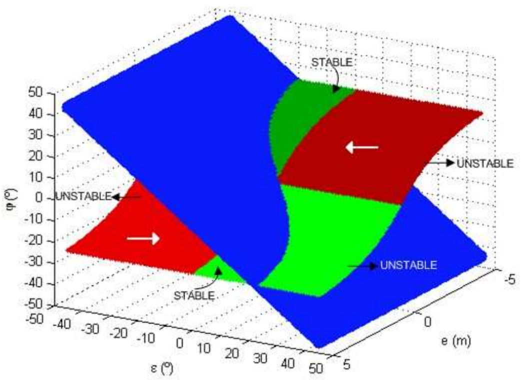


Figure 7: Lyapunov analysis.

Clearly, there are subspaces of the state space for which the control law (1) does not results in the candidate function being a Lyapunov function. However, the state trajectories when the HANS vehicle lies in these regions can be easily forced to move towards the stability regions. For instance, in cases 2 and 3 the vehicle has an orientation no adequate to the starting/ending of an overtaking maneuver. By simply controlling the heading of the vehicle before starting/ending the overtaking maneuver, the system state moves to the subspaces corresponding to situations 1/4, from which (1) yields a stable trajectory.

The analysis above does not demonstrate global assymptotic stability for the hybrid system that globally models the HANS system. However, assuming again reasonable conditions, the results in (Hespanha and Morse, 1999) can be used to claim that, from a practical point of view, the system is stable. The rational behind this claim is that driving in a highway does not require frequent switching between behaviors, which amounts to say that the dwell time between discrete state transitions tends to be large enough to allow each of the behaviors entering in a stationary phase before the next switching occurs. Using Figure 7, this means that once the system enters a stability region it stays there long enough to approach the equilibrium state $(e, \varepsilon, \dot{e}, \dot{\varepsilon}) = (0,0,0,0)$. Alternatively, global assymptotic stability can be claimed by using the generalized version of Lyapunov second method (see for instance (Smirnov, 2002)) which only requires that the last behavior triggered in any sequence of maneuvers is stable.

5 RESULTS

The system was tested in two different conditions. The imaging module was tested in real conditions, with the camera mounted in front, behind the car's windshield. Figure 8 shows the results.

Figure 8: Testing the imaging module in real conditions.

The testing of the system including the motion behaviors was conducted in a laboratory environment using the ATRV vehicle shown in Figure 2. This robot was controlled through the kinematics transformation, $u = v\cos(\varphi)$, $\omega = \frac{v}{L}\sin(\varphi)$.

In the first test the robot is moving in the right lane and, after detecting a car travelling in the same lane at a lower speed, initiates the overtaking maneuver. Figure 9 shows the resulting trajectory.

While travelling in the right lane, the *Normal* behavior is active. Once the car in front is detected the robot switches to the *Overtake* behavior and initiates the overtaking maneuver moving to the left lane. Since the velocity of the robot is greater than that of the car being overtaken, after a while the Perception block detects that the car is behind the robot and decision system switches to *Return*. When the centre of the right lane is reached, the active behavior returns to *Normal* completing one cycle of the decision finite state machine.

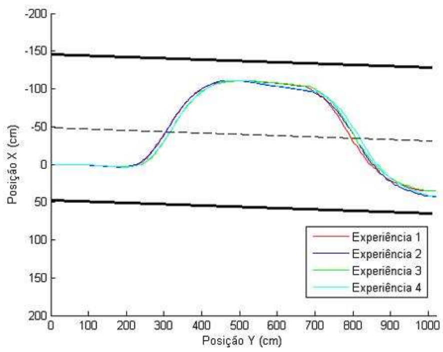


Figure 9: Overtaking a single vehicle.

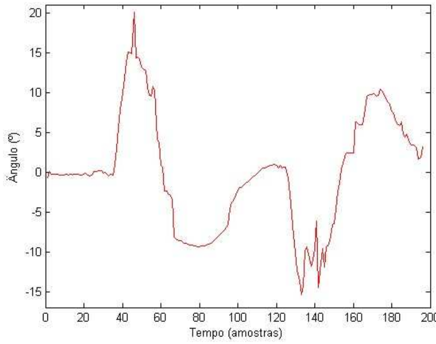


Figure 10: Steering wheels angle during overtaking.

Figure 10 shows the angle of the steering wheels. It reaches a maximum around 20° slightly after switching to *Overtake* ($t = 40$), making the robot turning left, towards the left lane. After reaching the maximum direction angle, the lateral error and directional angle starts decreasing.

Approximately at $t = 60$ the steering wheels angle is null, meaning that the position of the lookahead point is at center of the left lane, i.e., the robot is aligned with the axis of left lane. When φ drops to negative values the robot turns right aiming at reducing the orientation error and keeping the alignment with the axis of the left lane. The lane change is completed at approximately $t = 120$, when the steering angle stabilizes at 0, meaning the robot is moving in the centre of the left lane.

At $t = 125$ the system switches to the *Return* behavior. The trajectory towards the right lane is similar (tough inverse) to the one carried out previously from the right to left lane.

In both lane changes the steering wheel angle is noisier in the zones where the maximum values are reached ($t = 40$ and $t = 130$). At these moments the orientation of the robot is high and therefore one of the road lines (the one farther to the robot) has to be extrapolated because it is not visible in the camera image. This results in some oscillation in the line's position resulting in a noisier position error and steering wheel angle.

The second experiment aims at verifying the system's response when it is necessary to interrupt an ongoing maneuver to overtake a second vehicle. Figure 11 shows the resulting trajectory.

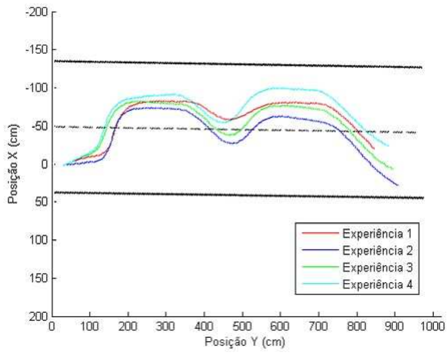


Figure 11: Double overtaking.

While circulating in the right lane, the robot detects a car in front and initiates the overtaking maneuver following the routine already described in the previous experiment. When switching to *Return* the robot does not detect a second car which is circulating ahead in the road, in the right lane, at a lower speed. This situation occurs if the car is outside the HANS' detection area but inside the area it needs to complete the return maneuver. If the car was inside the detection area, HANS would continue along the left lane until passing by the second car and then return to the right lane. In this situation, however, the car is only detected when the robot is approximately in the middle of the return phase (and of the road). Immediately, the decision system changes the behavior to *Overtake*, the robot initiates a new overtaking maneuver. After passing by the car, the robot returns to the right lane to conclude the double overtaking maneuver.

Figure 12 shows the evolution of the steering wheels angle. The switching from *Return* to *Overtake*) is visible around instant 125 through the drastic change in the control signal from -18 to 15 degrees. Still, the system performs a smooth trajectory.

6 CONCLUSION

Despite the low resolution of the chosen sensors, the perception block was able to generate a representation of the surrounding environment with which decisions could be made towards a safe autonomous navigation.

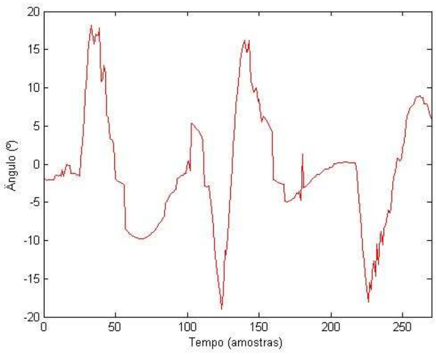


Figure 12: Steering wheels angle during the double overtaking.

Obviously, the use of additional sensors, such as more cameras and laser range finders, would give the system enhanced capabilities. Still, the experiments presented clearly demonstrate the viability of using low cost sensors for autonomous highway driving of automobile vehicles.

The sonar data processing uses crude filtering techniques, namely the occupancy grid and the voting system, to filter out data outliers such as those arising due to reflection of sonar beams. Similarly with image processing, supported in basic techniques that nonetheless where shown effective, with the line search based on the edge detection showing robustness to lighting variations.

The small number of behaviours considered was shown enough for the most common highway driving situations and exhibit a performance that seems to be comparable to human. Nevertheless, the decision system can cope easily with additional behaviors that might be found necessary in future developments. For instance, different sets of A, K parameters in (1) yield different behaviors for the system.

Future work includes the use of additional low cost cameras and the testing of alternative control laws, for instance including information on the velocity of the vehicles driving ahead.

ACKNOWLEDGEMENTS

This work was supported by Fundação para a Ciência e a Tecnologia (ISR/IST plurianual funding) through the POS_Conhecimento Program that includes FEDER funds.

REFERENCES

Broggi, A., Bertozzi, A. Fascioli, A., and Guarino, C. (1999). The Argo autonomous vehicles vision and control systems. *Int. Journal on Intelligent Control Systems (IJICS)*, 3(4).

Canny, J. (1986). A computational approach to edge detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8(6):679–698.

Coulaud, J., Campion, G., Bastin, G., and De Wan, M. (2006). Stability analysis of a vision-based control design for an autonomous mobile robot. *IEEE Trans. on Robotics and Automation*, 22(5):1062–1069. Universit Catholique de Louvain, Louvain.

Dickmanns, E. (1999). Computer vision and highway automation. *Journal of Vehicle System Dynamics*, 31(5-6):325–343.

Elfes, A. (1989). Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6):46–57.

EUROSTAT (Accessed 2006). http://epp.eurostat.ec.europa.eu.

Graham, R. (1972). An efficient algorithm for determining the convex hull of a finite planar set. *Information Processing Letters*, 1:132–133.

Hespanha, J. and Morse, A. (1999). Stability of switched systems with average dwell-time. In *Procs of 38th IEEE Conference on Decision and Control*, pages 2655–2660.

Hong, S., Choi, J., Jeong, Y., and Jeong, K. (2001). Lateral control of autonomous vehicle by yaw rate feedback. In *Procs. of the ISIE 2001, IEEE Int. Symp. on Industrial Electronics*, volume 3, pages 1472–1476. Pusan, South Korea, June 12-16.

Jochen, T., Pomerleau, D., B., K., and J., A. (1995). PANS - a portable navigation platform. In *Procs. of the IEEE Symposium on Intelligent Vehicles*. Detroit, Michigan, USA, September 25-26.

Smirnov, G. (2002). *Introduction to the Theory of Differential Inclusions*, volume 41 of *Graduate Studies in Mathematics*. American Mathematical Society.

Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M., Hoffmann, G., Lau, K., Oakley, C., Palatucci, M., Pratt, V., Stang, P., Strohband, S., Dupont, C., Jendrossek, L., Koelen, C., Markey, C., Rummel, C., van Niekerk, J., Jensen, E., Alessandrini, P., Bradski, G., Davies, B., Ettinger, S., Kaehler, A., Nefian, A., and Mahoney, P. (2006). Stanley: The robot that won the DARPA Grand Challenge. *Journal of Field Robotics*, 23(9):661–692.

Tsugawa, S. (1999). An overview on control algorithms for automated highway systems. In *Procs. of the 1999 IEEE/IEEJ/JSAI Int. Conf. on Intelligent Transportation Systems*, pages 234–239. Tokyo, Japan, October 5-8.

Whittaker, W. (2005). Red Team DARPA Grand Challenge 2005 technical paper.

REAL-TIME INTER- AND INTRA- CAMERA COLOR MODELING AND CALIBRATION FOR RESOURCE CONSTRAINED ROBOTIC PLATFORMS

Walter Nisticò and Matthias Hebbel
Robotics Research Institute (IRF), Universität Dortmund, Otto-Hahn-Str. 8, Dortmund, Germany
{*walter.nistico, matthias.hebbel*}@*uni-dortmund.de*

Keywords: Camera modeling, color correction, evolutionary algorithms, real time.

Abstract: This paper presents an approach to correct chromatic distortion within an image (vignetting) and to compensate for color response differences among similar cameras which equip a team of robots, based on Evolutionary Algorithms. Our black-box approach does not make assumptions concerning the physical/geometrical roots of the distortion, and the efficient implementation is suitable for real time applications on resource constrained platforms.

1 INTRODUCTION

Robots which base their perception exclusively on vision, without the help of active sensors such as laser scanners or sonars, have to deal with severe additional constraints compared to generic computer vision applications. The robots have to interact with a dynamic (and sometimes even competitive or hostile) environment and must be able to take decisions and react to unforeseen situations in a fraction of a second, thus the perception process has to run in real-time with precise time boundaries. In particular, autonomous robots are typically constrained even in terms of computational resources, due to limitations in power supply, size, and cost. A very popular approach especially for indoor environments is color-based image segmentation, and even though dynamic approaches have been demonstrated (for example, (Iocchi, 2007) (Schulz and Fox, 2004)), static color classification (Bruce et al., 2000) is still the most widely used solution, due to its efficiency and simplicity. In a group of homogeneous robots, which make use of color information in their vision system, it is important that all cameras produce similar images when observing the same scene, to avoid to have to individually calibrate the vision system of each robot, a procedure which is both time consuming and error prone. Unfortunately, even when the cameras are from the same model and produced from the same manufacturer, such assumption does not always hold (Röfer, 2004).

1.1 The Platform

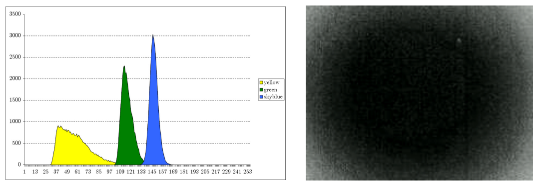
This work has been developed on the popular Sony Aibo ERS-7 robot (Sony Corporation, 2004), which is the only complete standard platform widely adopted for robotic applications to date. The robot is equipped with a 576MHz 64bit RISC CPU, 64MB of main memory, and a low-power CMOS camera sensor with a maximum resolution of 416×320 pixel. Images are affected by a ring-shaped dark blue cast on the corners, and different specimen of the same model tend to produce slightly different color responses to the same objects and scene. Our experiments have been conducted using the YUV color space which is natively provided by most common cameras, but the approach can be applied unaltered to the RGB color space. The Aibo production has been recently discontinued by Sony, but several new commercially available robotic kits are being introduced on the market, with similar characteristics in terms of size and power, often equipped with embedded RISC CPUs or PDAs and low quality compact flash cameras.

1.2 Related Work

Whenever object recognition is mostly based on color classification, the dark / colored cast on the corners of the images captured by the on board camera is a serious hindrance. Vignetting is a radial drop of image brightness caused by partial obstruction of light from the object space to image space, and is usually dependent on the lens aperture size (Nanda and Cutler, 2001) (Kang and Weiss, 2000). These approaches, as well as (Manders et al., 2004), treat the problem in terms of its physical origins due to geometrical defects in the optics, and are mostly focused on radiometric calibration, i.e. ensuring that the camera response to illumination after the calibration conforms to the principles of homogeneity and superposition. However, none of the proposed methods deals with chromatic distortion, as is the case of our reference platform, and other inexpensive low power CMOS sensors. Recently, a few papers have attempted to tackle such problem. These solutions share a similar approach to minimize the computational costs by using lookup tables to perform the correction in real time, while the expensive calibration of the correction tables is performed off-line. In (Xu, 2004) the author uses a model based on a parabolic lens geometry, solved through the use of an electric field approach. No quantitative analysis of the results is provided, but this technique has been successfully used in practice by one team of autonomous robots in the RoboCup Four-Legged League.[1] Another successful technique used in RoboCup has been presented in (Nisticò and Röfer, 2006), based on a purely black-box approach where a polynomial correction function is estimated from sample images using least square optimization techniques. Since this approach does not rely on assumptions concerning the physics of the optical system, we feel that it can be more effective in dealing with digital distortions such as saturation effects. Again no quantitative analysis has been presented, and both papers do not address the problem of inter-robot camera calibration, which has been treated in (Lam, 2004) with a simple linear transformation of the color components considered independently.

2 COLOR MODEL

The first step to understand the characteristics of this chromatic distortion, was to capture images of special cards that we printed with uniform colors, illuminating them with a light as uniform as possible, trying to avoid shadows and highlights.[2] Then we calculated the histograms of the three image spectra, with a number of bins equal to the number of possible values that each spectrum can assume, i.e. 256. Under these conditions, the histograms of such uniform images should be uni-modal and exhibit a very narrow distribution around the mode (in the ideal case, such distribution should have zero variance, i.e. all the pixels have exactly the same color) due only to random noise. Instead, it could be observed that the variance of the distribution is a function of the color itself; in case of the U channel, it appears very narrow for cold / bluish color cards, and very wide for warm / yellowish cards (Figure 1(a)). Consequently, we model the chromatic distortion d_i for a given spectrum i of a given color I as a function of I_i itself, which here we will call *brightness component* $\lambda_i(I_i)$.



(a) (b)

Figure 1: a) Histograms of the U color band for uniformly colored images: yellow, green and skyblue. Notice how the dispersion (due to the vignetting) increases inverse proportionally to the position of the mode. b) Brightness distribution of the U color band for a uniformly yellow colored image.

The distribution itself is not centered around the mode, but tends to concentrate mostly on one side of it. The reason for this becomes apparent by observing the spatial distribution of the error (cf. Figure 1(b)); the phenomenon itself is nothing but a ring shaped blue/dark cast, whose intensity increases proportionally to the distance from the center of the distortion (u_d, v_d), which lies approximately around the optical center of the image, the principal point. So, let $r = \sqrt{(x-u_d)^2 + (y-v_d)^2}$, then we define the *radial component* as $\rho_i(r(x,y))$. Putting together brightness and radial components, we obtain our distortion model:

$$d_i(I(x,y)) \propto \rho_i\left(r(x,y)\right) \cdot \lambda_i\left(I_i(x,y)\right) \quad (1)$$

Now, due to the difficulty to analytically derive $\rho_i, \lambda_i, \forall i \in \{Y,U,V\}$ about which little is known, we decided to use a black-box optimization approach. Both sets of functions are non-linear, and we chose to approximate them with polynomial functions

[1] RoboCup is an international joint project to promote AI, robotics, and related fields. *http://www.robocup.org/*

[2] However, this is not so critical, and the use of a professional diffuse illuminator is not necessary, as our approach can deal well with noise and disturbances (see Section 2.1).

(McLaurin series expansion).

$$\begin{aligned} \rho_i(r) &= \sum_{j=0}^{n} \varrho_{i,j} \cdot r^j \\ \lambda_i(I_i) &= \sum_{j=0}^{m} l_{i,j} \cdot I_i^j \end{aligned} \tag{2}$$

The unknown polynomial coefficients $\varrho_{i,j}$ and $l_{i,j}$ can be estimated, from a set of samples, using naturally inspired machine learning techniques. So, the final vignetting correction function is as follows:

$$I_i'(x,y) = I_i(x,y) - \rho_i(r(x,y)) \cdot \lambda_i(I_i(x,y)) \tag{3}$$

where $I_i'(x,y)$ is the corrected value of the spectrum i of the given pixel.

2.1 Reference Color Estimation

To be able to model our error functions, we must first estimate how the pictures should look like, if they were not affected by any chromatic distortion. Since most of the image area exhibits little to no distortion, we define as *reference color* of a single image the most frequent color appearing in its spectra. So we calculate the histograms of the 3 color spectra (number of bins = number of color levels = 256) and we find the modes $\bar{r}_i^Y, \bar{r}_i^U, \bar{r}_i^V$, which represent the reference color for image i. However, a single image can be affected by other sources of noise, such as a temporary change of light intensity, strobing (aliasing between the camera shutter speed and the light source power frequency, which affects fluorescent lights) or shadows: consequently, the histograms' modes might have temporary fluctuations, or exhibit multiple nearby modes. To make our system more robust toward this kind of noise, we collect multiple pictures of each colored cards in a log file, and we partition the images therein contained into image classes[3], where a class represents a certain color card (e.g. yellow, orange, green, blue) at a certain light intensity and camera settings. For each image class j we want to have a single reference color $\bar{R}_j^Y, \bar{R}_j^U, \bar{R}_j^V$: of course this could be obtained by averaging the references from all the images which belong to the class, but this would still be affected by outliers. Instead, we track the current reference of a given class using a simple first order linear Kalman filter (Welch and Bishop, 2003):

- For each image i in the log, a reference value is estimated for the 3 spectra $\bar{r}_i^Y, \bar{r}_i^U, \bar{r}_i^V$, as the modal value of the corresponding histogram
- Constant value process model: the predicted reference color $\hat{R}_j^Y, \hat{R}_j^U, \hat{R}_j^V$ for class j at the following step (image i) remains the same as the one after the measurement update of image $i-1$
- We use the output of the Kalman filter to perform the partitioning into color classes on the fly; a new class is generated when:

  $$\exists c \in \{Y, U, V\} : \left| \hat{R}_j^c - \bar{r}_i^c \right| > \vartheta \tag{4}$$

  where ϑ is a confidence threshold; so if at least one of the spectra in the reference of the current image differs too much from the expected reference for the current class, then we need to create a new color class. A new class reference is initialized to the value extracted from the current image.
- Measurement model: we update the running class reference $\hat{R}_j^Y, \hat{R}_j^U, \hat{R}_j^V$ using the reference extracted from the current image

[3]With *image class* or *color class* here we will refer to a color card captured under certain lighting settings, i.e. the same color card can be represented by different color classes, like blue-dark and blue-light.

2.2 Inter-camera Model

In the general case, it is possible that to completely correct the difference in color response between two cameras it is necessary to rotate the color space of one camera to align with the other. This however, is not feasible for a real time implementation for robotic applications. To capture the dependencies between the 3 color components and the distance from the image center, we would need a $256^3 \cdot \rho_{max}$ look-up table, which would have a size of over 2GB even for our low resolution camera. Otherwise, we could perform the rotation with the multiplication of a 3×3 matrix by our color vector; since such costly operation would have to be performed for every pixel, it would slow down the image processing too much.

In (Lam, 2004) the author suggests a simple linear transformation of the 3 color components of the camera to be calibrated, treating them independently from each other; such an approach is used in image processing programs to perform adjustments in the color temperature of a picture. Since we are going to use an evolutionary approach to the optimization process, we have decided to give more freedom to the inter-camera transformation, by using higher order polynomials instead:

$$I_i''(x,y) = A_i(I_i'(x,y)) = \sum_{j=0}^{4} a_{i,j} \cdot (I_i'(x,y))^j \tag{5}$$

$I_i'(x,y)$ is the i-th color component of pixel x, y of the camera that we want to calibrate, $a_{i,j}$ are the transformation coefficients which have to be learned, and $I_i''(x,y)$ is the resulting value, for a certain color spectrum $i \in \{Y, U, V\}$, which should match the color as it would appear if taken by the camera of the reference ("alpha") robot. Further, we must obtain I'_i from I_i by applying the vignetting correction as described. All robots in the team have to be calibrated to match the color representation of the "alpha" robot, hence each robot will have its own set of coefficients $a_{i,j}$.

When different cameras exhibit similar vignetting characteristics, the vignetting correction polynomials can be calculated only once for all cameras, then the inter-camera calibration can be performed independently and using much less sample images. In fact, in our experiments we have seen that learning both polynomial sets at the same time can easily lead to over-fitting problems, with the inter-camera calibration polynomials which also end up contributing to compensate the vignetting distortion (for example by clipping the high and low components of the color spectra), but this results in a poor generalization to the colors which are out of the training set.

2.3 Realtime Execution

After optimizing off-line the parameters $\varrho_{i,j}, l_{i,j}, a_{i,j}$ with the techniques presented in Section 3, we are able to calculate the corrected pixel color values (Y'', U'', V''), given the uncorrected ones (Y, U, V) and the pixels' position in the image (x, y). Performing the polynomial calculations for all the pixels in an image is an extremely time-consuming operation, but it can be efficiently implemented with Look-Up Tables:

- $radialLUT[x, y] = \sqrt{(x - u_d)^2 + (y - v_d)^2}$ stores the pre-computed values of the distance of all the pixels in an image from the center of distortion (u_d, v_d);
- $colorLUT[r, l, i] = A_i\left(l - \rho_i\left(r\right) \cdot \lambda_i\left(l\right)\right)$ where $r = radialLUT[x, y]$, $l = I_i(x, y) \in [0 \ldots 255]$ and $i \in \{Y, U, V\}$ is the color spectrum. We fill the table for all the possible values of r, l, c, the size of the table is $256 \cdot r_{max} \cdot 3$ elements, so it occupies only $\approx$ 200KBytes in our case

The look-up tables are filled when the robot is booted; afterward it is possible to correct the color of a pixel in real-time by performing just 2 look-up operations.

3 PARAMETER OPTIMIZATION

The goal is to find an "optimal" parameter set for the described color model given a set of calibration images as previously described. Thus, we consider as optimal the parameter set which minimizes the "function value" (from now on referred to as *fitness*), defined as the sum of squared differences of the pixels in an image from the reference value calculated for the color class in which the image belongs. To calculate the fitness of a certain parameter set, given a log file of images of colored cards, we proceed as follows:

- For each image $I_{i,k}$ in the log file (i is the color band, k the frame number), given its reference value previously estimated $R_{i,k}$, the current fitness F_i^k is calculated as:

$$F_i^k = \sum_{(x,y)} \left(I''_{i,k}(x, y) - R_{i,k}\right)^2 \qquad (6)$$

- The total fitness F_i is calculated as the sum of the F_i^k where each k is an image which belong to a different color class; this to ensure that the final parameters will perform well across a wide spectrum of colors and lighting situations, which otherwise might only fit a very specific situation.

The optimization process is performed independently for each color band $i \in \{Y, U, V\}$. To optimize the vignetting correction, we use for $A()$ (from Equation 5) the identity function, and the references $R_{i,k}$ which are extracted from the same log (and same robot) that we are using for the optimization process. In case of inter-robot calibration instead, only $A_i()$ will be optimized, $\rho_i(), \lambda_i()$ will be fixed to the best functions found to correct the vignetting effect, and the references $R_{i,k}$ are extracted from a log file generated from *another* ("alpha") robot, which is used as reference for the calibration. If we want to optimize the center of distortion (u_d, v_d), we need a set of $\rho_i(), \lambda_i()$ calculated in a previous evolution run[4]: as fitness for this process we use the sum of the fitnesses of all 3 color channels, under the assumption that there is only one center of distortion for all color bands.

3.1 Simulated Annealing (SA)

In Simulated Annealing (Kirkpatrick et al., 1983) in each step the current solution is replaced by a random "nearby" solution, chosen with a probability that depends on the corresponding function value ("Energy") and on the annealing temperature T. The current solution can easily move "uphill" when T is high (thus jumping out of *local minima*), but goes almost exclusively "downhill" as T approaches zero. In our work we have implemented SA as follows (Nisticò and Röfer, 2006).

[4]Otherwise, changing u_d, v_d would have no effect on the fitness

In each step, the coefficients $\varrho_{i,j}, l_{i,j}$ (vignetting reduction) or $a_{i,j}$ (inter-robot calibration) or (u_d, v_d) (center of distortion) are "mutated" by the addition of zero mean gaussian noise, the variance of which is dependent on the order of the coefficients, such that high order coefficients have increasingly smaller variances (decreasing order of magnitude) than low order ones, following the idea that small changes in the high order coefficients produce big changes in the overall function. The mutated coefficients are used to correct the image, as in Equation 5.

For each image $I_{i,k}$ in the log file (i is the color spectrum, k the frame number), given its reference value previously estimated $R_{i,k}$, the current "energy" E for the annealing process is calculated as in Equation 6. The "temperature" T of the annealing is lowered using a linear law, in a number of steps which is given as a parameter to the algorithm to control the amount of time spent in the optimization process. The starting temperature is normalized relative to the initial energy, so that repeating the annealing process on already optimized parameters has still the possibility to perform "uphill moves" and find other optimum regions. The process ends when the temperature reaches zero; the best parameters found (lowest energy) are retained as result of the optimization process.

This approach has proved to work well with our application, however it has 2 shortcomings:

- The search space is very irregular and has multiple local and even *global* optima. One reason for the latter is that the function to be optimized is the product of different terms $(\rho_i() \cdot \lambda_i())$, so that exactly the same result can be achieved by multiplying one term by a factor and the other by its reciprocal, or inverting the sign for both terms, etc.
- The variances used to mutate the coefficients give a strong bias to the final results, as it is not possible to fully explore such a wide search space in a reasonable time; the depicted approach lacks the ability to find "good" mutation parameters, apart from the simple heuristic of decreasing the variance order at the increase of the coefficient order

Both issues can be dealt with efficiently by Evolution Strategies.

3.2 Evolution Strategies (ES)

Evolution Strategies (Schwefel, 1995) use a parent population of $\mu \geq 1$ individuals which generate $\lambda \geq 1$ offsprings by *recombination* and *mutation*. ES with *self-adaption* additionally improve the control of the mutation strength: each parameter which has to be optimized (object parameter) is associated with its own mutation strength (strategy parameter). These strategy parameters are also included in the encoding of each individual and are selected and inherited together with the individual's assignments for the object parameters.

An offspring is created by recombination of the parents. We use two parents to generate an offspring, then such offsprings are subject to mutation.

The selection operator selects the parents for the next generation based on their fitness. In case of the (μ, λ)-strategy only the best μ individuals out of the offsprings are chosen to be the parents of the next generation. Our implementation works similarly as the annealing process, with the following exceptions:

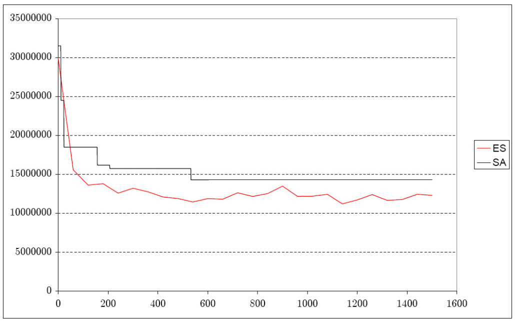
- (μ, λ) strategy with self-adaption, the strategy parameters are initialized with the same sigmas used in the annealing process;
- We terminate the evolution process when n generations have passed without any fitness improvement

Having $\mu > 1$ means that several different "optimal" areas of the search space can be searched in parallel, while the self-adaptation should make the result less dependent from the initial mutation variances.

4 EXPERIMENTS AND RESULTS

At first we compared the two optimization techniques to see what is most suitable for our application. We ran the optimization on a log file containing 9 image classes, the evolution strategy used 8 parents and 60 offsprings, stopping the evolution after 10 generations without improvements. For simulated annealing, we set the number of steps to match the number of fitness evaluations after which ES aborted, ≈ 5000; the total optimization time, for both algorithms, is around 5 minutes, on a 1.7GHz Pentium-M CPU. As it can be seen in Figure 2, both algorithms significantly reduce the image error, but ES found better optima than SA; the results of these techniques are affected by random factors, however except for extremely low evolution times ($< 1min$), ES outperforms SA consistently.

To test the performance of our approach for inter-robot calibration, we created 2 log files containing images representing 6 different color cards taken from 2 robots (one of which we use as reference and call "alpha robot") which show a very different color response, especially for the yellow and pink cards. Table 1 shows that the optimization succeeded in reducing the error standard deviation often by a factor of 3 or more, as well as shifting the modes of the im-



(a) The fitness curves for the correction of the U-channel.

	Y	U	V
init	49677830	31517024	53538715
SA	23608403	14298870	16127666
ES	20240155	10819027	15581825

(b) The initial and achieved best fitnesses.

Figure 2: Results of the vignetting reduction.

Table 1: Inter-robot calibration. $\Delta\mu_{start}, \Delta\mu_{end}$ represent the difference of the mode of the robot from the reference ("alpha") before and after the calibration. $\sigma_{start}, \sigma_{end}$ are the standard deviation from the reference mode.

Color Card	$\Delta\mu_{start}$ Y/U/V	$\Delta\mu_{end}$ Y/U/V	σ_{start} Y/U/V	σ_{end} Y/U/V
green	1/0/3	1/0/4	6.2/3.8/7.4	3.2/3.0/3.9
orange	18/12/5	7/2/3	13.3/17.8/9.3	4.6/4.8/4.2
cyan	13/2/8	5/2/4	11.3/3.9/4.8	4.4/3.8/4.0
red	4/9/5	2/1/1	7.6/15.2/7.5	3.1/5.2/3.6
yellow	6/7/30	5/4/5	21.5/8.3/24.8	6.0/3.9/5.5
pink	17/6/5	3/1/2	24.6/12.6/8.9	5.3/4.3/5.6

ages much closer to the reference values provided by the alpha robot. The total optimization time, split in 2 runs (one for the vignetting, the other for the inter-robot calibration) was approximately 6 minutes.

5 CONCLUSIONS

We have presented a technique to correct chromatic distortion within an image and to compensate for color response differences among similar cameras which equip a team of robots. Since our black-box approach does not use assumptions concerning the physical/geometrical roots of the distortion, this approach can be easily applied to a wide range of camera sensors and can partially deal with some digital sources of distortion such as clipping / saturation. Its efficient implementation has a negligible run-time cost, requiring just two look-up operations per pixel, so it is suitable for real time applications on resource constrained platforms.

REFERENCES

Bruce, J., Balch, T., and Veloso, M. (2000). Fast and inexpensive color image segmentation for interactive robots. In *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '00)*, volume 3, pages 2061–2066.

Iocchi, L. (2007). Robust color segmentation through adaptive color distribution transformation. In *RoboCup 2006: Robot Soccer World Cup X*, Lecture Notes in Artificial Intelligence. Springer.

Kang, S. B. and Weiss, R. S. (2000). Can we calibrate a camera using an image of a flat, textureless lambertian surface? In *ECCV '00: Proceedings of the 6th European Conference on Computer Vision-Part II*, pages 640–653. Springer-Verlag.

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science, Number 4598, 13 May 1983*, 220, 4598:671–680.

Lam, D. C. K. (2004). Visual object recognition using sony four-legged robots. Master's thesis, The University of New South Wales, School of Computer Science & Engineering.

Manders, C., Aimone, C., and Mann, S. (2004). Camera response function recovery from different illuminations of identical subject matter. In *IEEE International Conference on Image Processing 2004*.

Nanda, H. and Cutler, R. (2001). Practical calibrations for a real-time digital omnidirectional camera. Technical report, CVPR 2001 Technical Sketch.

Nisticò, W. and Röfer, T. (2006). Improving percept reliability in the sony four-legged league. In *RoboCup 2005: Robot Soccer World Cup IX*, Lecture Notes in Artificial Intelligence, pages 545–552. Springer.

Röfer, T. (2004). Quality of ers-7 camera images. http://www.informatik.uni-bremen.de/˜roefer/ers7/.

Schulz, D. and Fox, D. (2004). Bayesian color estimation for adaptive vision-based robot localization. In *Proceedings of IROS*.

Schwefel, H.-P. (1995). *Evolution and Optimum Seeking*. Sixth-Generation Computer Technology. Wiley Interscience, New York.

Sony Corporation (2004). *OPEN-R SDK Model Information for ERS-7*. Sony Corporation.

Welch, G. and Bishop, G. (2003). An introduction to the kalman filter.

Xu, J. (2004). Low level vision. Master's thesis, The University of New South Wales, School of Computer Science & Engineering.

USEFUL COMPUTER VISION TECHNIQUES FOR A ROBOTIC HEAD

O. Deniz, M. Castrillon, J. Lorenzo and L. A. Canalis
Instituto Universitario de Sistemas Inteligentes y Aplicaciones Numericas en Ingenieria
Universidad de Las Palmas de Gran Canaria
Edificio central Parque Cientifico-Tecnologico, Campus de Tafira
35017 - Las Palmas - Spain
odeniz@dis.ulpgc.es

Keywords: Social robotics, sociable robot, person detection, omnidirectional vision.

Abstract: This paper describes some simple but useful computer vision techniques for human-robot interaction. First, an omnidirectional camera setting is described that can detect people in the surroundings of the robot, giving their angular positions and a rough estimate of the distance. The device can be easily built with inexpensive components. Second, we comment on a color-based face detection technique that can alleviate skin-color false positives. Third, a person tracking and recognition system is described. Finally, a simple head nod and shake detector is described, suitable for detecting affirmative/negative, approval/disapproval, understanding/disbelief head gestures.

1 INTRODUCTION

In the last years there has been a surge in interest in a topic called social robotics. As used here, social robotics does not relate to groups of robots that try to complete tasks together. For a group of robots, communication is simple, they can use whatever complex binary protocol to "socialize" with their partners. For us, the adjective social refers to humans. In principle, the implications of this are much wider than the case of groups of robots. Socializing with humans is definitely much harder, not least because robots and humans do not share a common language nor perceive the world (and hence each other) in the same way. Many researchers working on this topic use other names like human-robot interaction or perceptual user interfaces. However, as pointed out in (Fong et al., 2003) we have to distinguish between conventional human-robot interaction (such as that used in teleoperation scenarios or in friendly user interfaces) and socially interactive robots. In these, the common underlying assumption is that humans prefer to interact with robots in the same way that they interact with other people.

Human-robot interaction crucially depends on the perceptual abilities of the robot. Ideal interaction sessions would make use of non-invasive perception techniques, like hands-free voice recognition or computer vision. Computer vision is no doubt the most useful modality. Its non-invasiveness is the most important advantage. In this paper, four computer vision techniques for human-robot interaction are described. All of them have been used in a prototype social robot. The robot is an animal-like head that stands on a table and has the goal of interacting with people, see (Deniz, 2006) for details.

2 OMNIDIRECTIONAL VISION

Most of social robots built use two types of cameras: a wide field of view camera (around 70 deg), and a foveal camera. The omnidirectional camera shown in Figure 1 gives the robot a 180 deg field of view, which is similar to that of humans. The camera is to be placed in front of the robot. The device is made up of a low-cost USB webcam, construction parts and a curved metallic surface looking upwards, in this case a kitchen ladle.

As for the software, the first step is to discard part of the image, as we want to watch only the frontal zone, covering 180 degrees from side to side. Thus, the input image is masked in order to use only the upper half of an ellipse, which is the shape of the mirror

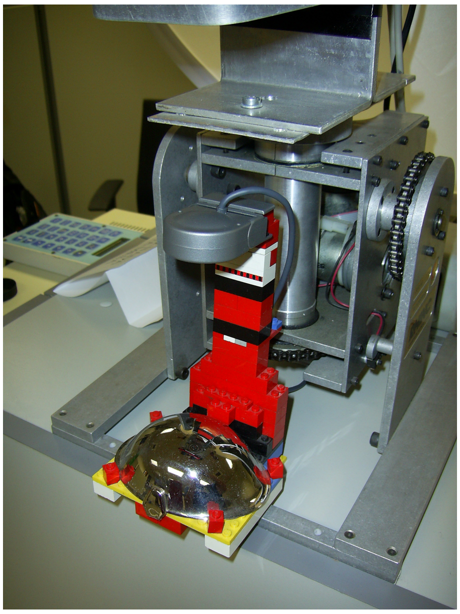


Figure 1: Omnidirectional camera.

as seen from the position of the camera.

A background model is obtained as the mean value of a number of frames taken when no person is present in the room. After that, the subtracted input images are thresholded and the close operator is applied. From the obtained image, connected components are localized and their area is estimated. Also, for each connected component, the Euclidean distance from the nearest point of the component to the center of the ellipse is estimated, as well as the angle of the center of mass of the component with respect to the center of the ellipse and its largest axis. Note that, as we are using an ellipse instead of a circle, the nearness measure obtained (the Euclidean distance) is not constant for a fixed real range to the camera, though it works well as an approximation. The robot uses this estimate to keep an appropriate interaction distance.

The background model M is updated with each input frame:

$$M(k+1) = M(k) + U(k) \cdot [I(k) - M(k)] \quad (1)$$

, where I is the input frame and U is the updating function:

$$U(k) = \exp(-\beta \cdot D(k)) \quad (2)$$

$$D(k) = \alpha \cdot D(k-1) + (1-\alpha)|I(k) - I(k-1)| \quad (3)$$

α (between 0 and 1) and β control the adaptation rate. Note that M, U and D are images, the x and y variables have been omitted for simplicity. For large values of α and β the model adaptation is slow. In that case, new background objects take longer to enter the model. For small values of α and β, adaptation is faster, which can make animated objects enter the model.

The method described up to this point still has a drawback. Inanimate objects should be considered background as soon as possible. However, as we are working at a pixel level, if we set the α and β parameters too low we run the risk of considering static parts of animate objects as background too. This problem can be alleviated by processing the image D. For each foreground blob, its values in D are examined. The maximum value is found, and all the blob values in D are set to that level. Let the foreground blobs at time step k be represented as:

$$B_i = \{x_{ij}, y_{ij}\} \;\; ; \; i = 1,..,NB \; ; \; j = 1,..,N_i \quad (4)$$

There are NB blobs, each one with N_i pixels. Then, after (3) the following is applied:

$$m_i = \max_{j=1,..,N_i} D(x_{ij}, y_{ij}, k) \;\; ; \; i = 1,..,NB \quad (5)$$

$$D(x_{ij}, y_{ij}, k) = m_i \; ; \; i = 1,..,NB \; ; \; j = 1,..,N_i \quad (6)$$

With this procedure the blob only enters the background model when all its pixels remain static. The blob does not enter the background model if at least one of its pixels has been changing.

3 FACE DETECTION

Omnidirectional vision allows the robot to detect people in the scene, just to make the neck turn towards them (or somehow focus its attention). When the neck turns, there is no guarantee that omnidirectional vision has detected a person, it can be a coat stand, a wheelchair, etc. A face detection module should be used to detect people (and possibly facial features). Facial detection commonly uses skin-color as the most important feature. Color can be used to detect skin zones, though there is always the problem that some objects like furniture appear as skin, producing many false positives. Figure 2 shows how this problem affects detection in the ENCARA facial detector (M. Castrillon-Santana and Hernandez, 2005), which (besides other additional cues) uses normalized red and green color components for skin detection.

In order to alleviate this problem, stereo information is very useful to discard objects that are far from

Figure 2: Skin color detection. Note that wooden furniture is a distractor for facial detection. Both the bounding box and the best-fit ellipse are rather inaccurate (left).

the robot, i.e. in the background. Stereo cameras are nowadays becoming cheaper and faster. A depth map is computed from the pair of images taken by a stereo camera situated under the nose of the robot. The depth map is efficiently computed with an included optimized algorithm and library. The map is thresholded and an AND operation is performed between this map and the image that the facial detector uses. Fusion of color and depth was also used in (Darrell et al., 1998; Moreno et al., 2001; Grange et al., 2002). The results are shown in Figure 3. Note that most of the undesired wood colored zones are filtered out.

Figure 3: Skin color detection using depth information.

4 PERSON RECOGNITION

In (Schulte et al., 1999) three characteristics are suggested as critical to the success of robots that must exhibit spontaneous interaction in public settings. One of them is the fact that the robot should have the capability to adapt its human interaction parameters based on the outcome of past interactions so that it can continue to demonstrate open-ended behaviour. CASIMIRO is intended to interact with people. Humans will be the most important ”object” in its environment. Data associated to humans (gathered throughout the interaction) should be stored in memory, so that the robot could take advantage of previous experiences when interacting with them. Breazeal (Breazeal, 2002) argues that to establish and maintain relationships with people, a sociable robot must be able to identify the people it already knows as well as add new people to its growing set of known acquaintances. In turn, this capacity will be part of the robot's autobiographical memory.

In order to make this person memory possible, gathered data should be unambiguously associated to the correct person. Facial recognition would be the perfect approach. However, the experience of the author with face recognition is somewhat negative: face recognition still does not work well in unrestricted scenarios. Recognition rates fall as more time passes since the training samples were taken. Illumination, pose and expression variations normally reduce recognition rates dramatically.

Colour histograms of (part of) the person's body could also be used as a recognition technique. Colour histograms are simple to calculate and manage and they are relatively robust. The price to pay is the limitation that data in memory will make sense for only one day (at the most). Colour histograms of a person's body were used for short-term identification people in (Maxwell, 2003; Kahn, 1996; Maxwell et al., 1999) and also for people tracking (Krumm et al., 2000; Collins and Dennis, 2000).

CASIMIRO achieves person identity maintenance by using colour histograms in conjunction with a simple person tracking algorithm. Tracking is done in 1D, for the interesting position is the angle of the person with respect to the robot.

The implemented tracking algorithm is very simple. Each person is represented as a single point in two sets of horizontal positions (positions range from 0 to 180) at times $t-1$ and t. The association of points between the two sets is obtained as that which minimizes the total sum of distances between points of the two sets. This minimization involves a factorial search, though it is practical for the number of people that will be expected to interact with the robot. Ties can appear, for example in the case of crossings, see the example of Figure 4. These ties are broken by selecting the association with lowest variance of distances, 1 with A and 2 with B in the case of the example. This always selects non-crossings.

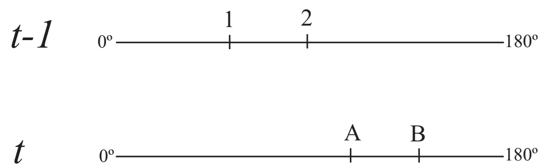


Figure 4: Tie in sum of distances. The sum of distances $|1-A|+|2-B|$ is equal to $|1-B|+|2-A|$. Without further information, we can not know if the two individuals have crossed or not.

Crossings are detected by considering that, in a crossing, there is always a fusion and a separation of person blobs. Person blobs are detected by the omnidirectional vision system (see above). Fusions and separations are detected as follows:

- A blob fusion is detected when the number of blobs in the whole omnidirectional image decreases by one at the same time that one of the blobs increases its area significantly.
- A blob separation is detected when the number of blobs in the image increases by one at the same time that a fusioned blob decreases its area significantly.

The only way to know if a there is a crossing is by maintaining some sort of description of the blobs before and after the fusion. Histograms of U and V colour components are maintained for each blob. The Y component accounts for luminance and therefore it was not used. Whenever a separation is detected, the histograms of the left and right separated blobs are compared with those of the left and right blobs that were fusioned previously. Intersection (Swain and Ballard, 1991) was used to compare histograms (which must be normalized for blob size). This procedure allows to detect if there is a crossing, see Figure 5. The histogram similarities calculated are shown in Figure 6. A crossing is detected if and only if $(b + c) > (a + d)$. Note that in the comparison no threshold is needed, making crossing detection relatively robust.

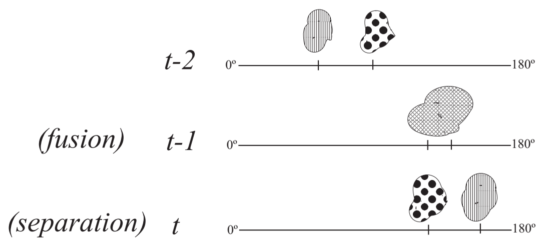


Figure 5: Crossings can be detected by comparing blob histograms at fusion and separation events.

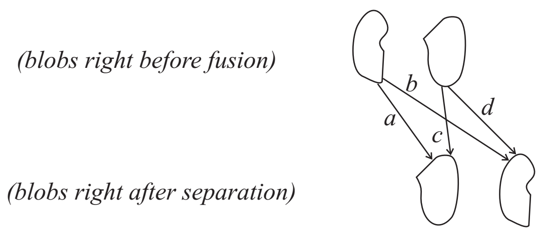


Figure 6: Blob similarities calculated.

In order to achieve person identification, a set of Y-U histograms are stored for each person detected. The zone from which these histograms are calculated is a rectangle in the lower part of the image taken from the stereo camera placed under the nose of the robot. The rectangle is horizontally aligned with the centre of the face rectangle detected, and extends to the lower limit of the image (chest and abdomen of standing people will always occupy that lower part of the image). The upper edge of the rectangle is always under the lower edge of the face rectangle detected. The width of the rectangle is proportional to the width of the face rectangle detected.

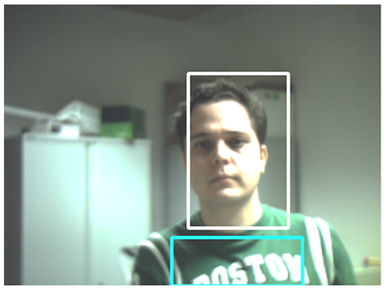


Figure 7: Region used for person identification.

When the robot fixates on a person that the tracking system has labelled as new (the tracking system detects a new person in the scene when the number of foreground blobs increases and no blob separation is detected), it compares the histograms of the fixated individual with those of previously met individuals. This search either gives the identity of a previously seen individual or states that a new individual is in the scene. In any case the set of stored histograms for the individual is created/updated.

5 HEAD NOD/SHAKE DETECTION

Due to the fact that practical (hands-free) voice recognition is very difficult to achieve for a robot, we decided to turn our attention to simpler (though useful) input techniques such as head gestures. Head nods and shakes are very simple in the sense that they only provide yes/no, understanding/disbelief, approval/disapproval meanings. However, their importance must not be underestimated because of the following reasons: the meaning of head nods and shakes is almost universal, they can be detected in a relatively simple and robust way and they can be used as the minimum feedback for learning new capabilities.

The system for nod/shake detection described in (Kapoor and Picard, 2001) achieves a recognition accuracy of 78.46%, in real-time. However, the system uses complex hardware and software. An infrared sensitive camera synchronized with infrared LEDs is used to track pupils, and a HMM based pattern analyzer is used to the detect nods and shakes. The system had problems with people wearing glasses, and could have problems with earrings too. The same pupil-detection technique was used in (Davis and Vaks, 2001). That work emphasized the importance of the timing and periodicity of head nods and shakes. However, in our view that information is not robust enough to be used. In natural human-human interaction, head nods and shakes are sometimes very subtle. We have no problem in recognizing them because the question has been clear, and only the YES/NO answers are possible. In many cases, there is no periodicity at all, only a slight head motion. Of course, the motion could be simply a 'Look up'/'Look down'/'Look left'/'Look right', though it is not likely after the question has been made.

For our purposes, the nod/shake detector should be as fast as possible. On the other hand, we assume that the nod/shake input will be used only after the robot has asked something. Thus, the detector can produce nod/shake detections at other times, as long as it outputs right decisions when they are needed. The major problem of observing the evolution of simple characteristics like intereye position or the rectangle that fits the skin-color blob is noise. Due to the unavoidable noise, a horizontal motion (the NO) does not produce a pure horizontal displacement of the observed characteristic, because it is not being tracked. Even if it was tracked, it could drift due to lighting changes or other reasons. In practice, a horizontal motion produces a certain vertical displacement in the observed characteristic. This, given the fact that decision thresholds are set very low, can lead the system to error. The performance can be even worse if there is egomotion, like in our case (camera placed on a head with pan-tilt).

The proposed algorithm uses the pyramidal Lucas-Kanade tracking algorithm described in (Bouguet, 1999). In this case, there is tracking, and not of just one, but multiple characteristics, which increases the robustness of the system. The tracker looks first for a number of good points to track over the whole image, automatically. Those points are accentuated corners. From those points chosen by the tracker we attend only to those falling inside the rectangle that fits the skin-color blob, observing their evolution. Note that even with the LK tracker there is noise in many of the tracking points. Even in an apparently static scene there is a small motion in them.

The method is shown working in Figure 8. The LK tracker allows to indirectly control the number of tracking points. The larger the number of tracking points, the more robust (and slow) the system. The method was tested giving a recognition rate of 100% (73 out of 73, questions with alternate YES/NO responses, using the first response given by the system).

Figure 8: Head nod/shake detector.

What happens if there are small camera displacements? In order to see the effect of this, linear camera displacements were simulated in the tests. In each frame, an error is added to the position of all the tracking points. If (D_x, D_y) is the average displacement of the points inside the skin-color rectangle, then the new displacement is D_x+e_x and D_y+e_y. The error, which is random and different for each frame, is bounded by $-e_{max} < e_x < e_{max}$ and $-e_{max} < e_y < e_{max}$. Note that in principle it is not possible to use a fixed threshold because the error is unknown. The error also affects to the tracking points that fall outside the rectangle. Assuming that the objects that fall outside the rectangle are static we can eliminate the error and keep on using a fixed threshold, for $(D_x + e_x) - (F_x + e_x) \approx D_x$ and $(D_y + e_y) - (F_y + e_y) \approx D_y$. For the system to work well it is needed that the face occupies a large part of the image. A zoom lens should be used. When a simulated error of $e_{max} = 10$ pixels was introduced, the recognition rate was 95.9% (70 out of 73). In this case there is a slight error due to the fact that the components F_x and F_y are not exactly zero even if the scene outside the rectangle is static.

Another type of error that can appear when the camera is mounted on a mobile device like a pan-tilt unit is the horizontal axis inclination. In practice, this situation is common, especially with small inclinations. Inclinations can be a problem for deciding between a YES and a NO. In order to test this effect, an inclination error was simulated in the tests (with the correction of egomotion active). The error is a rotation of the displacement vectors **D** a certain angle α

clockwise. Recognition rates were measured for different values of α, producing useful rates for small inclinations: 90% (60 out of 66) for $\alpha = 20$, 83.8% (57 out of 68) for $\alpha = 40$ and 9.5% (6 out of 63) for $\alpha = 50$.

6 CONCLUSIONS

four simple but useful computer vision techniques have been described, suitable for human-robot interaction. First, an omnidirectional camera setting is described that can detect people in the surroundings of the robot, giving their angular positions and a rough estimate of the distance. The device can be easily built with inexpensive components. Second, we comment on a color-based face detection technique that can alleviate skin-color false positives. Third, a simple head nod and shake detector is described, suitable for detecting affirmative/negative, approval/disapproval, understanding/disbelief head gestures. The four techniques have been implemented and tested on a prototype social robot.

ACKNOWLEDGEMENTS

This work was partially funded by research projects *UNI2005/18* of Universidad de Las Palmas de Gran Canaria and the Spanish Ministry for Education and Science and FEDER *(TIN2004-07087)*.

REFERENCES

Bouguet, J. (1999). Pyramidal implementation of the Lucas Kanade feature tracker. Technical report, Intel Corporation, Microprocessor Research Labs, OpenCV documents.

Breazeal, C. L. (2002). *Designing social robots*. MIT Press, Cambridge, MA.

Collins, G. and Dennis, L. A. (2000). System description: Embedding verification into microsoft excel. In *Conference on Automated Deduction*, pages 497–501.

Darrell, T., Gordon, G., Harville, M., and Woodfill, J. (1998). Integrated person tracking using stereo, color, and pattern detection. In *Procs. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 601–608, Santa Barbara, CA.

Davis, J. and Vaks, S. (2001). A perceptual user interface for recognizing head gesture acknowledgements. In *Proc. of ACM Workshop on Perceptual User Interfaces*, Orlando, Florida.

Deniz, O. (2006). *An Engineering Approach to Sociable Robots*. PhD thesis, Department of Computer Science, Universidad de Las Palmas de Gran Canaria.

Fong, T., Nourbakhsh, I., and Dautenhahn, K. (2003). A survey of socially interactive robots. *Robotics and Autonomous Systems*, 42(3-4).

Grange, S., Casanova, E., Fong, T., and Baur, C. (2002). Vision based sensor fusion for human-computer interaction. In *Procs. of IEEE/RSJ International Conference on Intelligent Robots and Systems*. Lausanne, Switzerland.

Kahn, R. (1996). *Perseus: An Extensible Vision System for Human-Machine Interaction*. PhD thesis, University of Chicago.

Kapoor, A. and Picard, R. (2001). A real-time head nod and shake detector. In *Proceedings from the Workshop on Perspective User Interfaces*.

Krumm, J., Harris, S., Meyers, B., Brumitt, B., Hale, M., and Shafer, S. (2000). Multi-camera multi-person tracking for easyliving. In *3rd IEEE International Workshop on Visual Surveillance*, Dublin, Ireland.

M. Castrillon-Santana, H. Kruppa, C. G. and Hernandez, M. (2005). Towards real-time multiresolution face/head detection. *Revista Iberoamericana de Inteligencia Artificial*, 9(27):63–72.

Maxwell, B. (2003). A real-time vision module for interactive perceptual agents. *Machine Vision and Applications*, (14):72–82.

Maxwell, B., Meeden, L., Addo, N., Brown, L., Dickson, P., Ng, J., Olshfski, S., Silk, E., and Wales, J. (1999). Alfred: the robot waiter who remembers you. In *Procs. of AAAI Workshop on Robotics*.

Moreno, F., Andrade-Cetto, J., and Sanfeliu, A. (2001). Localization of human faces fusing color segmentation and depth from stereo. In *Procs. ETFA*, pages 527–534.

Schulte, J., Rosenberg, C., and Thrun, S. (1999). Spontaneous short-term interaction with mobile robots in public places. In *Procs. of the IEEE Int. Conference on Robotics and Automation*.

Swain, M. and Ballard, D. (1991). Color indexing. *Int. Journal on Computer Vision*, 7(1):11–32.

A ROBOTIC PLATFORM FOR AUTONOMY STUDIES

Sergio Ribeiro Augusto and Ademar Ferreira

Departamento de Engenharia de Telecomunicações e Controle, Escola Politécnica da Universidade de São Paulo
Av. Prof. Luciano Gualberto 3/380, 05508-900 Sao Paulo, Brazil
sergio@pratica.com.br, ademar@lac.usp.br

Keywords: Mobile robotics, supervised learning, radial basis function networks, teleoperation.

Abstract: This paper describes a mobile robotic platform and a software framework for applications and development of robotic experiments integrating teleoperation and autonomy. An application using supervised learning is developed in which the agent is trained by teleoperation. This allows the agent to learn the perception to action mapping from the teleoperator in real time, such that the task can be repeated in an autonomous way, with some generalization. A radial basis function network (RBF) trained by a sequential learning algorithm is used to learn the mapping. Experimental results are shown.

1 INTRODUCTION

In robotics navigation problems, including learning or not, navigation techniques must be tested in real robots to be useful (DORIGO, 1996). This is due to the uncertainties involved, non uniformity of sensors measurements and real time requirements. To deal with these severe characteristics, this paper proposes a mobile robotics platform developed in a modular and hierarchical way, to be used in real time autonomy studies. The objective is to create a flexible development environment for studies in which teleoperation can be easily integrated with autonomous operation. The idea is to join teleoperation with supervised learning in a way that innate or prior knowledge can be acquired, or that an agent can be taught to realize specific navigation tasks. Such possibility allows a robotic agent to learn with its own operation. Kaelbling (1996) points out that without prior knowledge an agent can not learn with effectiveness. Unsupervised learning techniques, as for example reinforcement learning, have a long convergence time and do not provide operational agents from the beginning. Therefore, it is important to mix such methods with supervised ones (Ye et al., 2003; Er and Deng, 2005).

Although miniature like robots, as for instance the Khepera (Mondada et al., 1993), have been used in studies and papers related to autonomous robotics, as in Er and Deng (2005), it is more realistic to perform the same experiments using larger robots due to the dynamic effects associated, which places them closer to real service robots. For this reason, we decide to build a mobile robotic platform with dynamic characteristics that could be applied in a flexible way to navigation and learning experiments. In this sense, the platform allows sensory-motor data to be stored and recovered during or after operation, and new sensors to be added and configured according to the application.

Differently from Ye et al. (2003) and Er and Deng (2005), in our work the supervised learning takes place in a real environment, not in a simulated one, and in real time. The objective is to teach the agent to perform simple navigation tasks using ultrasound sensors.

In order to have incremental learning with some generalization, a radial basis function neural network (RBF) is developed. We adapted the resource allocation algorithm proposed in Platt (1991) for the function interpolation field, to obtain supervised learning in real time, while the robot is teleoperated. In this aspect, our work is also different from Reignier et al. (1997), where the supervised learning is off line, implemented in a GAL ("Grow and Learn") network, with results verified in simulation.

This paper is organized as follows. Section 2 describes the platform and the software framework developed. Section 3 introduces the supervised learning application. Section 4 presents some experimental results that we got until now. Finally, conclusions are drawn in Section 5.

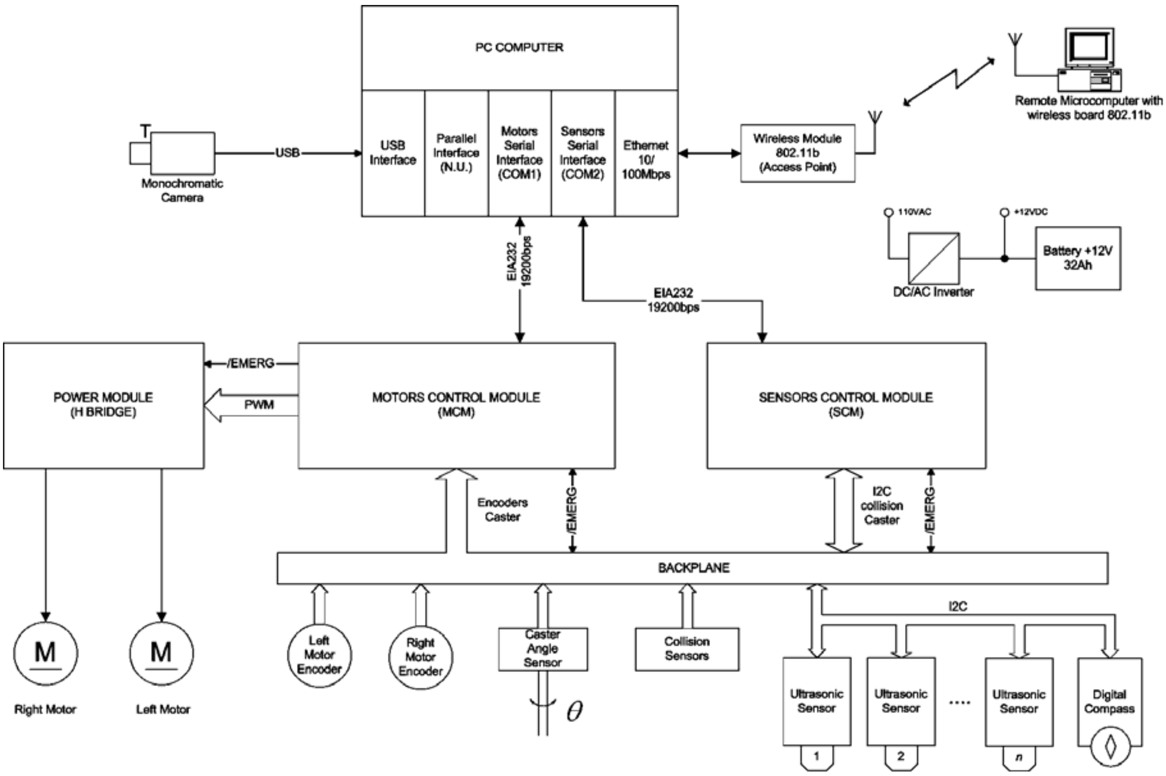


Figure 1: Block diagram of the robotic system.

2 THE ROBOTIC PLATFORM

The robotic platform in its present version is designed for indoor experiments. It measures 50 cm (diameter) by 80 cm (height) and weights 40 kg. In the sequence we describe some aspects of its hardware and software architectures.

2.1 Hardware Architecture

A block diagram of the robotic system is shown in Figure 1. The control of the robot locomotion is accomplished by two motor wheels, powered independently by two DC motors, using differential steering (Dudek and Jenkin, 2000) and a caster wheel. The platform has an image module, and seven ultrasound sensors distributed in its frontal side. The sensors are allocated in a way that objects on the floor can be detected. A digital compass and an angular sensor connected to the caster allow sensorial integration techniques to be exploited to assist in the navigation. Two incremental encoders are used for odometry and velocity control. Collision sensors protect the robot lower perimeter. Other sensors can be added to the platform using a synchronous serial interface available in the system, so that the user can configure it to different types of studies and experiments.

The hardware architecture is arranged in hierarchical processing modules, each one responsible for some of the tasks involved in the mobile robot control. There are four main modules: the Management Module, the Motors Control Module (MCM), the Power Module and the Sensors Control Module (SCM). The Management Module is responsible for the coordination of the robotic unit. Currently a PC on board computer is used for this function. It runs the software framework, described in the section 2.2, for developing of user's applications. The Power Module is responsible for the steering of the motor wheels under control of the MCM. The Motors Control Module implements two PID (proportional plus integral plus derivative) controllers in parallel, allowing independent velocity control of each wheel. The SCM module permits acquiring data from the diverse sensors on the robot. It has a synchronous serial interface (I2C) for sensors expansion and it can also provide for emergency stopping in case of collision.

The platform also has wireless TCP/IP communication resources, allowing remote monitoring, data exchange and teleoperation. The energy system gives the robot at least one hour of navigation's autonomy.

2.2 Software Architecture

The software environment provides the robot with autonomous navigation as well as teleoperation. The software architecture is divided in two main applications: The On Board Management Software (OBMS), running in the Management Module of the mobile platform, and the Remote Control and Supervision Software (RCSS), executing in a remote microcomputer. The communication between them is made using the Client-Server paradigm (Andrews, 2000) and through the wireless network available in the system.

A block diagram of the OBMS is shown in Figure 2. The architecture is arranged in four main levels: the Communication Level, the Management and Supervision Level, the Execution Level and the Software Interface.

The Communication Level implements a TCP/IP server that is responsible for receiving commands from the remote microcomputer and sending data back to it. Simultaneous connections are possible and data can be exchanged with more than one remote microcomputer if desired. The Management and Supervision Level deals with the commands received at the TCP/IP server, interpreting and executing them. This level also performs the management of the mobile unit concerning its operation mode, autonomous or teleoperated, which is controlled through commands sent by the RCSS. The effective control of the robot is made in the Execution Level, which implements the operation modes. This level is easily adapted to the application required using a library of functions available to the user. Each operation mode has a template which the user can modify or adapt to his own necessities. In the application described in this work, the learning algorithm is added to the teleoperation mode and the learned neural network is recovered and executed in the autonomous mode. The Software Interface isolates the hardware aspects of the robot creating an application program interface (API). This permits that hardware modifications can be made without any change in the other levels of the architecture, supplying modularity.

The software framework has a multithreaded architecture (Andrews, 2000), which is adequate to implement real time applications.

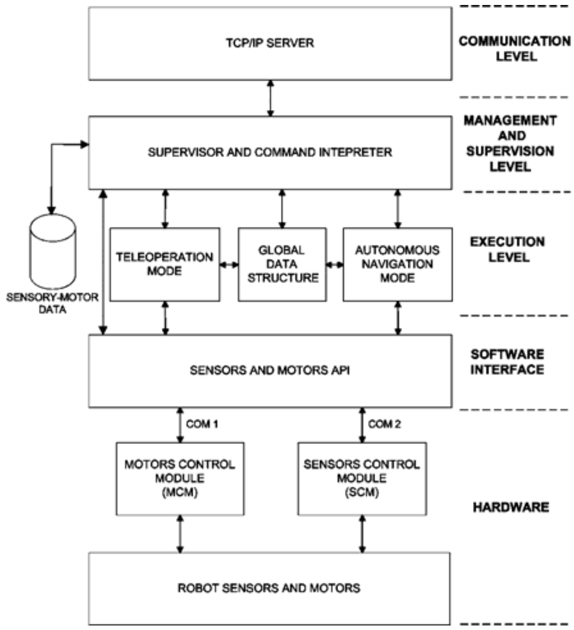


Figure 2: Block diagram of the OBMS.

Sensory-motor data are stored in a data base for analysis and utilization. Sensory-motor coordination aspects (Pfeifer and Scheier, 1997) can then be exploited in the training and learning of autonomous agents. A global data structure allows data exchange among the several software modules in execution.

The RCSS has the main objective of informing the OBMS concerning the operation mode requested by the user. In the teleoperation mode, the robot is controlled through a joystick connected to the remote microcomputer. A TCP/IP client in the RCSS communicates with the OBMS allowing messages and commands exchange.

3 APPLICATION: LEARNING EXPERIMENTS

Using the facilities of the platform, a supervised learning application, assisted by teleoperation, was developed. An RBF neural network was trained in a sequential way appropriate to real time applications. The network starts with no computational units and grows by allocating units (hidden units), or centers, based on the "novelty" of an observation. The novelty is characterized by two joint criterions: the distance criterion and the prediction error criterion. The former is based on the distance between the input pattern observed and the network units. The latter uses the errors among the desired outputs and the network ones due to the input pattern. The network forms a compact representation and it has a quick learning. Learning patterns do not have to be repeated. The units only respond to a local region of the space of the input values, making easy the incremental learning. If a new pattern is presented to the network and the joint criterion is satisfied, a new unit is allocated. Else, the network parameters are update using the LMS (Least Mean Square) algorithm. Instead of using LMS, an algorithm based on the extended Kalman filter (EKF) has been proposed in the literature (Kadirkamanathan and Niranjan, 1993) to speed up the convergence of the network. Because of the computational complexity involved in the EKF, requiring longer processing time, we decided to use LMS in our real time application. The experimental results in section 4 show that our choice was sufficient for the navigation tests that were realized, allowing the training of the robot to the tasks proposed. It is not our objective in this work to minimize the number of teleoperations for learning, so the speed of convergence of the network is not our main approach.

In our proposal, teleoperations are used for training an RBF network that has the seven ultrasound sensors of the platform as input pattern and the angular velocities of the two motor wheels as outputs. The network parameters are updated in real time during the teleoperation and stored in the end of the training. A new teleoperation can be made with the network starting with the stored parameters. The autonomous mode implements the learned network in such a way that the robot can repeat the task with some generalization. This means that the robot produces coherent outputs for similar inputs, although not equal to those encountered during training. If the performance is not good, the platform allows that new teleoperations can be made, starting from the parameters that have already been learned.

The learning algorithm, adapting from Platt (1991), Kadirkamanathan and Niranjan (1993), is described mathematically as follows, where: $\mathbf{x}(n)$ is the input pattern at the instant n; $y_j(n)$ is the angular velocity desired for each wheel (j=1,2); $s_j(n)$ are the network outputs; $\mathbf{u}_k$ is the unit k of the network; w_{jk} is the weight connecting the unit k to the output j; $\varepsilon(n)$ is the value of the distance threshold in the iteration n that inserts a new unit; ε_{max} and ε_{min} are respectively the maximum and minimum values of $\varepsilon(n)$; e_{min} is the threshold to the network prediction error; k_d is the overlap factor to the network units; γ $(0 < \gamma < 1)$ is a decay constant and $\mathbf{u}_{nr}$ is the nearest center to the input $\mathbf{x}(n)$. ε_{max} and ε_{min} represent the scale of resolution in the input space, respectively the largest and the smallest scale of interest.

The network outputs are written as:

$$s_j(n) = \sum_{k=0}^{m} w_{jk}(n-1)\Phi_k(\mathbf{x}(n)) \qquad (1)$$

where m is the number of units or centers of the network, $\Phi_k(\mathbf{x}(n)) = 1$ for $k = 0$ and

$$\Phi_k(\mathbf{x}(n)) = \exp\left(\frac{-\|\mathbf{x}(n)-\mathbf{u}_k(n-1)\|^2}{\sigma_k^2}\right) \qquad (2)$$

for $k \neq 0$.

Algorithm:

In the first iteration ($n = 0$):
$\varepsilon(n) = \varepsilon_{max}$, $w_{j0}(n) = y_j(n)$ ($k = 0$)
For each observation ($\mathbf{x}(n)$; $y_1(n)$, $y_2(n)$)
{
$e_j(n) = y_j(n) - s_j(n)$
If $|e_j(n)|$ ($\forall j \in \{1,2\}$) $> e_{min}$ and $\|\mathbf{x}_n\text{-}\mathbf{u}_{nr}\| > \varepsilon(n)$
Allocate a new unit:
$\mathbf{u}_{m+1} = \mathbf{x}(n)$
$w_{j(m+1)} = e_j(n)$
If it is the first unit
$\sigma_{m+1} = k_d\, \varepsilon(n)$
Else
$\sigma_{m+1} = k_d \|\mathbf{x}(n)\text{-}\mathbf{u}_{nr}\|$
Else execute LMS
$\varepsilon(n) = \max\{ \varepsilon_{max}\gamma^n, \varepsilon_{min}\}$
}

The updating of the network parameters in accordance with the LMS algorithm is given by equations (3) and (4). The time index n was omitted for clarity. Equation (3) is the correction term for the component i of each center k. The dimension of each unit is d, the same of the input pattern (i = 1, 2, …, d). The correction to the weights is given by equation (4). In both equations η is the learning rate.

$$\Delta u_{ki} = \frac{2\eta}{\sigma_k^2}(x_i - u_{ki})\Phi_k(\mathbf{x})\sum_{j=1}^{2} w_{jk}\left(y_j - s_j\right) \qquad (3)$$

$$\Delta w_{jk} = \eta\left(y_j - s_j\right)\Phi_k(\mathbf{x}) \quad (4)$$

4 EXPERIMENTAL RESULTS

Experiments were performed to verify the robot ability to learn the perception to action mapping from the teleoperator in real time, checking if the mobile unit can repeat the navigation task autonomously. The parameters values used in the tests were: $\varepsilon_{min} = 0.03$, $\varepsilon_{max} = 0.5$, $\gamma = 0.9$, $\eta = 0.3$ and $k_d = 0.5$. The input and output data used to train the network were normalized in the range [-1, 1]. Figure 3 shows the results of teleoperating the robot in a corridor with a turn to the right.

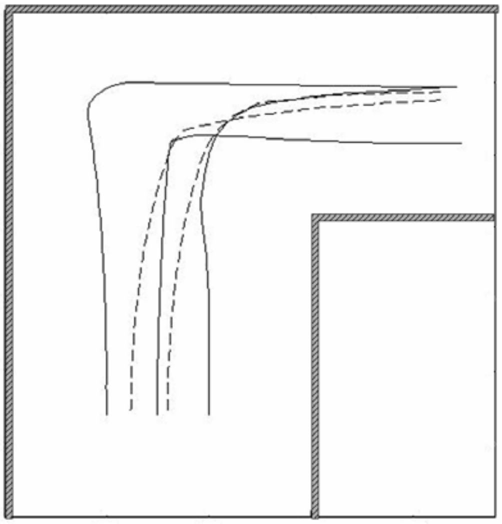


Figure 3: Navigating in a corridor with a turn to the right.

The training consisted of three teleoperations, each one starting at a different initial position. In the figure, the solid lines represent the trajectories executed during the training and the dashed ones are the routes performed autonomously by the robot. The trajectory is the path traversed by the medium point between the two motor wheels. It should be remembered that the unit has a diameter of 50 cm. The training resulted in a network with 121 units.

The idea of the experiments was to demonstrate the capacity of the method in acquiring reactive movements such as wall-following and obstacle avoidance. In Figure 4 the robot goes around an obstacle. Four teleoperations were realized for training, resulting in a network with 119 centers. After the learning phase the autonomous mode was activated and the agent executed the task with success.

The agent was also trained to make a path in the shape of an 8 around two obstacles. Figure 5 shows the test environment.

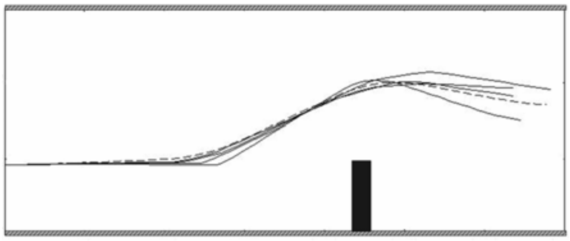


Figure 4: Robot avoiding an obstacle.

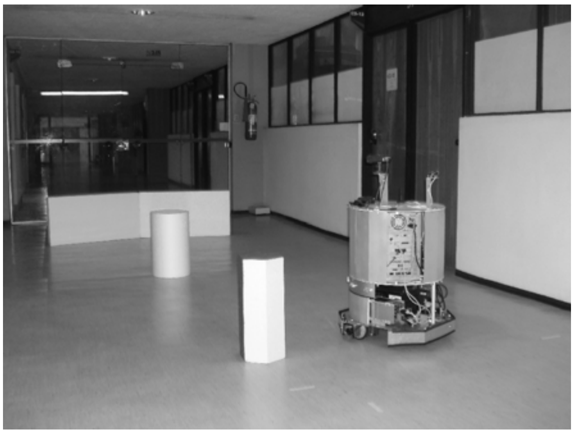


Figure 5: Robot in the test environment.

A few teleoperations were realized to train the robot in this navigation task. In the autonomous mode, it was observed that the robot continued surrounding one of the obstacles instead of completing the 8 when arriving at the trajectory cross point. We then added the value of the digital compass available in the robotic platform to the network input pattern, in a way that the agent could infer the direction of the movement during the learning. Such alteration allowed the robot to complete the task autonomously. The initial training allocated 262 units in the network. Sometimes the robot did not complete the navigation task by itself with success. In those cases the autonomous mode was finished and the agent was teleoperated, completing the route. The incremental and local learning characteristics of the neural network allowed new units to be added to the network encompassing such situations. In the end a neural network with 935 units resulted.

In Figure 6 we have some training trajectories (solid lines) and some paths realized by the robot when operating in the autonomous mode (dash and dash-dot lines).

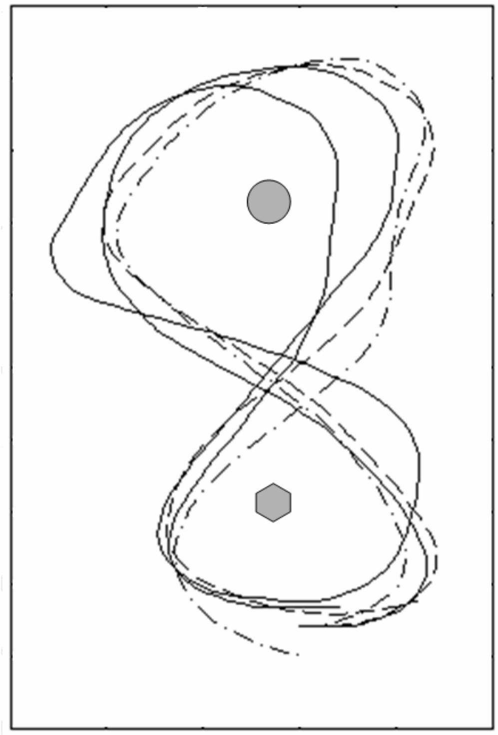


Figure 6: Robot making a path in the shape of an 8.

5 CONCLUSIONS

The mobile robotic platform proposed showed to be efficient to the realization of teleoperated as well as autonomous experiments and studies. The navigation results obtained with the sequential and local learning algorithm used are promising. The results exhibited some generalization, although no specific experiment has yet been made to verify that more systematically. The technique can be applied, for instance, to get prior learning in reactive robotic applications, speeding up real time learning.

As future work, real time pruning techniques should be developed and added to the algorithm to minimize the number of units in the neural network.

REFERENCES

Andrews, G. R. (2000). *Foundations of Multithreaded, Parallel, and Distributed Programming*, Addison-Wesley. New York.

Dorigo, M. (1996). Introduction to the Special Issue on Learning Autonomous Robots. *IEEE Trans. on Systems, Man, and Cybernetics*, Part-B, vol. 26, no. 3, pp. 361-364.

Dudek G. and Jenkin, M. (2000). *Computational Principles of Mobile Robotics*, Cambridge University Press. Cambridge.

Er, M. J. and Deng, C. (2005). Obstacle Avoidance of a Mobile Robot Using Hybrid Learning Approach. *IEEE Trans. on Industrial Electronics*, vol. 52, no. 3, pp. 898-905.

Kadirkamanathan, V. and Niranjan, M. (1993). A Function Estimation Approach to Sequential Learning with Neural Networks. *Neural Computation*, vol. 5, pp. 954–975.

Kaelbling, L. P. (1996). On Reinforcement Learning for Robots. In *Proc. of IEEE Int. Conf. on Intelligent Robots and Systems*, Osaka, Japan, pp. 1319-1320.

Mondada, R., Franzi, E. and Ienne, P. (1993). Mobile Robot Miniaturization: A Tool for Investigation in Control Algorithms. In *Proc. of the Third International Symposium on Experimental Robots*, Kyoto, Japan, pp. 501-513.

Pfeifer, R. and Scheier, C. (1997). Sensory-motor coordination: The metaphor and beyond. *Robotics and Autonomous Systems*, vol. 19, pp. 244-257.

Platt, J. C. (1991). A Resource-Allocating Network for Function Interpolation. *Neural Computation*, vol. 3, no. 2, pp. 213-225.

Reignier, P., Hanser, V. and Crowley, J. L. (1997). Incremental Supervised Learning for Mobile Robot Reactive Control. *Robotics and Autonomous Systems*, vol. 20, pp. 157-178.

Ye C., Yung, N. H. C. and Wang, D. (2003). A Fuzzy Controller with Supervised Learning Assisted Reinforcement Learning Algorithm for Obstacle Avoidance. *IEEE Trans. on Systems, Man, and Cybernetics*, Part-B, vol. 33, no.1, pp. 17-27.

DEVELOPMENT OF THE CONNECTED CRAWLER ROBOT FOR ROUGH TERRAIN

Realization of the Autonomous Motions

Sho Yokota, Yasuhiro Ohyama, Hiroshi Hashimoto and Jin-Hua She
School of Bionics, Tokyo University of Technology, 1404-1 Katakuramachi Hachioji, Tokyo, Japan
yokota@bs.teu.ac.jp, ohyama@bs.teu.ac.jp, hiro-hashimoto@ieee.org, she@cc.teu.ac.jp

Kuniaki Kawabata
RIKEN, 2-1 Hirosawa Wako, Saitama, Japan, kuniakik@riken.jp

Pierre Blazevic
Institut des Sciences et Techniques des Yvelines, 7 rue Jean Hoet, Mantes, France, blazevic@lisv.uvsq.fr

Hisato Kobayashi
Faculty of Eng., Hosei University, 2-17-1 Fujimi Chiyodaku, Tokyo, Japan, h@k.hosei.ac.jp

Keywords: Crawler, Rough terrain, Step climbing, Autonomous motion, Connected crawler.

Abstract: The purpose of this paper is to develop a rough terrain mobile system. Our mobile system adopts the connected crawler mechanism. It had 3 connected stages with the motor-driven crawler tracks on each side. RC-servo motors were used for driving joints between the stages. This system also has a high mobility. In this paper, we showed the mechanical features, and proposed the operation strategies for autonomous motions. We have also made verification experiment of proposed operation strategy. For this verification, we did 2 types of experiment. One was that the robot passes over bumps with different heihgts. The other was stairs ascending. Both experiments had a great success. There were remarkable points in these experiments. These experiments showed that the robot can pass over the different height and different structual obstacles by using only (same) strategy. Moreover the sensors which realize proposed strategy were very simple, and the number of sensor was very small. Therefore it can be concluded that proposed strategy has extremely high usefulness.

1 INTRODUCTION

Since there is a great meaning to use crawler mechanisms as a mobile function on rough terrain, the construction machineries, the tanks, and a lot of rough terrain mobile robots adopt a crawler mechanism. Especially many rescue robots use the crawler mechanisms. Because, in general, crawler mechanisms can obtain big impulsion on rough terrain than the leg mechanism and the wheel mechanisms. On the contrary, it also has weak points as a poor stability in complex geographical features. And the mobility on the area such as stairs is inferior to that of the leg(Hirose, 2000).

Therefore, a lot of researches have tried to supplement with these weak points. The main theme common to those researches is to improve the mobility performance on rough terrain. Generally the variable crawler structure is adopted as an approach for this main theme. In order to realize this transformation, many research proposed connected crawler mechanisms which crawler stages were connected by active joints. Lee et al (Lee, 2003) designs two stages one active joint type resucue robot that uses tow triangular crawlers, and shows the high mobility by the comparison of climb-able step height between proposed mechanism and a usual one track type. ”Souryu-III” (Takayama, 2004) is the connected crawler robots of 3 stages 2 joints type for resuce operations, and it shows high mobility by some basic experiments such as climbing up a step and passing over a gap. ”MOIRA”(Osuka, 2003) is also rescue robot which is 4 stages 3 joints type connected crawler. As mentioned above, the mobility performance was improved by using connected crawler mechanisms.

Although we can see such research, there are no robots which can move autonomously. The one of the most important reason to introducing rescue robots to disaster places is to automate a sufferer searching in place of the manpower searching. If many rescue robots can search sufferers automatically, it is enable to search wider and faster than conventional manpower searching, that brings early detections of sufferers. However current rescue robots don't realize the autonomous operations, therefore that has not achieved above mentioned important reason of introducing robots to disaster places.

Thus this research proposes a rough terrain mobile robot which can realize autonomous motion in disaster places. Especially, this paper proposes operation strategies for passing over obstacles autonomously, as well as constructing of connected crawler robot as the first step of this research.

Also this paper is organized as follows. Chapter 2 introduces the outline of our prototype robot. This mobile robot consists of 3 connected stages with the motor-driven crawler tracks on each side. RC-servo motors were used for driving joints between the stages. Chapter 3 presents operation strategies. Chapter 4 addresses the verification experiments. This experimental results will show that the proposed operation strategies can be adapted to various shape of terrain. Chapter 5 describes the conclusions and future works.

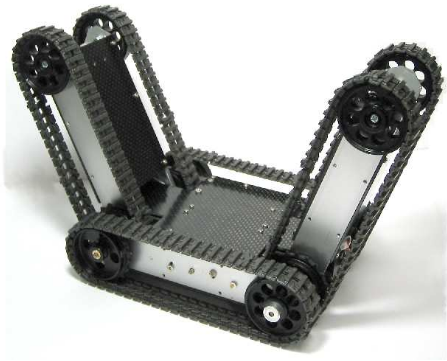


Figure 1: The overview of Connected crawler robot.

Table 1: Specifications of the test model.

Length(maximum)	354.0[mm]
Length(minimum)	118.0[mm]
Width	125[mm]
Mass	0.608[kg]
Radius of the sprockets	20.0[mm]

2 THE PROTOTYPE

This chapter shows the outline of the prototype. The mobile function of our prototype adopts crawler mechanisms. Because The crawler mechanism shows the high mobile ability on various terrains; moreover it is simple mechanism and easy to control. But conventional single track mechanism has also mobility limitations; the limitation is determined by attacking angle, radius of sprockets, and length of crawler. In order to improve its mobility, we add some active joints to conventional crawler tracks, namely that is connected crawler mechanisms.

2.1 Mechanical Structure

Our mobile mechanism has 3 connected stages with the motor-driven crawler tracks on each side(Fig. 1). The features of the proposed mechanism are as follows.

- This mechanism has high mobility to passing over the obstacles.
- It can adjust the size of the robot.
- It can adjust the attack angle.
- It can minimize the grounding area.

Table. 1 also shows the specifications.

RC-servo motors are used for driving joints between the stages. The left and right crawlers are driven by 2 DC motors independently, while the 3 crawlers on each side are driven by a motor simultaneously (Fig.2). The output of each motor is transmitted to the sprockets of the three crawlers through several gears.

2.2 Control Structure

We adopt a hierarchical control structure by installing an intelligent servo driver to each actuator. We connect each of them to the master contoll unit by UART serial line. The parts marked by red line in Figure 3 are servo drivers. Each servo driver consists of one

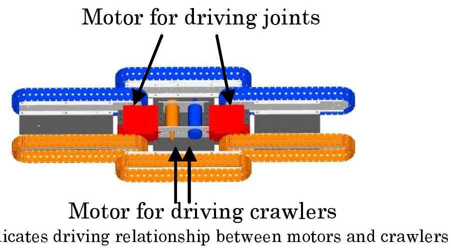


Figure 2: The driving system.

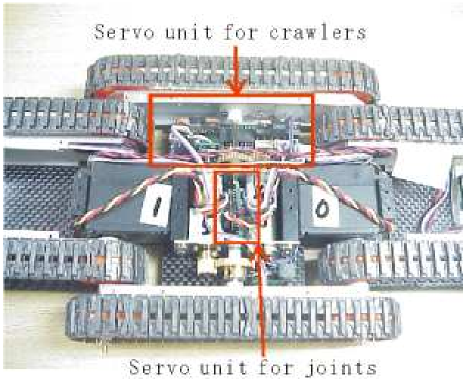


Figure 3: The overview of the servo unit.

Table 2: Communication data format.

<table>
<tr><td colspan="8">1 byte</td><td>2 byte</td><td>3 byte</td></tr>
<tr><td colspan="8">Data 1</td><td>Data 2</td><td>Check Sum</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td rowspan="2">0∼254</td><td rowspan="2">Data1|Data2</td></tr>
<tr><td colspan="4">Mode=0∼2</td><td colspan="4">ID=0∼3</td></tr>
</table>

microcontroller (PIC16F873) and 2 DC motor drivers (TA8440H). One microcontroller is installed to control the two RC-servo units for the joint control, where RC-servo is controlled only by PWM signal.

Figure 4 shows the control structure of this system. The master unit is equipped with several sensors which are increnometers, PSD distance sensor and photo reflector. The usage of these sensors will be shown in Chapter 3. Master unit culclates high level task (setting trajectory, sensing environment, etc), and servo driver works for low level tasks. The master unit processes the high level task, and derive the data to low level task (crawler velocity, joint angel and so on), and send them to the servo drivers. After receiving these data, the servo drivers control their motor by conventional feedback control low. Table 2 shows the communications data formats. The command sent by master unit consists of 3 bytes. First byte indicates mode ID and motor ID. The mode ID distinguishes 3 kinds of control modes: position control, velocity control and compliance control. The motor ID is used for selecting motor to control. Second byte shows the data depends on control modes (crawler velocity, joint angle). The third byte is checksum.

3 OPERATION STRATEGIES

A rough terrain such as disaster places has various shapes. Hence, it is difficult to derive each autonomous motion relative to each shape. But it can

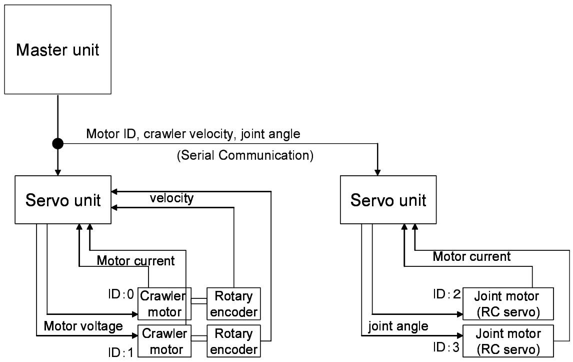


Figure 4: The control system.

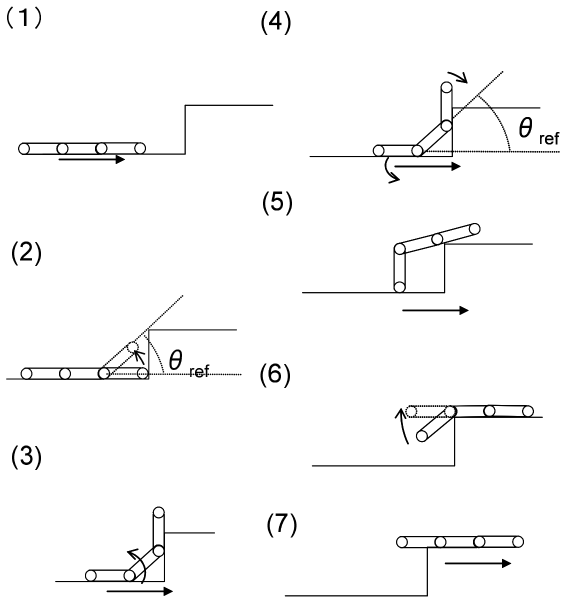


Figure 5: Proposed operation strategies.

be assumed that these shapes are consisted of many bumps. Therefore, in this paper, we set the environment to one bump, and consider about the operation strategies to climb up this one bump. Because the climbing bump ability is important as one of the most fundamental mobility index (Inoh, 2005), in addition climbing bump experiment is adopted by many researches as an evaluation experiment for mobilities. The proposed operation sterategies has 7 steps (Figure 5). It is low level operations(tasks), namely it is not high level operations such as path planning etc. In this paper we assume that the trajectory of the robot is already given. Therefore the proposed operation deals with how the robot can pass over the obstacles. Following sections will show the details of each steps.

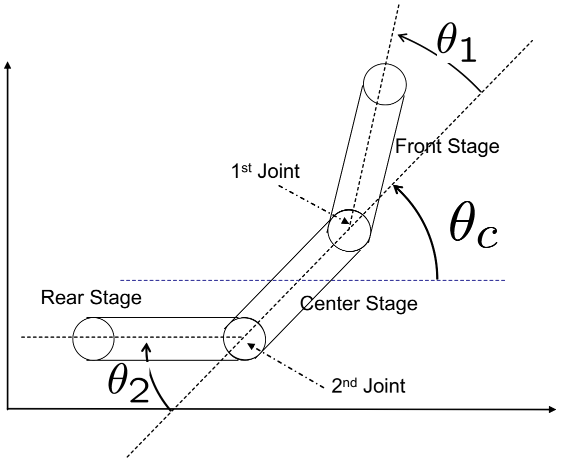


Figure 6: The definition of the parameters.

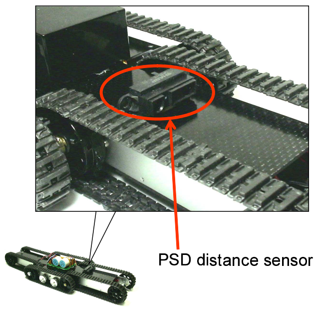


Figure 7: The PSD distance sensor.

Figure 6 is the difinition of the parameters. Here θ_c is the orientation of the center stage. θ_1 and θ_2 are the 1st and 2nd joint angle related to the θ_c. Our proposed operation sterategies can work by using only 3 very simple sensors.

3.1 First Step

First, the robot goes forward until detecting the wall. If the robot faces the wall, then robot stops moving. PSD distance sensor which is attached to the 1st stage is used for detecting the wall (Figure 7). The information of the PSD sensor is managed by the main controller (Figure 4).

3.2 Second Step

In this step, 1st joint are driven to detect θ_{ref}. θ_{ref} is the 1st joint angle when the tangent of front stage meets the edge of the bump (Figure 8).

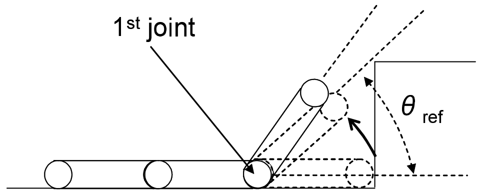


Figure 8: The definition of θ_{ref}.

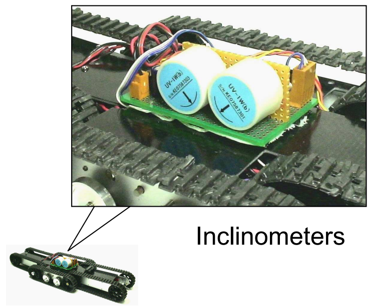


Figure 9: Inclinometers for detecting θ_c.

3.3 Third Step

In third step, 2nd joint is driven while the robot goes forward. The purpose of this step is to get the traction forces by keeping a grounding of rear stage. If the robot goes forward without driving 2nd joint, then robot could not get enough traction forces due to the lift of rear stage. In order to keep the grounding of rear stage, 2nd joint angle should be set to angle of center stage, namely the 2nd joint is driven in the following condition.

$$\theta_2 = \theta_c$$

Here, the inclinometers which are attached to the center stage are used to detect the angle of the center stage (Figure 9).

3.4 Fourth Step

In this step, 1st joint angle is set to 0 rad, 2nd joint is driven to let the angle between rear stage and ground be right angle. At this moment, the robot continues moving. There are two purpose in this step. One is to obtain the traction forces, that is the role of 1st joint motion. The other is to lift up the robot as high as possible, that is the purpose of 2nd joint motion. 2nd joint angle is determined by following condition. By this condition, rear stage can always stand with keeping right angle to the ground.

$$\theta_2 = \frac{\pi}{2} - \theta_c$$

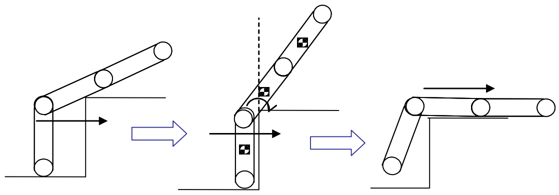


Figure 10: The situation of climbing up a bump.

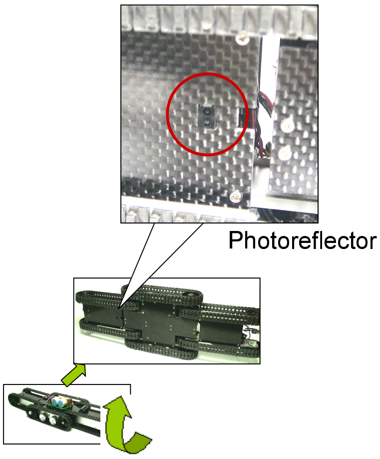


Figure 11: Contact detection device.

The trigger to shift third step to fourth step is the θ_{ref}. In the third step, when the orientation of center stage θ_c is equal to θ_{ref}, operation step is shifted.

3.5 Fifth Step

The robot goes forward with keeping above mentioned conditions. When the center of gravity of the robot is in the right side of the bump edge, then the clock wise moment is generated around the edge, the robot can climb a bump. Figure 10 shows the situation of this case.

3.6 Sixth and Seventh Step

At the end, 2nd joint angle is set to the initial position, not to interfere robot's moving. The trigger for this motion is contact between bump and rear stage. The photoreflector is adopted to detect this contact. This photoreflector is attached to root and bottom of the rear stage (Figure 11).

By above steps, climibing a bump is completed.

4 EXPERIMENTS

In order to confirm proposed operation strategies, verification experiments are conducted. We prepare two kinds of experiments. One is that the robot passes over two bumps with different height. The other is stairs ascending. There are remarkable points in these experiments. These experiment verifies whether the robot can pass over the different height and different structures obstacles by using only proposed strategies. Moreover the sensors which realize proposed strategies is very simple, and the number of sensor is very small. Therefore if these experiments success, it can be concluded that proposed strategies has extremely high usefulness.

4.1 Passing Over the Different Hight Bumps

In this chapter, the robot passes over the different height bumps. The heights of bumps are 150 mm and 40 mm. The experimental environment is indoor, and robot goes only forward, does not rotate and reverse. We made the experiment by implementing proposed strategies to main controller.

The result is shown in Figure 12. This Figure shows that the robot can pass over the different hight bumps autonomously.

4.2 Stairs Ascending

Next experiment is stairs ascending. The height between stairs is 150 mm, that is conventional stairs. The implemented software to main controller is the same as experiment in 4.1, namely we do not add any modification, that is completely same. Then we conducted the experiment.

The result is Figure 13. From this Figure, it is turned out that the robot could ascend stairs autonomously with driving joints.

5 CONCLUSIONS

The purpose of this research is to develop a rough terrain mobile robot which can realize autonomous motion in disaster places. Especially, this paper proposed autnomous operation strategy for passing over obstacles, as well as constructing of connected crawler robot as the first step of this research. The connected crawler robot consisted of 3 crawler stages with active joints. The operation strategies was proposed in Chapter 3. This operation strategies was consisted

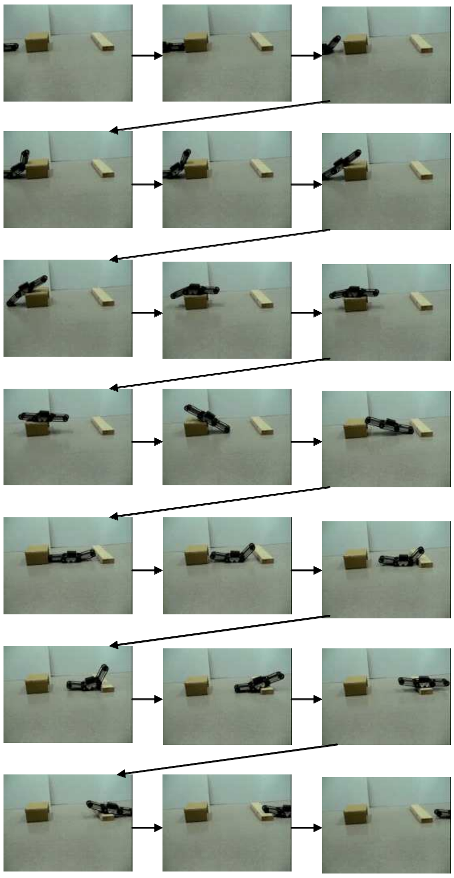


Figure 12: The experimental results of passing over bumps.

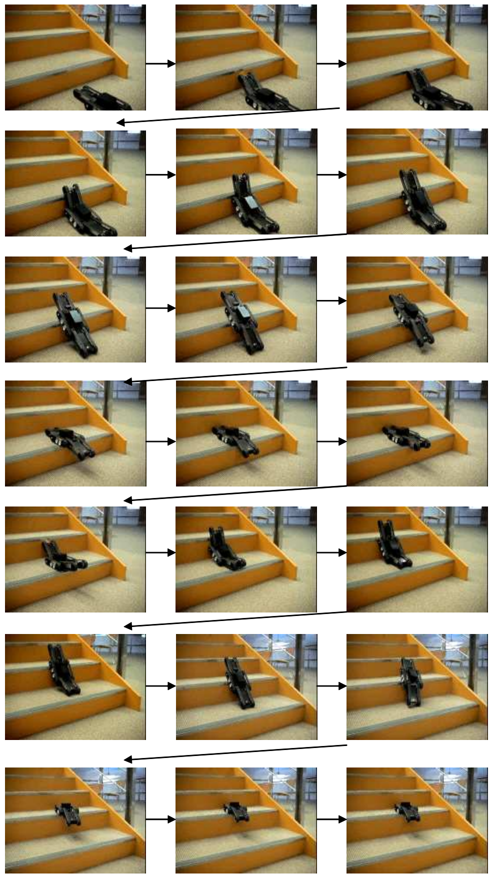


Figure 13: The experimental results of stairs ascending.

of 7 steps, and it needed only 3 simple sensor which were PSD distance sensor, inclinometers and photoreflectors. We have also made verification experiment of proposed operation strategy. For this verification, we did 2 types of experiments. One was that the robot passes over bumps with different heights. The other was stairs ascending.

Both experiments had a great success. There were remarkable points in these experiments. These experiments showed that the robot can pass over the different height and different structual obstacles by using only (same) strategy. Moreover the sensors which realize proposed strategy were very simple, and the number of sensor was very small. Therefore it can be concluded that proposed strategy has extremely high usefulness.

Future works: Proposed method was verified by two experiments. In addition, we are going to verify proposed method by many different types of experiments. Moreover, this paper derived the autonomous motion empirically. Therefore we have to analyze motions of the passing over the obstacles as next step. Furthermore we have to compare empirical results and analytical results.

REFERENCES

Hirose, S. (2000). *Mechanical Designe of Mobile Robot for External Environments (in Japanese)*. Journal of Robotics Society of Japan, Tokyo, vol.18, no.7, pp904-908 edition.

Inoh, T. (2005). Mobility of the irregular terrain for resucue robots. In *10th Robotics symposia*. pp 39-44.

Lee, C. (2003). *Double -track mobile robot for hazardous environment applications*. Advanced Robotics, Tokyo, vol. 17, no. 5, pp 447-495, 2003 edition.

Osuka, K. (2003). Development of mobile inspection robot for rescue activities:moira. In *Proceedings of the 2003 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems*. IEEE.

Takayama, T. (2004). Development of connected crawler vehicle souryu-iii for rescue application. In *Proc. of 22nd conference of Robotics Society of Japan CD-ROM*. Robotics Society of Japan.

A NEW PROBABILISTIC PATH PLANNER
For Mobile Robots Comparison with the Basic RRT Planner

Sofiane Ahmed Ali, Eric Vasselin and Alain Faure
Groupe de Recherce en Electrotechnique Automatique (GREAH) Le Havre University, France
25 Philippe Lebon Street 76058 Le Havre Cedex
sofiane.ahmed-ali,eric.vasselin,alain.faure@univ-lehavre.fr

Keywords: Robotics, motion planning, rapidly exploring random trees.

Abstract: the rapidly exploring random trees (RRTs) have generated a highly successful single query planner which solved difficult problems in many applications of motion planning in recent years. Even though RRT works well on many problems, they have weaknesses in environments that handle complicated geometries. Sampling narrow passages in a robot's free configuration space remains a challenge for RRT planners indeed; the geometry of a narrow passage affects significantly the exploration property of the RRT when the sampling domain is not well adapted for the problem. In this paper we characterize the weaknesses of the RRT planners and propose a general framework to improve their behaviours in difficult environments. We simulate and test our new planner on mobile robots in many difficult static environments which are completely known, simulations show significant improvements over existing RRT based planner to reliably capture the narrow passages areas in the configuration space.

1 INTRODUCTION

Motion planning can be defined as finding path for a mobile device (such a robot) from a given start to a given goal placement in workspace without colliding with obstacles in the workspace. Beside the obvious application within robotics, motion planning also pays an important role in animation, virtual environments, computer games, computer aided design and maintenance, and computational chemistry.

Despite the success of the earlier deterministic motion planning algorithms, path planning for a robot with many degrees of freedom is difficult. Several instances of the problem have been proven to be PSPACE-hard (Reif, 1979) or even undecidable. In recent years random sampling has emerged as a powerful approach for motion planning problems. It breaks the computational complexity in (Reif, 1979) and shows efficiency and its easy way to implement in high dimensional configuration space. Current random-sampling based algorithms can be divided into two sets of approaches: multiple query and the single query methods

The primary philosophy behind the multiple query methods is that substantial pre-computational time may be taken so that multiple queries for the same environment can be answered quickly. The probabilistic roadmap method (PRM) (Svestka, 1997) (Kavraki, 1994) is an example of such method.

The multiple query methods may take considerable pre-computation time thus; different approaches were developed for solving single-query problems. The rapidly exploring random trees (RRTs) is a popular motion planning technique which was primarily designed for single-query holonomic problems and problems with differential constraints (LaValle, 1998), The success of this approach provide their extensions to different motion planning issues from problems with complicated geometries (Ferré, 2004), to manipulation problem and motions of closed articulated chains in, (Yershova and LaValle, 2007). Adapted versions of RRT for non holonomic and kinodynamic motions also exists (Lamiraux and Ferré, 2004),

Even though RRT works well in many applications, they have several weaknesses, which cause them to perform poorly in some cases. Narrow passages are small region which naturally restrict the movements of the mobile robots in one or many directions. Leading to a prohibitively many expensive operations (i.e. collision checks) are being performed during the execution of the algorithm. It is unlikely that a basic RRTs algorithm can overcome this major difficulty entirely.

Recently a new probabilistic approach to find paths through narrow passages areas was proposed (Ahmed ali, Vasselin, and Faure, 2006). The approach is based on the idea of adapting the sampling domain to the geometry of the workspace. In this paper, we illustrate the weaknesses of the RRT planner and we propose a general framework based on the approach (Ahmed Ali, Vasselin, and Faure, 2006) to minimize the effects of some of these weaknesses. The result is a simple new planner that shows significant improvements over existing RRT planners, in some cases by several orders of magnitude. The key idea in (Ahmed Ali, Vasselin, and Faure, 2006) is what we call the *Angular-Domain* a specialized sampling strategy for narrow passages that takes into account the obstacles in the configuration space. Although the idea is general enough and should be applicable to other motion planning problems (e.g. planning for closed chains, non holonomic planning), we focus in this work only on holonomic problems.

The remaining part of the paper is organized as follows. First, the original RRT planers are presented with an illustration of the Voronoi biased exploration strategy. In the end of section 2 we analyze the performance of the RRT algorithm on one challenging example for the RRT planners. Section 3 gives a formal characterization of the *Angular Domain* as a new sampling strategy for narrow passages areas Simulations results in case of holonomic robots are shown in the end of section 3. a sort summary concludes the paper.

2 THE RRT FRAMEWORK

2.1 General Approach

The rapidly random exploring trees (RRT) are incremental search algorithm. They incrementally construct a tree from the initial state to the goal state (bidirectional versions exists as well). At each step, a random sample is chosen and its nearest neighbour in the search tree computed. A new node (representing a new configuration in the free configuration space) is then created by extending the nearest neighbour toward the random sample. See Figure 1 for the construction of the tree and Figure 2 for a pseudo code of the algorithm.

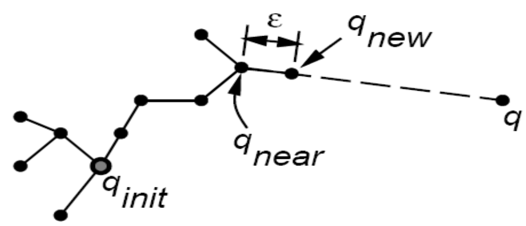


Figure 1: Incremental construction of a basic RRT tree.

Build_RRT(q_{init})

1 $\tau.init.(q_{init})$;

2 **for** $k = 1$ **to** N **do** (until the maximum number of nodes is reached)

3 $q_{rand} \leftarrow Random\_Config()$;

4 $q_{near} \leftarrow Nearest\_Neighbor.(q_{rand}, \tau)$

5 **if** CONNECT $(\tau, q_{rand}, q_{near}, q_{new})$;

6 $\quad \tau.add\_vertex.(q_{new})$;

7 $\quad \tau.add\_edge.(q_{near}, q_{new})$;

8 **if** the goal configuration q_{goal} is reached **then**

Exit $k = N$

Return

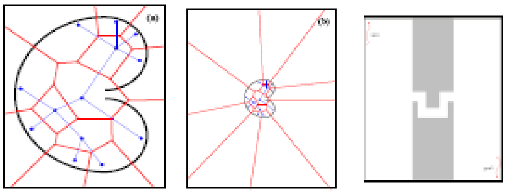
Figure 2: The basic RRT algorithm.

2.2 RRT and Voronoi Bias

This exploration strategy has an interesting property: it is characterized by Voronoi bias. At each iteration, the probability that a node is selected is proportional to the volume of its Voronoi region; hence, the search is biased toward those nodes with the largest Voronoi regions (the unexplored regions of the configuration space

2.3 Bug Trap and Narrow Passages

We consider the problems shown in Figure.3 (a). (c). the task is to move the robot outside the bug trap[1] for the first two figures, and from the left side to the right side through a narrow passage for the second.



(a) (b) (c)

Figure 3: A bug trap problem and a narrow passage in high dimension can be very challenging for RRT planners. The problem become more challenging when the sampling domain is enlarged (b) and (c).

The tree constructed by the RRT planner in the bug trap is shown in blue and the Voronoi region associated with the nodes of the tree are shown in red Figure.3.a. A *frontier* node are vertices in the tree that has their Voronoi region growing together with the size of the environment, while a *boundary* node are those that lie in some proximity to the obstacles. Note that *frontier* nodes are suitable for the RRT planners because they provide a strong bias toward the unexplored portions of the configuration space. The problem is that given the geometry of the narrow passages a *frontier* node is usually a *boundary* node*,* since that the *boundary* nodes are given more Voronoi bias than they can explore; prohibitively many expensive operations are being performed during the execution of the RRT. Finally the tree in the middle of the bug trap or in the narrow passage does not grow at all leading to a considerable slow-down in the performance of the RRT.

Thus, the goal of this paper is to find a way of reducing the number of expansive iterations in RRT. The obvious solution to this problem would be to limit the sampling domain to get more nodes in the middle of the bug trap and the narrow passage. We define a new sampling domain called the Angular domain which tends to get useful nodes which avoid expansive collision checking operations for the RRT.

3 ANGULAR DOMAIN PLANNER

A narrow passage is a difficult region which contains a lot of or huge obstacles and the free space is considerably limited To deal efficiently with a narrow passage we do not need many samples in large open region we do need samples that lies in the narrow passage. Therefore, we take into account in the construction of the tree the obstacles region see Figure 4. We start by giving some definitions we need to formulate the *Angular-Domain.*

3.1 Problem Definition

Let be an n dimensional space, and C_{obs} be the set of obstacles in this space. Let V a set of N collision free points lying inside $CS_{free} = C / C_{obs}$

Definition 1: for $\Im$ a local method that computes a path $\Im(v, v')$ (a straight line segment) between two given nodes in the tree. We define the visibility domain of a point v for $\Im$ as follows:

$$Vis_{\Im}(v) = \{v' \in CS_{free} \ \Im(v, v') \in CS_{free} \ \} \quad (1)$$

Definition 2: for a given goal configuration the visibility domain for a node v is defined as follows:

$$Vis_{\Im,v}(v_{goal}) = \{v \in CS_{free}, \Im(v, v_{goal}) \in CS_{free} \ \} \quad (2)$$

3.2 RRT with Obstacle

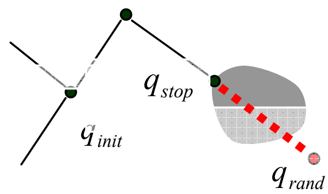


Figure 4: RRT with obstacles.

Once sampling a new configuration q_{rand} (Line 3 Figure 2) the proposed edge $(\overrightarrow{q_{near} q_{rand}})$ might not reach to q_{near}. In this case, a new edge is made from q_{near} to q_s the last possible point before hitting the obstacle (Figure 4). q_s is defined as the last configuration returning a positive response to the collision free test while the interpolation between q_{near} and q_{rand} is being performed. In this paper we use the incremental method as a collision checker indeed, during the interpolation between q_{near} and q_{rand}, we check for collision free test at every placement of the robot. If the interpolation succeeds it is clear that $q_s = q_{rand}$. Since the collision detection operations are the most time consuming steps in the RRT planners our planner must reduce the number of these collision detection operations. The main idea is to reduce the number of the nodes in the tree by adding only the q_s node. The expansion of our tree is performed then from q_s to the nearest neighbor sample which is selected according to a new sampling strategy see figure 5 that keeps the nearest sample candidate in the narrow passage and reduces the number of collision checks to interpolate it to q_s. The algorithm of selecting the nearest neighbor and the complete code of our planner are presented below:

3.3 RRT Planner with Controlling the Sampling Domain

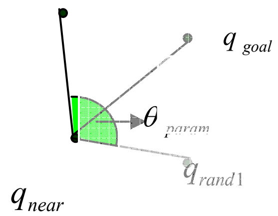


Figure 5: Controlling sampling domain with angular parameter.

Definition 3: given CS_{free} of the configuration space. For a node v_{near} and the goal configuration v_{goal}. The *Angular Domain* is defined as the intersection between CS_{free} and the samples candidate who satisfies the control sampling algorithm (in green colour Figure 5) defined below:

SELECT_NEIGHBOR (q)
1) **Repeat**
2) Pick q_{rand} at random from a uniform distribution over CS_{free} according to a suitable threshold distance R from q.
3) $\theta_1 \leftarrow Angle(\overrightarrow{qq_{rand}, Ox})$
4) $\theta_2 \leftarrow Angle(\overrightarrow{qq_{goal}, Ox})$
5) **If** $|\theta_1 - \theta_2| \leq \theta_{param}$
6) **Return** q_{rand}
7) **End**

BUILD PLANNER (q_{init})
1) $q_{near} = q_{init}$
2) **Repeat**
3) $q_{rand} \leftarrow Select\_Neighbor(q_{near})$
4) $q_s \leftarrow Stopp\_Configuration(q_{rand}, q_{near})$
5) $add\_vertex(q_s)$
6) $add\_edge(q_{near} q_s)$
7) **If** CONNECT $(q_{goal} q_s)$
8) Return path
9) **Else** $q_{near} = q_s$
10) End repeat

Figure 6: The control sampling algorithm and the Angular Domain planner in 2D environment.

3.4 Implementation

Point robot: for those types of robots we have 2 translational degrees of freedom. The configuration of the robot is a vector $q = (x, y)^T$. Once a random configuration q_{rand} is sampled according to threshold distance R which is computed by the Euclidean distance in R^2, the select neighbour algorithm computes two quantities. θ_1 represents the angle between the vector $(\overrightarrow{qq_{rand}})$and the horizontal axis in 2D workspace in which q is the current configuration in the tree and q_{rand} the sample candidate for interpolation. θ_2 is the angle between the vector $(\overrightarrow{qq_{goal}})$ and also the horizontal axis in 2D workspace see figure (5). If the absolute value of the difference between these two quantities is less or equal to some chosen $\theta_{prameter}$ value, the edge (qq_{rand}) is created. The local planner performs the interpolation and checks for collision free each placement.

3.5 Computational Analysis

The running time T for an RRT planner is given by the relation:

$$T = T_{node} \times N_{node} + T_{con} \times N_{con} \quad (3)$$

T_{node} : The average cost of sampling one node

N_{node} : The number of the nodes in the tree

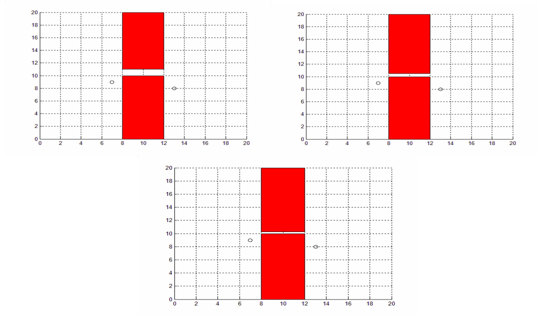
T_{con} : The average cost of checking collision-free connections between two nodes.

N_{con} : The number of calls to check collision-free connections between two nodes.

For the basic RRT algorithm collision checks operation is performed twice. First in line 5 Figure 2 to interpolate q_{rand} and q_{near} . The second collision detection is made to see if the goal configuration is reached or not. The *Angular Domain* planner performs also the collision detection operation twice. First to compute q_s in line 4 figure 6 in the structure of the *Angular Domain* planner. The second time in Line 7 to interpolate q_s to q_{goal} . The difference between the two approaches is in the number of placements we check for collision indeed; since we use the incremental method to check whether a placement is free or not, given two nodes the number of placements we check represent N_{con} . Recall that between two nodes we interpolate until q_s , we are able to reduce N_{con} comparing with the RRT which checks for all the placements between two node

3.6 Simulations

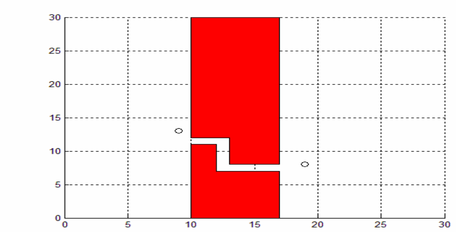
The *Angular Domain* RRT planner and the Basic RRT planner were simulated under Matlab environment. Simulations were performed on a 3.2 GHZ Pentium IV. For each example the performance of for the mono directional RRT algorithms and the *Angular Domain* RRT algorithms are compared. Comparison is performed in terms of the running time and the number of collision checks made by both planners.



	Angular domain planner		RRT planner
Time (1) (s)	1.7500		2.3408
Num.nodes N (1) in the tree	7		30
n_{mill} (1)	191		46
CD calls (1)	4279		3784
Success rate (%) (1)	100%		30%
Time (2) (s)	2.214		6.86
Num.nodes N (2)	15		80
n_{mill} (2)	238		149
CD calls (2)	3967		11479
Success rate (%) (2)	100%		0%
Time (3) (s)	best	worst	10.4220
	4.0160se	46.1250	
Num.nodes N (3)	best	worst	100
	15	20	
n_{mill} (3)	best	worst	202
	280	257853	
CD calls (3)	best	worst	15884
	7297	91850	
Success rate (%) (3)	100%		0%

Figure 7: simulations results for the environment in Figure.

The table Figure 7 shows the result obtained for an environment with a classic narrow passage the results are an averaged of 50 runs over the three environment. The success rate characterizes the performance of both planners to find the solution path. The first observation we made on these results is that as the width of the narrow passage became smaller the performance of the basic RRT planner deteriorate quickly (see the success rate lines for the three environments).the deterioration of the performance is explained by the fact that the size of the free space is considerably larger than the narrow opening in the three environments We make the second observation on the third environment, as it was mentioned in the computational analysis section the threshold distance and the angular parameter (set to 10 and $\pi/2$) must be chosen carefully. We can see that n_{mill} has a very large value (see Line14 Figure 7) leading to increase the total running time of the algorithm in the worst case.



	Angular domain				RRT N=5			
R	5	10	20	30	5	10	20	30
time	3.81	4.17	2.96	2.23	0.46	0.32	0.34	0.53
CD calls	8098	7081	3369	2113	571	554	579	565
n_{mill}	930	1431	868	69	18	5	6	6
Succ es (%)	100	100	100	100	0	0	0	0
	RRT N=80				RRT N=200			
R	5	10	20	30	5	10	20	30
time	3.71	3.92	2.96	4.12	24	24	25	25
CD calls	5597	5838	5884	6045	22335	23504	23892	23193
n_{mill}	75	66	64	68	293	297	283	285
Succ es (%)	0	0	0	0	0	0	0	0

Figure 8: simulation results for the environment with different N (the maximum number of the node for RRT).

The simulation results demonstrate the efficiency of the Angular Domain planner. We take different values of R, it appears that the optimal threshold distance for the environment figure 8 is 30; it gives also the smallest running time. Note that for a small threshold distance (5 and 10) we can see that n_{mill} is big leading to increase the total running time of our algorithm. Therefore for a given problem the balance between too small or to large value for the threshold distance can be difficult to find indeed; too small value may increase dramatically n_{mill} and by the way the total running time T in the other hand too large value may potentially add many nodes in the open free space while we need much nodes in the narrow passage.

4 CONCLUSIONS AND FUTURE WORK

There are to ways to improve the current work. First the threshold distance and the angular parameter are chosen manually a promising approach is to adjust these two parameters through on line learning. The tuning of these two parameters will be obviously based on the position of the obstacles in the workspace leading to get an efficient planner for different kinds of obstacles.

Another important direction is to apply this frame work for other constrained motion planning problems such articulated robot

REFERENCES

J. H. Reif, *"Complexity of the mover's problem and generalizations," in Proceedings of the 20th IEEE Symposium on Foundations of Computer Science, 1979, pp. 421-427.*

P. Svestka, *Robot motion planning using probabilistic roadmaps. Ph.D Thesis, Utrecht University, 1997.*

L. E.Kavraki, J.-C.Latombe*, Randomized preprocessing of configuration space for fast path planning, In:IEEE Int Conf On Robotics and Automation,1994,pp.2138-2145.*

S. M. La Valle *Rapidly-exploring random trees: A new tool for path planning. TR 98-11, Computer Science Dept., Iowa State University, Oct. 1998.*

E. Ferré and J.-P. Laumond *An iterative diffusion method for part disassembly. In IEEE Int.Conf .Robot.& Autom.,2004*

A. Yershova and S. M. LaValle. *planning for closed chains without inverse kinematics, in: IEEE Conf On Robotics and Automation, 2007 to appear*

F. Lamiraux, E.Ferre, and E. Vallee. *kinodynamic motion planning: connecting exploration trees using trajectory optimization methods. In IEEE In Conf Robot & Autom, pages 3987-3992, 2004.*

Sofiane Ahmed Ali, Eric Vasselin and Alain Faure. *"A new Hybrid sampling strategy for PRM planners to address narrow passages problem". In Proceedings 3rd International Conference on Informatics in Control, Automation and Robotics (ICINCO), 2006.*

SEGMENTATION OF SATELLITE IMAGES IN OPTOELECTRONIC SYSTEM

Andrey S. Ostrovsky, Ernesto Pino-Mota and Paulo C. Romero-Soría
Facultad de Ciencias Físico Matemáticas, Benemérita Universidad Autónoma de Puebla
A.P. 1152, 70000, Puebla, Pue., Mexico
andreyo@fcfm.buap.mx, epino@fcfm.buap.mx

Keywords: Image Segmentation, Image Classification, Synthetic Discriminant Functions, Optical Image Processing.

Abstract: The problem of segmenting the satellite images into homogeneous texture regions that correspond to the different classes of terrestrial surface is considered. It is shown that this problem may be successfully solved by using the method of spectral synthetic discriminant functions recently proposed by the authors for classification of random image fields and realized by means of a rather simple optoelectronic technique. The experimental results of segmenting the true satellite images are given.

1 INTRODUCTION

One of the central problems in automatic processing of satellite images is to segment the given image into homogeneous texture regions corresponding to different classes of the terrestrial surface such as different urban zones, mountainous zones, wooded zones, agricultural zones, aquatic zones, etc. (Wu *et al*, 1995). A specific feature of this problem is in the fact that the images to be classified have fundamentally random within-class variations so that they must be viewed as being perfectly random or stochastic. In this situation, one is better off talking about the random image field and not the image itself, i.e., as a deterministic function of space. Recently we proposed a new method for classification of such images in which we use the special discriminant functions being synthesized to separate linearly the power spectra of random image fields of different classes (Ostrovsky *et al*, 2003). We refer to this method as spectral synthetic discriminant function (SSDF) method. In this paper, we show how the SSDF method realized by menas of a rather simple optoelectronic system may be used for segmenting the satellite images.

2 SSDF METHOD

We consider a certain image of the *n*th classs ($n = 1,\ldots, N$) as the 2-D *k*th ($k = 1,\ldots, K$) sample function $f_{nk}(x,y)$ of a stationary and isotropic random field $f_n(x,y)$ with a power spectrum

$$S_n(\rho) = \lim_{R\to\infty} \frac{1}{R}\left\langle \left| F_{nk}(\rho,\theta;R)\right|^2 \right\rangle , \qquad (1)$$

where

$$F_{nk}(\rho,\theta;R) = \int_0^R \int_0^{2\pi} f_{nk}(r,\varphi) \times \exp[-i2\pi r\rho\cos(\varphi-\theta)]\, r\mathrm{d}r\mathrm{d}\varphi , \qquad (2)$$

is the finite Fourier transform of $f_{nk}(x,y)$ over the domain of radio *R* occupied by the image, (r,φ) and (ρ,θ) are the polar coordinates in the spatial and spatial-frequency domains respectively, and the angular brackets denote the expected value operation over the ensemble index *k*. The SSDFs are defined as linear combinations of power spectra $S_n(\rho)$, i.e.,

$$h_m(\rho) = \sum_{l=1}^{N} a_{ml} S_l(\rho), \quad m = 1,\ldots, N , \qquad (3)$$

such that the following identity is determined:

$$\int_0^\infty S_n(\rho) h_m(\rho)\,\mathrm{d}\rho \equiv \delta_{nm} , \qquad (4)$$

where δ_{nm} is the Kronecker symbol. On substituting for $h_m(\rho)$ from Eq.(3) into Eq. (4) and taking for granted the hypotesis of linear independence among the power spectra $S_n(\rho)$ for different classes, one can find the unknown coefficients a_{ml} as the solutions of the system of linear equations

$$\sum_{l=1}^{N} a_{ml} \int_0^{\infty} S_n(\rho) S_l(\rho) \mathrm{d}\rho = \delta_{nm} \ , \ n, m = 1, \ldots, N \ . \quad (5)$$

Once the SSDFs have been calculated in accordance with Eq. (3), a procedure for classifying the unknown sample image $f_{0k}(x, y)$ is to verify identity (4) for every m when substituting for $S_n(\rho)$ the power spectrum $S_0(\rho)$ of corresponding image field $f_0(\rho)$.

As can be seen from Eq. (3), in oder to determine the SSDFs, is necessary to know each power spectrum given by Eq. (1); this presupposes averaging over the infinite ensemble of infinitely extensive sample images. Actually, we always have available a finite number of finitely extensive sample images, a fact that leads to the statistical formulation of the problem.

The quantity that can be directly measured in an experiment is the sample power spectrum integrated in the azimuthal direction, i.e.,

$$S_{nk}(\rho; R) = \frac{1}{2\pi R} \int_0^{2\pi} \left| F_{nk}(\rho, \theta; R) \right|^2 \mathrm{d}\theta \ . \quad (6)$$

In the stage of SSDF synthesis, when we commonly dispose a sufficiently large number of sample images, the consistent estimate of power spectrum $S_n(\rho)$ can be obtained by averaging the sample spectra (6) over the ensemble index k:

$$\hat{S}_n = \frac{1}{K} \sum_{k=1}^{K} S_{nk}(\rho; R) . \quad (7)$$

In the stage of classification, usually just one sample image is available, so that identity (4) has to be substituted for the equation

$$\sum_{l=1}^{N} a_{ml} \int_0^{\infty} S_{nk}(\rho; R) \hat{S}_n(\rho) \mathrm{d}\rho = u_{nmk} \ , \quad (8)$$

where u_{nmk} is the kth sample value of some random variable u_{nm}. To maximize the reliability of correct classification it is obvious to require that

$$\langle u_{nm}(a_{ml}) \rangle = \delta_{nm} \quad (9)$$

and

$$\mathrm{Var}\, u_{nm}(a_{ml}) \to \min_{a_{ml}} . \quad (10)$$

This can be readily achieved by applying the well known least-square technique. Once the SSDFs have been calculated in this way, the decision on the class to which the sample image $f_{0k}(x, y)$ belongs can be made according to index m of the largest value u_{0mk}.

3 OPTICAL REALIZATION

As apears from the previous section, the fundamental problem with practical realization of the SSDF method is calculating the sample power spectrum given by Eq. (6). For this purpose the coherent optical Fourier processor shown in Fig. 1 may be employed.

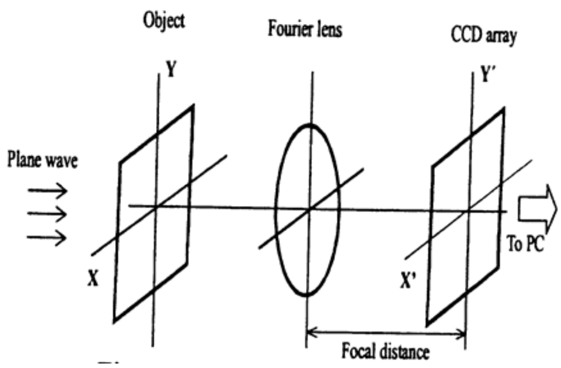


Figure 1: Optical Fourier processor.

As is well known (Goodman, 1986), if in the object plane of this processor a transparency with amplitude transmittance $f(x, y)$ within a finite domain D of radio R is placed, then the intensity distribution of light field registered by the CCD detector array in the back focal plane of the Fourier transforming lens is given by

$$I(p,q) = \left| \iint_{(D)} f(x,y) \times \exp[-i2\pi(xp+yq)]\,\mathrm{d}x\mathrm{d}y \right|^2, \qquad (11)$$

where $p = x'/\lambda f$, $q = y'/\lambda f$, λ is the wavelength of illumination, and f is the focal length of the lens. Using the polar coordinates for input and output planes of the Fourier processor, Eq. (11) may be rewritten as follows:

$$I(\rho,\theta) = \left| \int_0^R \int_0^{2\pi} f(r,\varphi) \times \exp[-i2\pi r\rho(\varphi-\theta)]\,r\mathrm{d}r\mathrm{d}\varphi \right|^2. \qquad (12)$$

Comparing the latter equation with Eq. (2) and (5), we come to conclusion that the sample power spectrum $S_{nk}(\rho;R)$ may be easily calculated in any PC-compatible system connected with CCD array.

4 EXPERIMENTAL RESULTS

We performed a physical simulation experiment on segmentation of true satellite (Landsat) images into homogeneous regions that correspond to four different classes of terrestrial surface, to wit, “sea”, “mountains”, “crops” and “settlement”. An example of such an image is shown in Fig. 2. Each class of texture images to be classified was considered a spatially stationary and isotropic random field.

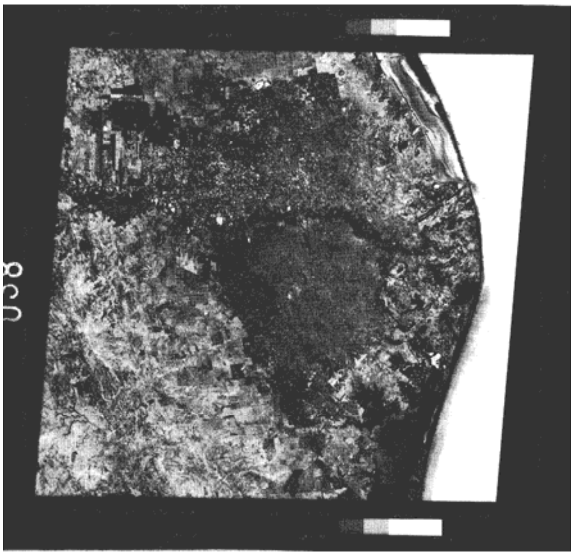


Figure 2: True satellite image used in the experiment.

The experimental setup is sketched schematically in Fig. 3. The images to be processed in this setup were previously converted into numerical files using a standard scanning technique. To provide the capture of the digitalized images into the optical Fourier processor, the liquid crystal spatial light modulator HoloEye-LC2002 (800×600 pixels) controlled by PC was used. The detection of the light distribution in the output plane of the Fourier processor was realized by means of the CCD camera SONY-SSC-M374 (768×494 pixels).

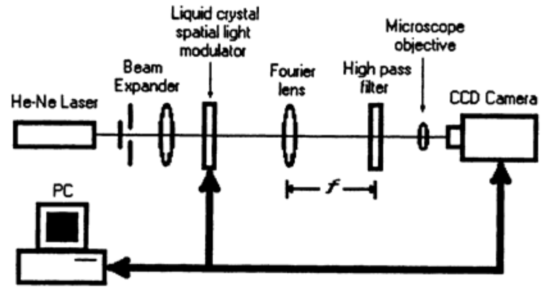


Figure 3: Experimental setup.

In our experiment, at the stage of SSDF synthesis, we used 50 images of 10×10 mm for each of four classes. At the stage of segmentation we used the composed full-scale image of 100×100 mm and realized its 2-D scanning by an aperture of 5×5 mm with a discrete step of 5 mm. At every step of image scanning the corresponding texture image was classified in accordance with the SSDF method. Calculation of SSDFs and values u_{0mk} (see Eq. (8)) was realized in a PC-system using specially designed software. Decision on image class at every step of scanning was made on the basis of thresholding the output data. If none or more than one of u_{0mk} values had exceeded the threshold, the corresponding image region was considered to be unclassified. The result of four-class segmentation that corresponds to the satellite image in Fig. 2 is shown in Fig. 4. The regions of the segmented image labeled by “0” correspond to unclassified images. As can be seen, in the main these regions correctly repeat the true shape of the boundaries between terrestrial surface images of different classes, but, in certain cases, they occupy a rather large area of the image. The latter can be explained by the fact that, in reality, our satellite image contains textures of more than four classes mentioned above (e.g., regions of the terrestrial surface covered by clouds), but more profound study of this problem exceeds the limits of our consideration.

2	2	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1
0	0	0	0	4	4	4	0	0	0	3	3	0	0	0	1	1	0	1	1
0	4	0	4	0	4	4	4	0	3	3	3	3	0	0	0	0	1	1	1
0	4	4	0	4	0	0	0	0	3	3	3	3	3	0	0	0	1	1	1
4	4	4	4	0	0	0	0	3	3	3	3	3	3	3	0	0	0	1	1
0	4	4	4	0	0	0	0	3	3	3	3	3	3	3	3	0	0	1	1
0	0	3	3	0	0	0	0	3	0	0	0	0	0	3	0	0	1	0	1
0	0	0	0	0	0	0	3	3	3	3	0	3	3	0	0	0	0	0	1
0	0	0	0	0	0	0	3	3	3	3	3	3	3	3	3	0	0	0	1
0	0	1	0	0	0	0	0	3	3	3	3	3	3	3	0	0	0	0	1
0	0	1	0	0	0	0	0	3	3	3	3	3	3	3	0	0	0	0	1
2	0	0	1	0	0	4	0	3	3	3	3	3	3	0	0	0	0	1	1
2	2	0	0	0	4	4	0	3	3	3	3	3	3	0	0	0	0	1	1
2	2	2	0	0	0	4	4	0	3	3	3	3	3	0	0	0	0	1	1
2	2	2	2	0	0	0	0	0	3	3	3	3	0	0	1	1	0	1	1
2	2	2	2	0	0	1	0	0	0	3	3	0	0	0	1	0	0	1	1
2	2	2	2	2	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1
2	2	2	2	2	2	0	0	1	0	0	0	1	1	0	0	0	0	1	1
2	1	2	2	2	2	2	0	0	0	0	0	1	0	0	1	0	1	1	1
2	2	2	2	2	2	2	0	1	1	1	0	0	0	0	0	0	1	1	1

Figure 4: Labled map of the satellite image shown in Fig. 2. Lables: 1 – “sea”, 2 – “mountains”, 3 – “crops”, 4 – “settlement”, 0 – “unclassified image”.

5 CONCLUSIONS

As has been shown the problem of segmentating the satellite images into homogeneous regions that correspond to different classes of the terrestrial surface may be successfully solved by using the SSDF method for classification of random image fields realized by means of rather simple optoelectronic technique.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the support of Mexican National Council of Science and Technology – CONACYT, under project SEP-2004-C01-46510.

REFERENCES

Wu, H.-S., Qian, J., Barba, J., 1995. A classified algorithm for segmentation of multispectral images, *Journal of Imaging Science and technology*, Vol. 39.

Ostrovsky, A.S., Pino-Mota, E., Palma-Cuatianquiz, J., 2003. Optical classification of random image fields using spectral synthetic discriminant functions. *Optics and Laser in Engineering*, Vol. 40/1-2.

Goodman, J.W., 1986. *Introduction to Fourier optics*, McGraw. New York, 2nd edition.

ULTRA VIOLET IMAGING TRANSDUCER CONTROL OF A THERMAL SPRAYING ROBOT

D. Breen, E. Coyle and D. M. Kennedy
Faculty of Engineering, Dublin Institute of Technology, Dublin, Ireland
dermot.breen@dit.ie

Keywords: Thermal spraying, robot control, ultraviolet lighting, image processing.

Abstract: The thermal spraying industry has a global market of $1.3 billion. This industry relies heavily on manual operation of the thermal spraying equipment or in some cases, robotic systems that require costly set up of material for surface coating and time consuming trajectory planning. The main objective of this research was to investigate novel ideas for automating the thermal spraying process. This requires transducers that can provide information about arbitrarily shaped and orientated material for spraying and generating the trajectory plan for the robot manipulator during the thermal spraying process in real time. The most significant difficulty for any transducer, particularly low cost vision systems is the thermal spraying process which in our research is molten material such as aluminium in an Oxy-Acetylene flame with temperatures exceeding 3100^0C. This paper outlines the concept and based on the experimental results presented demonstrates combined optical and image processing techniques for obtaining information about objects behind a butane flame.

1 THERMAL SPRAYING ROBOT

1.1 Introduction

Thermal spraying robotic research is concerned with investigating a number of novel ideas, which will contribute to the autonomous control of an articulated thermal spraying robot manipulator. This control of the thermal spraying process, which is used in the application of wear, corrosion and thermal barrier surface coatings will improve safety, efficiency and costs in the surface coating industry.

Thermal spraying has an estimated global market of $1.3 billion dollars (AZoM). The operation of thermal spraying equipment requires the consideration of health and safety issues.

1.2 Health and Safety

In industrial applications, thermal-spraying equipment is normally enclosed in specialist enclosures designed to reduce noise, fumes and observation via safety equipment by operators from a safe location. R&D may not have these specialist enclosures. Therefore health and safety risks must be managed via appropriate health and safety equipment and procedures. Powder Flame spraying with an Oxy-Acetylene torch which is the system used in this research produces intense bright flames with a peak temperature in excess of $3{,}100^0$C.

Two-wire electric arc and plasma spraying systems produce UV-B and UV-C with their associated health and safety risks to the operator. Figure 1 shows a typical flame from the thermal spraying process.

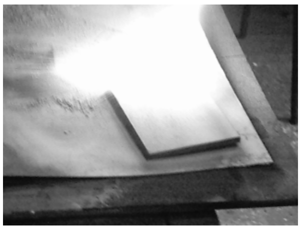


Figure 1: Thermal Spraying Process Flame.

The research presented in this paper uses UV-A lighting which is also present in the thermal spraying process.

1.3 Robot Control

The control of a robot manipulator requires information about the kinematics and dynamics of the robotic system being used for the thermal spraying system. Information about the position and orientation of the thermal spraying torch tip at different locations along the object to be sprayed and at different times which is known as trajectory planning is produced. This information is supplied into the robotic control system, which is used by the inverse kinematic equations and dynamic equations of motion to move the robot actuators to the desired locations. Thermal spraying automation provides this trajectory planning information via pre-programming for specific objects, which is time-consuming and costly.

The autonomous analysis of the position and orientation of the thermal-spraying torch, which would allow the spraying of unspecified objects at unspecified orientations, would significantly reduce set-up times and costs. However this level of automation is significantly hampered by the thermal spraying process. It is quite clear that the intense flame would hamper many object-measuring systems which could be used to obtain in real time, the position and orientation of the thermal spraying torch information. If however the flame could be removed from the scene and a low cost camera used to view the object with associated distance measuring techniques, this would accommodate the autonomous control of the thermal spraying process.

This research attempts to provide a possible solution to this difficult requirement.

2 FLAME REMOVAL

2.1 Ultra Violet Lighting

During the research on measuring the distance to objects with a low cost infra red laser and monochrome camera the problem of the thermal flame became a key issue. It was decided to investigate the use of a monochromatic light source and band pass filter to remove the thermal spraying flame. It was decided to use the UV-A spectrum (350 nm - 400 nm) as an initial area for research because it is reasonable to assume there is the full visible normal lighting (400 nm – 750 nm) and infra red (750 nm – 1 mm) in the thermal spraying scene and environment.

The light source used was a black light fluorescent lamp used in dance halls which has an amount of 387 nm wavelength light which matches our band pass filter.

2.2 Camera and Filter Spectral Response to Ultra Violet

A key aspect of the research was to use standard low cost equipment. The first objective was to ensure that the low cost monochrome camera has a response under ultra violet lighting, as the data sheet did not even provide data below 400 nm (Samsung). A 387 nm narrow band pass filter was used. Figure 2 shows the camera and filters relative spectral responses.

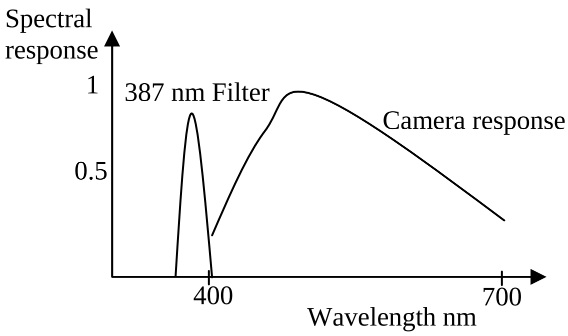


Figure 2: Camera and filter spectral response.

2.3 Ultra Violet Camera and Filter

A small piece of aluminium metal 50 mm x 60 mm with the letters D I T of height 15 mm written on it was used as a test piece. The test piece under internal daylight is shown in Figure 3. The test piece of aluminium with DIT and the background are clear and distinct.

Figure 3: Test piece of Aluminium.

A 387 nm filter was placed in front of the camera under internal daylight and the result is shown in Figure 4. The result shows a complete lack of response from the camera.

Figure 4: Camera response internal daylight and 387 nm filter.

A black light fluorescent lamp, which has a certain amount of 387 nm wavelength light, was then switched on and the cameras response is shown in Figure 5.

Figure 5: Camera response to filtered 387 nm lighting.

Due to the low intensity of 387 nm lighting, the camera was moved closer to the test piece. The background to the test piece is shown as dark stripes to the left and right of the image. The response of the camera clearly shows the letters D I T.

The monochrome image pixels have dynamic range values between 0 and 255. The response of the camera in this experiment provides a low dynamic range image. Using Matlab^{TM} this low dynamic range is shown quantitatively by its histogram in Figure 6. There are no intensity values between 185 and 255, however there is good separation between the letters DIT and the background shown by the dip in the histogram at an intensity value of 133.

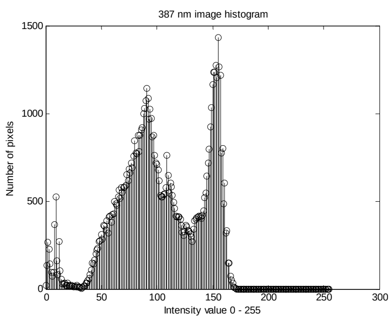


Figure 6: 387 nm image histogram.

The low dynamic range response is due not only to the response of the camera but from the lack of 387 nm intensity in the black light and the 387 nm filters attenuation effect.

2.4 Flame Removal from Image

Using a small butane lighter flame in front of the test piece under daylight lighting produces the image shown in Figure 7.

Figure 7: Daylight with flame.

Clearly image information behind the flame is completely obliterated because of the saturation effects of the flame on the cameras photo sensors, which is shown quantitatively in the images histogram in Figure 8.

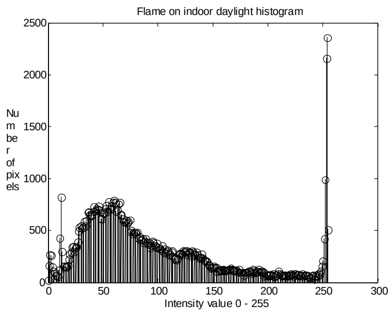


Figure 8: Flame on daylight histogram.

The histogram shows 8.2% of the pixels in the image have what we would consider saturated values between 250 and 255, which are caused by the butane flame. It would be extremely difficult to obtain information from behind the flame such as the area or centroid in pixels of the letter I in this image.

The main developments reported in this paper will detail a process for obtaining this and other information about the letter I, a process which could be developed and applied to the thermal spraying control process.

Placing the 387 nm filter in front of the camera and turning on the black light with the butane flame on produces the image shown in Figure 9.

Figure 9: 387 nm lighting flame on.

This technique was extremely encouraging, as the letters D I T are clearly visible. There is a slight transmission of flame intensity just above the letter I. Letters on the butane torch are also visible. Figure 10. is a histogram of the flame on in the 387nm image.

The histogram for the image shown in Figure 10 suggests this is a low contract image and there is considerable room for improving image information (contrast) above pixel value 185. This could be achieved by increasing the intensity of the 387 nm lighting source.

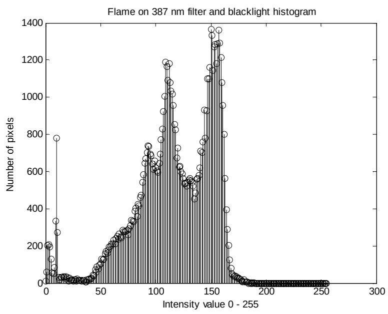


Figure 10: Flame on 387 nm image histogram.

3 IMAGE PROCESSING

3.1 Canny Edge Setection

Using the Matlab™ image processing toolbox the image in Figure 9 was processed using the canny edge detector with a Gaussian filter standard deviation value of 1.5 and high-low threshold values of 0.16 and 0.064 respectively which produced an edges image shown in Figure 11.

Figure 11: Edges image.

The edges image in Figure 11 was image processed further to remove the perimeter objects using the Matlab function `imclearborder` leaving only the letters D I T. Using the Matlab™ functions for labelling, selecting and infilling the letter I, `bwlabel`, `bwselect` and `imfill` the letter I was extracted as shown in Figure 12. For contrast the flame image is shown beside the extracted letter I

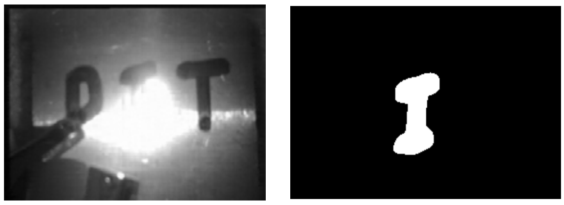


Figure 12: Flame on and extracted letter.

3.2 Feature Extraction

Using the Matlab™ image processing toolbox and the Matlab™ function `regionprops` a number of characteristics for the letter I were obtained. Some of these features are:

- Area 2561 pixels
- Centroid 131, 112 measured from top left corner
- Eccentricity 0.9148
- Orientation 83^0
- Perimeter 264 pixels

From analysis of the above, it is a straightforward process to obtain accurate real world values from image pixel values for actuating a robot manipulator using perspective transformations, inverse kinematics and camera calibration techniques.

4 THERMAL SPRAYING SPECTRA

4.1 Thermal Spraying Process

To determine the band pass filter and lighting wavelength for the removal of the thermal spraying flame and combustion material spectrum in the thermal spraying process would require extensive testing and the purchase of a range of filters. The reason for this is that there are a number of thermal spraying processes such as powder, arc, plasma and a vast range of surface coating materials all producing their own combustion spectra

The following is a list of some of the more common surface coating materials.

- Tungsten carbide/cobalt
- Chromium carbide/nickel chromium
- Aluminium bronze
- Copper nickel indium
- Hard alloys of iron

To apply this technique of using monochromatic ultra violet lighting and narrow band pass filter to remove the combustion process, theoretical research into the spectrum produced by the specific process where autonomous control would be beneficial is required. The reason for this is that the emission spectra of flames is sensitive to (Zirack):

- temperature
- gas/air or gas/oxygen mixture ratio
- gas purity
- burner type
- gas flow (laminar or turbulent)
- coating materials
- height of observation in the flame

Research can however provide reasonable indicators of a location for the band pass filter and where spectral problems may arise. The thermal spraying process used for this research was powder thermal spraying using an Oxy-Acetylene torch.

4.2 Oxy-Acetylene Flame

The Oxy-Acetylene flame is a chemical reaction resulting from the combination of acetylene C_2H_2 with oxygen 0_2. Figure 13 shows the two stages of the chemical reactions (Materials Engineering Group, MEG)

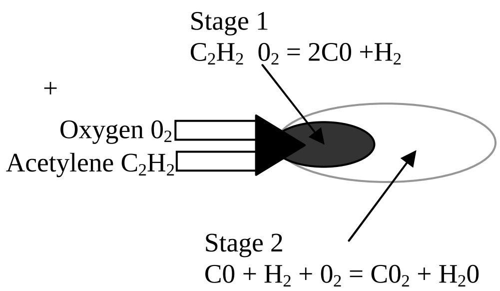


Figure 13: Oxy-Acetylene flame.

A neutral flame with products of combustion CO_2 and H_2O is produced with maximum heat output when equal quantities of oxygen and acetylene are used (MEG). Controlling this mixture would form part of the overall thermal spraying robot control system.

This is an idealised view and many other ordinary molecules and unstable radicals are produced in an Oxy-Acetylene flame in air.

4.3 Oxy-Acetylene Emission Spectra

The visible spectrum runs from 400 nm to 750 nm and the infra red spectrum runs from 750 nm to 1 mm (HyperPhysics). This suggests a portion of the ultra violet spectrum between 350 – 400 nm commonly known as the UV-A spectrum for the research as it excludes the visible and infra red spectrum.

Research is now concentrated on identifying weak spectra between 350 nm and 400 nm from the powder flame spraying Oxy-Acetylene in air flame with a range of molten surface coating materials, which is widely used in the powder spraying industry.

The ordinary molecules which are the stable products of combustion, H_20_2, $C0_2$, C0, 0_2 or N_2 in hydrogen flames do not provide spectra of any appreciable strength in the visible or ultra violet spectrum (Zirack).

The only product of combustion that may have an appreciable spectrum in the UV band is the hydroxyl radical OH which give band peaks at 281 nm 306 nm and 343 nm. Oxyacetylene flames not only produce spectra of hydrogen flames but also emit radiation of hydrocarbon radicals. Between the 350 nm and 400 nm wavelengths a weak CH band occurs at 387/9 nm and a strong band at 432 nm are found in air acetylene flames.

This suggests many wavelengths between 350 and 400 nm may be suitable for removing the Oxy-Acetylene flame in air but we must add the spectrum from the surface coating material to ensure there is no appreciable interference from the molten material in our chosen UV band. This is an area for continued research. However a review of published work by De Saro relating to emission spectra of molten elements such as aluminium and copper provides information on spectra of interest as follows:

- Aluminium 390 – 400 nm
- Iron 260 – 262 nm
- Magnesium 380 - 385 nm
- Copper 320 – 330 nm

Results so far suggest using a narrow band pass filter and lighting between 350 and 370 nm

In addition to the interference from the emission spectra, an added complication is the molten material itself. This will act as a dust cloud and have the effect of reducing contrast in the image.

The image processing techniques necessary for this research are those associated with low contrast images and reconstructing edges and shapes such as those provided by techniques like the Hough transform. (Young)

5 CONCLUSION

This paper has detailed a system of combining optical filtering and image processing which can be used to obtain information about low contrast objects behind or within a test butane flame.

The paper also suggests a region within the UV-A spectrum, which shows promise for implementing ultra violet image control of a thermal-spraying robot. Further work on identifying the spectra of a greater range of surface coating materials is required.

The ability to see through a flame could have benefits in other industries such as the fire fighting service and welding. The system detailed could be fitted as a single eye head up display or fitted to a small mobile robot where there are low smoke flame environments.

REFERENCES

AZoM, Thermal Spraying – An Overview, http://www.azom.com/detailes.asp?AtricleID=511 (last accessed January 2007).

DeSaro R, Weisberg A, Craparo J. 2005, *In Situ, Real Time Measurement of Melt Constituents in the Aluminium, Glass and Steel Industries*, Energy Research Co., Staten Island New York.

England, G.,. *“Wear Resistance”* http://www.gordonengland.co.uk/wear.htm (last accessed January 2007)

HyperPhysics, http://hyperphysics.phyastr.gsu.edu/hbse/ems3.html (last accessed January 2007)

Materials Engineering Group, *Oxy-Gas Welding*, http://www.meg.co.uk/meg/app04.htm (last accessed January 2007)

Matlab^{TM}. http://www.mathworks.com (last accessed january 2007)

Samsung *1/3 CCD image sensor for CCIR cameras*, http://www.datasheetcatalog.com/datasheets_pdf/K/C/7/3/KC73129MP.shtml (last accessed January 2007).

thefabricator.com, *Theraml spray safety and OSHA compliance,* (last accessed January 2007) http://www.thefabricator.com/Safety/Safety_Article.cfm?ID=45

Young, David, 1995, *Hough transforms*,University of Sussex, Brighton, U.K. http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/YOUNG/vision4.html (last accessed January 2007)

Zirak Giorgio, 2000, *Flame Emission Spectroscopy: Fundamentals and Applications*, NILES University of Cairo,Egypt

COMPARISON OF FOCUS MEASURES IN FACE DETECTION ENVIRONMENTS

J. Lorenzo, O. Déniz, M. Castrillón and C. Guerra
Inst. of Intelligent Systems and Numerical Appl. in Engineering (IUSIANI)
University of Las Palmas de Gran Canaria, Campus Univ. de Tafira, 35017 Las Palmas, Spain
Campus Univ. de Tafira, 35017 LAs Palmas, Spain
[jlorenzo,odeniz,mcastrillon,cguerra]@iusiani.ulpgc.es

Keywords: Human-Computer Interaction, Computer Vision, Autofocus measures.

Abstract: This work presents a comparison among different focus measures used in the literature for autofocusing in a non previously explored application of face detection. This application has different characteristics to those where traditionally autofocus methods have been applied like microscopy or depth from focus. The aim of the work is to find if the best focus measures in traditional applications of autofocus have the same performance in face detection applications. To do that six focus measures has been studied in four different settings from the oldest to more recent ones.

1 INTRODUCTION

In face detection and face recognition methods (Pentland et al., 1994; Rowley et al., 1998; Gross et al., 2001; Hjelmas and Low, 2001; Yang et al., 2002; Zhao et al., 2003), borders play an important role because they define the facial features that appear in the face such as eyes, mouth and nose, which are needed to carry out the task. In a blurred image these facial features are not well defined, and so the detection or identification can not be done. Blurred images can be obtained in a defocused camera because defocusing can be modelled as a low-pass filtering process, opposite to focused images which have a higher frequency content (Nayar, 1994). Thus, it is desirable that the image acquisition system has an autofocus mechanism.

Automatic focusing methods fall into two main categories: active and passive systems. Active methods are based on emiting a wave in order to estimate the distance to the object of interest and consequently adjust the lens position. Passive methods estimate the position of the lens by means of finding the position where image sharpness is maximum. Passive focusing methods are normally found in low cost consumer digital cameras, while active focus is widely used in professional digital cameras and computer vision applications such as microscopy or inspection.

As mentioned above, passive autofocusing methods adjust the focus lens to maximize the high frequency components in the image. Therefore a focus value is needed to measure the amount of high frequencies in the image. As autofocusing is a long-standing topic, in the literature a wide variety of focus algorithms have been proposed each one with an associated focus measure (Krotkov, 1987; Nayar, 1994; Lee et al., 1995; Subbarao and Tyan, 1998; Choi and Ki, 1999; Lee et al., 2001; Nathaniel et al., 2001; Kehtarnavaz and Oh, 2003; Kristan and Pernus, 2004; Shirvaikar, 2004; Park and Kim, 2005; Kristan et al., 2006).

Most of the published autofocusing algorithms solve the problem of planar objects like in microscopy applications (Sun et al., 2004) or single object of interest like depth from focus applications (Nayar, 1994). In these applications, focus measures exhibit an ideal curve with a peak with step slopes at the lens position where the object is focused because there is only one object in the image or because it is a planar image. However, in human computer interaction, people do not always hold the same position in the image and exists more objects in the scene so the focus measure does not exhibit a clear maximum. In digital photograhy this drawback is eluded because the photographer selects the object of interest and centers it.

In this work we analyze the performance of some focus measures in different scenarios including typical human computer interaction. These measures are explained in Section 2. Section 3 describes the experiments and finally in Section 4 the results achieved are commented.

2 FOCUS ALGORITHMS

As explained in the previous section, many focus measures have been proposed in the last years to solve the autofocus problem. All of them rely on the fact that a focused image has high contents of higher frequencies so any measure which computes these frequencies can be used. In this work, six of these measures have been chosen to make the comparison. We have compared well known focus measures with more recent ones. Below, we briefly describe each one.

The Tenenbaum Gradient (Tenengrad) (Krotkov, 1987) was one of the first proposed focus measures. This measure convolves the image with vertical (S_x) and horizontal (S_y) Sobel operators. To get a global measure over the whole image, the square of the gradient vector components are summed.

$$F_{Tenengrad} = \sum\sum S_x(x,y)^2 + S_y(x,y)^2 \quad (1)$$

The entropy measure proposed by Firestone et al. (Firestone et al., 1991) is based on the idea that the histogram of a focused image contains more information than the histogram of a defocused one. In this measure the histogram is normalized to get the probability $p(i)$ for each gray level i.

$$F_{Entropy} = -\sum_{intensities} p(i)\log p(i) \quad (2)$$

The Sum of Modified Laplace (SML) (Nayar, 1994) is based on the linear differential operator Laplacian which has the same properties in all directions and is therefore invariant to rotation. Thus, the SML measure sums the absolute values of the convolution of the image with the Laplacian operators.

$$F_{SML} = \sum\sum |L_x(x,y)| + |L_y(x,y)| \quad (3)$$

Energy Laplace (Subbarao and Tyan, 1998) is based on the same idea of the SML mesasure but the image is convolved with the following mask,

$$L = \begin{bmatrix} -1 & -4 & -1 \\ -4 & 20 & -4 \\ -1 & -4 & -1 \end{bmatrix}$$

which computes the second derivate $D(x,y)$ of the image. The value of the focus measure is the sum of the squares of the convolution results.

$$F_{EnergyLaplace} = \sum\sum D(x,y)^2 \quad (4)$$

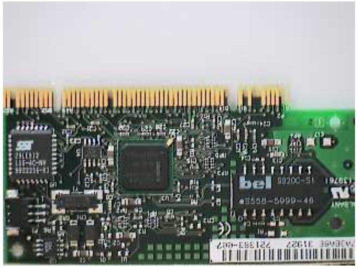


Figure 1: Printed circuit board image.

Figure 2: Box picture.

Nanda and Cutler (Nanda and Cutler, 2001) proposed a focus measure from the contrast of a image as the absolute difference of a pixel with its eight neighbors, summed over all the pixels of the image.

$$F_{Contrast} = \sum\sum C(x,y) \quad (5)$$

where the contrast $C(x,y)$ for each pixel in the gray image $I(x,y)$ is computed as

$$C(x,y) = \sum_{i=x-1}^{x+1}\sum_{j=y-1}^{y+1} |(I(x,y) - I(i,j)|$$

Kristan et al. (Kristan et al., 2006) described M_{Be}, which is one of the most recent focus measures. It is based on the coefficients of the discrete cosine transform obtained after dividing the image into 8x8 non overlapped windows and then averaging over all the 8x8 windows. It must be noticed that in our implementation we have no filtered components corresponding to high order frequencies as Kristan proposes.

$$F_{M_{Be}} = \frac{\sum M'_{Be}}{\text{num. of 8x8 windows}} \quad (6)$$

where M'_{Be} is computed from the DCT coefficients $F(\omega,\nu)$ as

$$M'_{Be} = 1 - \frac{\sum |F(\omega,\nu)|^2}{(\sum |F(\omega,\nu)|)^2}$$

Figure 3: First human computer interaction scenario.

Figure 4: Second human computer interaction scenario.

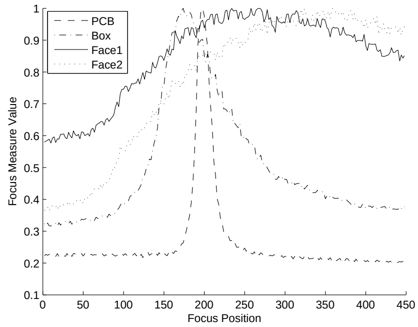


Figure 5: Normalized curves for the Tenengrad focus measure.

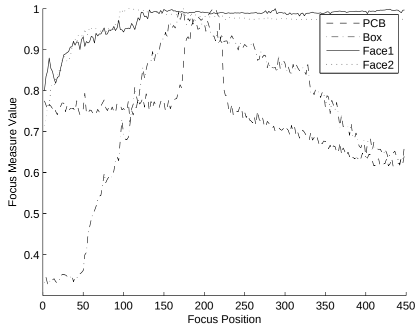


Figure 6: Normalized curves for the entropy focus measure.

3 EXPERIMENTAL STUDY

The images were acquired with a Sony DFW-VL500 firewire color camera with a 16x integrated zoom in an indoor environment and four different settings were analyzed. The first setting corresponds to a printed circuit board which yields a planar image with all the scene elements to the same distance of the camera; we refer to this setting as `PCB` (Fig. 1). The second setting is very common in depth from focus applications where an isolated object appears in the image, this setting will be refered as `Box` (Fig. 2). The third and fourth settings are the ones that we typically found in a human computer interaction application where a person appears either in front of the camera or in an office enviroment. They will be refered as `Face1` (Fig. 3) and `Face2` (Fig. 4).

As the camera has 450 focus positions, 224 images for each of the previosly described settings were acquired with a 2 focus position step. For each acquired image the six focus measures were computed and the criterium to assess the quality of each measure was the similarity of the resulting curve with an "ideal" focus curve which exhibits only a sharp peak.

Figure 5 shows the normalized curves of the Tenengrad focus measure for the four examples. This measure gives good results for `PCB` and `Box` examples because in both cases there is a well defined maximum at positions 198 and 176 respectively. On the other hand in the examples `Face1` and `Face2`, the obtained curve does not show a sharp peak so the maximum search is more difficult.

Figure 6 shows the normalized curves of the entropy focus measure for the four examples. As it is shown in the graphics, the behaviour of this measure is not so good as Tenengrad measure. Entrogy based measure only gives a good focus curve for the `PCB` example with the maximum located at position 216. For `Face1` and `Face2` examples the curve increases its value until it reaches a plateau where a maximum is really difficult to find.

The results of the SML measure in the four examples are shown in Figure 7. This measure exhibits a well defined peak in `PCB` and `Box` with maxima at positions 200 and 180 respectively. In relation to `Face1` example, we get a curve with with a maximum at 246 although the peak is not so sharp as in `PCB` and `Box` examples. In example `Face2` the resulting curve for the SML measure has a flattened peak with the maximum located at 368.

Table 1: Maximum of each focus measure.

Example	$F_{Tenengrad}$	$F_{Entropy}$	F_{SML}	$F_{EnergyLaplace}$	$F_{Constrast}$	$F_{M_{Be}}$	Expert
`PCB`	200	216	200	200	200	200	202
`Box`	176	176	180	176	280	180	178
`Face1`	270	290	246	240	272	264	240
`Face2`	362	108	368	362	334	284	330-370

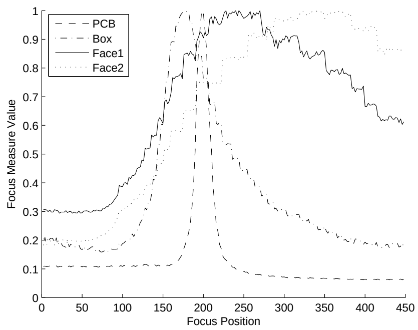


Figure 7: Normalized curves for the Sum of Modified Laplacian focus measure.

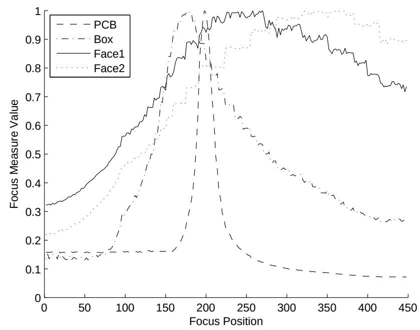


Figure 9: Normalized curves for the focus measure proposed by Nanda and Cutler.

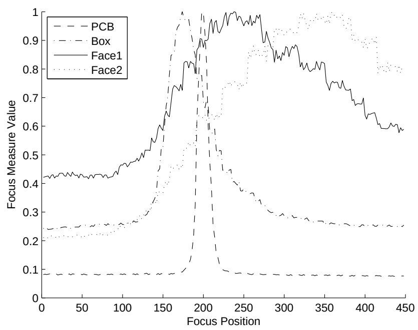


Figure 8: Normalized curves for the Energy Laplace focus measure.

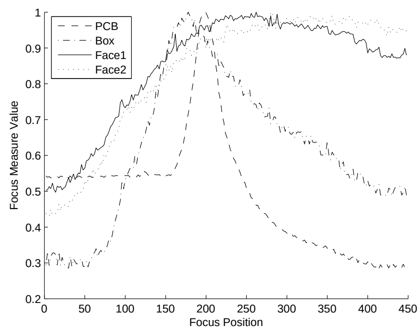


Figure 10: Normalized curves for the M_{Be} focus measure.

Energy Laplace and SML measures are based on the Laplacian of the image because it gives a high response to the higher frequencies in the image. Thus the results we get with Energy Laplace are very similar to those obtained with SML measure as it is shown in Figure 8. For `PCB` example the position of the maximum is the same than in the SML measure. In `Box` example the maximum is at position 176 while for the SML is in position 180, which are very close. The difference in focus position of this measure and SML in examples `Face1` and `Face2` is 6 steps in both cases that confirms the similar behavior of both measures.

As shown in Figure 9, the measure proposed by Nanda and Cutler gives similar results to those obtained with the previously analyzed measures (except entropy) in examples `PCB` and `Box`. For `Face1` and `Face2`, the results are similar too because in neither cases a sharpened peak appears at position of maximum which is located at 272 and 334 respectively.

Finally, the results obtained with the M_{Be} measure are shown in Figure 10. As one of the most recent focus measures it is expected to achieve the best results in the four settings. However, for the `PCB` and `Box` examples, the obtained results are very similar to the other measures even worse than more classical measures as SML or Tenengrad. In `Face1` and `Face2`, the results are not so good as other measures because the curves does not exhibit a sharp peak in the maximum value so more elaborated methods for the maximum location are needed.

The previous analisys only takes into account the shape of the focus curve but to assess the quality of the lens position which corresponds to the maximum it need to be tested by an expert because there is no possibility to compare with a reference focus measure. Sun et al. (Sun et al., 2004) propose as reference to test the accuracy of the compared methods the difference between the lens position given by the method and the lens position selected by an expert.

In this work we have followed a similar approach and for each scenario an expert was asked for getting the most focused image. The range of focus value are shown in the column labelled as `Expert` of table 1. To notice that for the `Face2` scenario the range of focus values in which the image is focused is wider because the person is further and so the depth of field is larger. Comparing in Table 1, the best focus value given by each measure and the one selected by the expert, it is observed that the most accurate measures are those based on Laplacian as SML and Energy Laplace. The others exhibit a similar accuracy except Entropy that as in the previous analysis about the shape of the curve exhibits the worst accuracy.

4 CONCLUSIONS

In this work a comparison of six focus measures have been carried out to investigate the performance of the measures in a face detection application. In face detection applications the person, which is the object of interest, normally is in an office environment so the obtained curves do not exhibit a sharp peak at one defined focus position. Instead, flattened peaks are obtained which make more difficult to get the best focus position. From the six compared focus measures, all of them, except the entropy measure, give very similar results in non face detection applications. In the two face detection settings the best results were obtained with SML and Energy Laplace measures and surprisingly the most recently published measure does not give as good results as previous ones. So we have concluded that for face detection applications the best performance is obtained with Laplacian based measures but it is necessary to use more elaborated maximum finding methods because there does not exist very sharp peaks in the focus curves. Also, a test about the accuracy of the focus position was carried out, using as reference the focus position given by an expert for each example. The results are very similar to the previous given as the two most accurate measures those based on Laplacian and the worst accuracy the Entropy measure.

ACKNOWLEDGEMENTS

This work has been partially supported by the Spanish Ministry of Education and Science and FEDER funds under research project TIN2004-07087.

REFERENCES

Choi, K.-S. and Ki, S.-J. (1999). New autofocusing technique using the frequency selective weighthed median filter for video cameras. *IEEE Transactions on Consumer Electronics*, 45(3):820–827.

Firestone, L., Cook, K., Culp, K., Talsania, N., and Preston, K. (1991). Comparison of autofocus methods for automated microscopy. *Cytometry*, 12:195–206.

Gross, R., Shi, J., and Cohn, J. (2001). Quo vadis face recognition? - the current state of the art in face recognition. Technical Report CMU-RI-TR-01-17, Robotics Institute, Carnegie Mellon University.

Hjelmas, E. and Low, B. K. (2001). Face detection: A survey. *Computer Vision and Image Understanding*, 83(3):236–274.

Kehtarnavaz, N. and Oh, H.-J. (2003). Development and real-time implementation of a rule-based auto-focus algorithm. *Real-Time Imaging*, 9:197–203.

Kristan, M. and Pernus, F. (2004). Entropy based measure of camera focus. In *Proceedings of the thirteenth Electrotechnical and Computer Science Conference ERK*, pages 179–182.

Kristan, M., Pers, J., Perse, M., and Kovacic, S. (2006). A bayes-spectral-entropy-based measure of camera focus using a discrete cosine transform. *Pattern Recognition Letters*, 27(13):1419–1580.

Krotkov, E. (1987). Focusing. *International Journal of Computer Vision*, 1:223–237.

Lee, J.-H., Kim, K.-S., and Nam, B.-D. (1995). Implementation of a passive automatic focusing algorithm for digital still camera. *IEEE Transactions on Consumer Electronics*, 41(3):449–454.

Lee, J.-S., Jung, Y.-Y., Kim, B.-S., and Sung-Jea, K. (2001). An advanced video camera system with robust AF,AE and AWB control. *IEEE Transactions on Consumer Electronics*, 47(3):694–699.

Nanda, H. and Cutler, R. (2001). Practical calibrations for a real-time digital onmidirectional camera. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR 2001)*.

Nathaniel, N., Neow, P., and Ang, M. (2001). Practical issues in pixel-based autofocusing for machine vision. In *Proc. of the 2001 IEEE International Conference on Robotics and Automation*, Seoul, Korea.

Nayar, Shree K.and Nakagawa, Y. (1994). Shape from focus. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(8):824–831.

Park, R. K. and Kim, J. (2005). A real-time focusing algorithms for iris camera recognition. *IEEE Transactions on Systems, Man and Cybernetics*, 35(3):441–444.

Pentland, A., Moghaddam, B., and Starner, T. (1994). View-based and modular eigenspaces for face recognition. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'94)*, Seattle, WA.

Rowley, H. A., Baluja, S., and Kanade, T. (1998). Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):23–38.

Shirvaikar, M. (2004). An optimal measure for camera focus and exposure. In *Proceedings of the Thirty-Sixth Southeastern Symposium on System Theory*, pages 472– 475.

Subbarao, M. and Tyan, J.-K. (1998). Selecting the optimal focus measure for autofocusing and depth-from-focus. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):864–870.

Sun, Y., Duthaler, S., and Nelson, B. J. (2004). Autofocusing in computer microscopy: Selecting the optimal focus algorithm. *Microscopy Research and Technique*, 65:139–149.

Yang, M., Kriegman, D., and Ahuja, N. (2002). Detecting faces in images: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1):34–58.

Zhao, W., Chellappa, R., Rosenfeld, A., and Phillips, P. (2003). Face recognition: A literature survey. *ACM Computing Surveys*, pages 399–458.

OPTIMAL NONLINEAR IMAGE DENOISING METHODS IN HEAVY-TAILED NOISE ENVIRONMENTS

Hee-il Hahn
Dept. Information and Communications Eng. Hankuk University of Foreign Studies, Yongin, Korea
hihahn@hufs.ac.kr

Keywords: Nonlinear denoising, robust statistics, robust estimation, maximum likelihood estimation, myriad filter, Cauchy distribution, amplitude-limited sample average filter, amplitude-limited myriad filter.

Abstract: The statistics for the neighbor differences between the particular pixels and their neighbors are introduced. They are incorporated into the filter to enhance images contaminated by additive Gaussian and impulsive noise. The derived denoising method corresponds to the maximum likelihood estimator for the heavy-tailed Gaussian distribution. The error norm corresponding to our estimator from the robust statistics is equivalent to Huber's minimax norm. This estimator is also optimal in the respect of maximizing the efficacy under the above noise environment. It is mixed with the myriad filter to propose an amplitude-limited myriad filter. In order to reduce visually grainy output due to impulsive noise, Impulse-like signal detection is introduced so that it can be processed in different manner from the remaining pixels. Our approaches effectively remove both Gaussian and impulsive noise, not blurring edges severely.

1 INTRODUCTION

Noise introduced into images via image acquisition devices such as digital cameras can be adequately assumed to be additive zero-mean Gaussian distributed. Such impulsive noise as caused by transmission of images can be more approximated as α stable distribution. In general, the noise with zero-mean and independent properties can be easily removed by locally averaging pixel values. A mean filter is known to be a maximum likelihood estimator for additive Gaussian noise and is optimal in the sense of minimizing mean square error. This filter, however, tends to degrade the sharpness of the boundaries between regions of an image although it effectively removes noise inside the smooth regions. Basically linear filters can not overcome this problem. That is why nonlinear methods should be employed for this purpose. One of the simplest nonlinear filtering algorithms is the median-based filter. It is a maximum likelihood estimator for Laplacian distribution. It has a relatively good property of preserving fine details except for thin lines and corners. It is known to be robust to impulsive noise. Stack filter, weighted median and relaxed median are among its variations to improve the performance. Median-based methods basically select one of the samples in the input window. Thus, it is known that they can not reduce noise effectively. Motivated by the above limitations, several kinds of myriad filters have been proposed, which are known to be maximum likelihood estimator under Cauchy distribution (Gonzalez, Arce, 2001), (Zurbach, et al., 1996). Optimality of myriad filters are presented under α stable distributions (Gonzalez, Arce, 2001). (Hamza and Krim, 2001) proposed mean-relaxed median and mean-LogCauchy filters by combining a mean filter with a relaxed median or LogCauchy filter. They are maximum likelihood estimators under the assumption that the noise probability distribution is a linear combination of normal distribution and heavy-tailed distribution such as Laplacian or Cauchy distribution. Another popular methods are the anisotropic diffusion techniques into which a variety of research has been devoted since the work of (Perona and Malik, 1990). Recent researches have shown that nonlinear methods such as median filters and anisotropic diffusions can be reinterpreted using the theory of robust statistics (Huber, 1981). Robust-statistics-based denoising algorithms are developed, which deal with intensity discontinuities to adapt the analysis window size (Rabie, 2005). He chose a Lorenzian redescending estimator in which the influence function tends to zero with increasing distance.

A large number of image denoising algorithms proposed so far are limited to the case of Gaussian noise or impulsive noise, not to both of them. The algorithms tuned for Gaussian noise or impulsive noise alone present serious performance degradation in case images are corrupted with both kinds of noise. To tackle the problem, an amplitude-limited sample average filter is proposed. It is also a maximum likelihood estimator in the density function which is Gaussian on $(-\delta,\delta)$, but Laplacian outside the region. Its idea is incorporated into the myriad filter to propose an amplitude-limited myriad filter. In order to reduce visually grainy output due to impulsive noise, Impulse-like signal detection is introduced so that it can be processed in different manner from the remaining pixels. Our approaches effectively remove both Gaussian and impulsive noise, not blurring edges severely.

After reviewing the problems of finding the best estimate of a model in terms of maximum likelihood estimate (MLE), given a set of data measurements, our estimators are interpreted based on the theory of robust estimation in both Gaussian and impulsive noise environment.

2 NOISE STATISTICS

In deriving our robust denoising filter, we employ an observed image model corrupted with additive Gaussian and impulsive noise

$$y_i = x_i + n_i, \quad i \in \mathbb{Z}^2 \tag{1}$$

where n_i is a zero-mean additive white Gaussian noise plus impulsive noise. n_i is uncorrelated to the image sequence x_i and y_i is the observed noise-contaminated image sequence. In this case, n_i can be assumed to have a density function whose tails are heavier than Gaussian. To ensure the unbiasedness of the maximum likelihood estimator, its density function is assumed to be symmetric. The density function of n_i is assumed to be Gaussian on $(-\delta,\delta)$, but Laplacian outside the region It has a shape of Gaussian distribution with heavier exponential tails given by

$$f(x) = \begin{cases} Ce^{-a\delta^2/2}e^{-a\delta(x-\delta)}, x > \delta \\ Ce^{-ax^2/2}, -\delta \le x \le \delta \\ Ce^{-a\delta^2/2}e^{a\delta(x+\delta)}, x < -\delta \end{cases} \tag{2}$$

where, of course, C should be chosen so that the density $f(x)$ has unit area by proper adjustment of a and δ. Its statistics can be modelled as symmetric α stable $(S\alpha S)$ distribution.

3 OUR PROPOSED FILTERS

3.1 Amplitude-Limited Sample Average Filter

Let us found out the MLE of the mean of a normal random variable with known variance from M independent observations. The density function for M independent observations is

$$p(\underline{x}/\mu) = \frac{1}{(2\pi)^{M/2}(\sigma^2)^{M/2}} e^{-\frac{1}{2}\sum_{i=1}^{M}(x_i-\mu)^2 / \sigma^2}. \tag{3}$$

The MLE of μ that maximizes the above density function is given by

$$\hat{\mu} = \frac{1}{M}\sum_{i=1}^{M} x_i = \arg\min_{\mu} \sum_{i=1}^{M}(x_i-\mu)^2. \tag{4}$$

The MLE is just the sample mean and $\hat{\mu}$ is known to be a minimum variance unbiased and consistent estimate. This means that the MLE for estimating the signal under the additive Gaussian model is a mean filter. It can be interpreted as optimum filter in the sense of mean-square errors.

Likewise, when the observations have a density of Laplacian instead of Gaussian, the density function for M independent observations is

$$p_L(x/\eta) = \frac{1}{(2)^{M/2}\sigma^M} e^{-\frac{\sqrt{2}}{\sigma}\sum_{i=1}^{M}|x_i-\eta|} \tag{5}$$

and the MLE of η that maximizes the above equation is given by

$$\hat{\eta} = \arg\min_{\eta} \sum_{i=1}^{M}|x_i-\eta| \tag{6}$$

Its MLE corresponds to the median filter which selects the sample located at the center after arranging the observations in the ascending order. Thus, combining the results given in Eq. (4) and (6) we obtain the MLE of θ for the density function given in Eq. (2).

$$\hat{\theta} = \arg\min_{\theta} \left\{ \sum_{|x_i| \le \delta} (x_i - \theta)^2 + \sum_{|x_i| > \delta} |x_i - \theta| \right\} \quad (7)$$

The corresponding filter can be easily implemented by

$$\hat{\theta}_{\delta} = \sum_{i=1}^{M} g(x_i) \quad (8)$$

where $g(x) = \begin{cases} a\delta, ...x > \delta \\ ax, . -\delta \le x \le \delta \\ -a\delta, ...x < -\delta \end{cases}$ (9)

We call this filter as an amplitude-limited sample average filter (ALSAF). The efficacy of the estimate can be found out as follows,

$$\xi = \frac{\left[\int_{-\infty}^{\infty} g'(y) f(y) dy \right]^2}{\int_{-\infty}^{\infty} g^2(y) f(y) dy} \quad (10)$$

In the above equation, $f(x)$, given in Eq.(2), represents the density function for each observation.

Since $g(x) = -\dfrac{f'(x)}{f(x)}$, Efficacy ξ has the maximum value. Thus, the ALSAF given above is the optimal estimate in terms of maximizing the efficacy under the above noise environment. The error norm corresponding to our estimator from the robust statistics is given by

$$\rho(x) = \begin{cases} ax^2/2 ....|x| \le \delta \\ a\delta|x| - a\delta^2/2 ....|x| > \delta \end{cases} \quad (11)$$

This is equivalent to Huber's minimax norm (Huber, 1981), (Black, et al., 1998). To apply our denoising filter, we need to choose the variables a and δ as given in Eq. (2) and Eq. (9), which depends on the statistics of the noisy images. The value of δ is inversely proportional to the amount of outliers such as impulsive noise. If the value of δ is equal to the standard deviation σ of the density function given in Eq. (2), the distribution will be similar to Gaussian, which means that the outliers rarely exist. Thus, δ should be less than σ (typically $\delta = 0.8\sigma$). The probability p_G that the noise is greater than δ is computed as

$$p_G = 2Ce^{-a\delta^2/2} \int_k^{\infty} e^{-a\delta(x-\delta)} dx = \frac{2C}{a\delta} e^{-a\delta^2/2} \quad (12)$$

And the probability p_L that the noise is less than δ is

$$p_L = C \frac{\sqrt{2\pi}}{\sqrt{a}} \left(\Phi\left(\sqrt{a}\delta\right) - \Phi\left(-\sqrt{a}\delta\right) \right) \quad (13)$$

where $\Phi(x) = \dfrac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{y^2}{2}} dy$

a is chosen empirically for each specific image such that $p_G = p_L$ to optimize the estimate. The ALSAF is iteratively applied to reduce any residual noise by estimating the variables a and δ from the statistics of the neighbor differences at each iteration. The algorithm stops when the residual error between the current and the next estimate falls below some threshold at each pixel, which is usually less than δ. Recall the Perona-Malik (PM) anisotropic diffusion (Perona and Malik, 1990)

$$I_t = \vec{\nabla} \cdot \left\{ h\left(|\nabla I|\right) \nabla I \right\} \quad (14)$$

where $\vec{\nabla}$, ∇ denotes divergency and gradient, respectively. Since the robust estimation can be posed as:

$$\min_I \int_{\Omega} \rho\left(|\nabla I|\right) d\Omega \quad (15)$$

where Ω is the domain of the image. Eq. (15) can be solved using the gradient descent as follows:

$$I_t = \vec{\nabla} \cdot \left\{ \rho'\left(|\nabla I|\right) \frac{\nabla I}{|\nabla I|} \right\} \quad (16)$$

Comparing Eq. (14) with Eq. (16), we can obtain the relation

$$h(x) = \frac{\rho'(x)}{x} = \begin{cases} a.................|x| \le \delta \\ ak\dfrac{\operatorname{sgn}(x)}{x}....|x| > \delta \end{cases} \quad (17)$$

Thus, our denoising algorithm can be implemented using PM anisotropic diffusion by selecting the edge stopping function $h(x)$ given in Eq. (17) (Black, et al., 1998).

3.2 Amplitude-Limited Myriad Filter

Similarly, the myriad filter which is the MLE of location under heavier-tailed Cauchy distributed noise is defined as

$$\hat{\beta}_k = \arg\min_{\beta} \prod_{i=1}^{M} \left(k^2 + \left(x_i - \beta \right)^2 \right) \qquad (18)$$

The behavior of the myriad filter is determined by the value of k , which is called the linearity parameter. Given a set of samples $x_1, x_2, \cdots, x_M$, the sample myriad $\hat{\beta}_k$ in Eq. (18) converges to sample mean $\hat{\mu}$ in Eq. (4), as $k \to \infty$ (Gonzalez, Arce, 2001). It is proposed in this paper that outliers which are samples outside the region $(-\delta, \delta)$, are limited, as shown in Eq. (9). That is, the sample myriad is computed as

$$\hat{\gamma}_k = \arg\min_{\beta} \prod_{i=1}^{M} \left(k^2 + \left(g\left(x_i\right) - \gamma \right)^2 \right) \qquad (19)$$

where $g(\cdot)$ is as given in Eq. (9). This filter is named an amplitude-limited myriad filter (ALMF). Its sample myriad $\hat{\gamma}_k$ results in amplitude-limited sample average $\hat{\theta}_\delta$ depicted in Eq. (8), as $k \to \infty$ .This can be easily proved in the same way as the myriad filter converges to a mean filter as $k \to \infty$ as given in (Gonzalez, Arce, 2001).

3.3 Filtering Scheme

As mentioned above, if the given image pixel is known to belong to one of the smooth regions Gaussian noise can be reduced by a mean filter This filter, however, tends to degrade the sharpness of the boundaries between regions of an image if it belongs to the boundary regions. This problem can be reduced effectively by the ALSAF, which however, produces visually grainy output as the amount of impulsive noise increases. Thus, our proposed approach utilizes the statistics of the samples in the window. The parameter k in Eq. (19).is determined according to the presence of impulsive noise in the window.

3.3.1 Processing of Impulsive Noise

Deciding which pixels in an image are replaced with impulsive noise is not clearly defined yet. Especially in cases they are also corrupted with Gaussian noise, the problem will be very complicated. Fortunately, image pixel values does not vary severely from its surrounding pixels even in the boundary regions. Thus, each pixel isolated with its neighbors is detected as an impulse-like pixel.

In order to decide how impulse-like each pixel is, the pixels within the window are arranged in the ascending order for each pixel location, and it is decided whether the pixel is located at some predefined range as given in Eq. (20),

$$D_i = \begin{cases} 0, & x_i \in \left\{ \left[w_i\right]_l, \left[w_i\right]_u \right\} \\ 1, & otherwise \end{cases} \qquad (20)$$

where $\left[w_i\right]_k$ is the k th-order statistics of the samples in the window of size $2N+1$, that is

$$\left[w_i\right]_1 \le \left[w_i\right]_2 \le \cdots \le \left[w_i\right]_{2N+1} \qquad (21)$$

and l and u are such that $1 \le l \le N+1 \le u \le 2N+1$. If D_i corresponding to the pixel x_i equals 0, then the ALSAF or ALMF with a large value of k is applied to the samples in the window because it is more probable the pixel belongs to smooth regions. However, when $D_i = 1$, the pixel is regarded as impulse-like if the mean of absolute values of its neighbour differences (MAD), as given in Eq. (22) is above the predefined threshold,

$$MAD = \sum_{x \in \Omega} \left| x - y \right| \qquad (22)$$

where y is the center pixel and Ω is the set of its neighbors. It is verified experimentally to be a good indicator of impulsive noise. Its idea is borrowed from (Garnett, et al., 2005). Fig. 1 and Fig. 2 depict mean MAD values on whole image pixels as functions of types of noise and its amount. Impulsive noise pixels have much larger mean MAD values than the uncorrupted pixels or the pixels corrupted with Gaussian noise. When impulsive noise exists at some pixel in Lena image, its mean MAD value is

127, which does not vary with the amount of Gaussian noise. In our method, the image pixels whose MAD values exceed 80 are classified as impulsive noise. The pixels decided to be impulse-like are separated to process with an ALMF, whose parameter k as given in Eq. (19) is small. In case there is no impulse within the window, k is set to a large value so that the ALMF may function as an ALSAF.

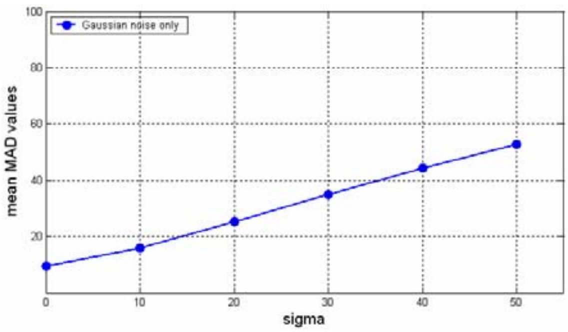


Figure 1: Mean MAD values as a function of standard deviation of Gaussian noise.

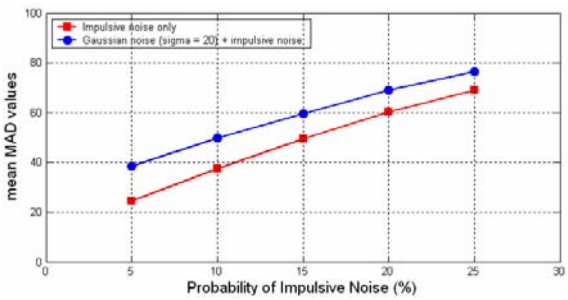


Figure 2: Mean MAD values as a function of probability of impulsive noise.

4 EXPERIMENTAL RESULTS

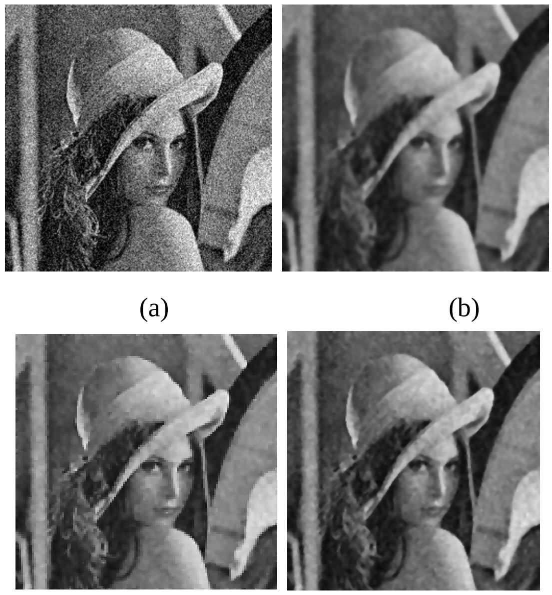
The widely used gray-scale Lena image is selected to test our proposed method. Impulsive noise as well as Gaussian noise are injected to the test image. In other words, the pixel corrupted with Gaussian noise is replaced randomly with impulse, which has the value of 0 ("black") or 255("white") with equal probability. Simulations are carried out for a wide range of noise density levels. The performance of our denoising filter is evaluated by way of mean-square-error (MSE) metric and peak signal-to-noise ratio (PSNR) given by

$$PSNR = 20\log_{10}\left(\frac{255}{\sigma_e}\right) \tag{23}$$

where σ_e is the standard deviation of the residual errors

$$\sigma_e = \sqrt{\frac{1}{|\Omega|}\sum_{i\in\Omega}\left(x_i - \hat{x}_i\right)^2} \tag{24}$$

In the above equation, $|\Omega|$ represents the number of pixels in the image.



(a) (b)

(c) (d)

Figure 3: (a) Corrupted Lena image degraded by Gaussian noise of variance $\sigma_n^2 = 924$, with a measured $PSNR = 18.5dB$ (b) PM anisotropic diffused image after 10 iterations with $\sigma_n^2 = 153.3$ and $PSNR = 26.3dB$ (c) Output of the ALSAF after 10 iterations with $\sigma_n^2 = 137.6$ and $PSNR = 26.8dB$ (d) Output of the ALMF with $\sigma_n^2 = 155.2$ and $PSNR = 26.2dB$.

Fig. 3 shows the simulation results when gray scale image of size 256×256 is corrupted with additive Gaussian noise of variance $\sigma_n^2 = 924$ $(PSNR = 20dB)$. Obviously, both our methods suppress additive Gaussian noise without severely destroying the fine details compared with PM equation in spite of the fact that there are no significant differences in their PSNR values. Simulation results are depicted in Fig. 4 when the Lena image is corrupted with both Gaussian noise of variance $\sigma^2 = 900$ and 10% of impulsive noise $(PSNR = 20dB)$. Simulation results show that the ALSAF is not effective in removing impulsive noise, while the myriad filter can be extended to reduce both Gaussian and impulsive noise by limiting the amplitude of samples outside predefined range as given in Eq. (9). This ALMF tends to preserve the fine details while reducing both Gaussian and impulsive noise.

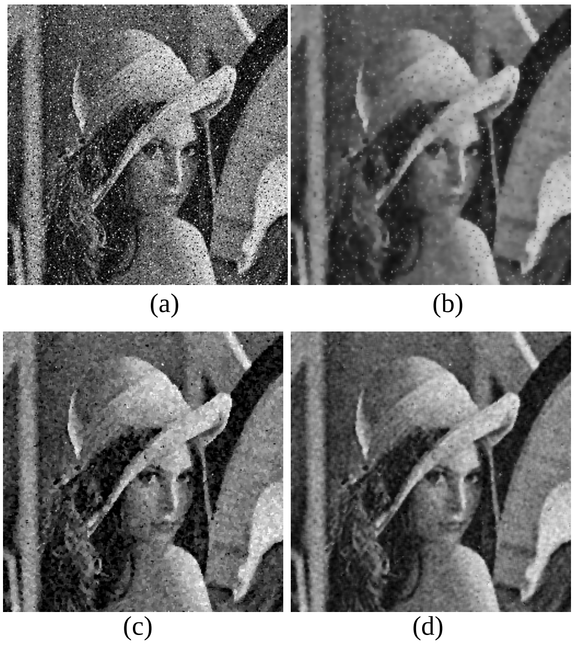


Figure 4: (a) Lena image corrupted with both Gaussian noise of $\sigma = 30$ and impulsive noise of $p = 10\%$, with a measured residual variance $\sigma_n^2 = 2059.3$ and $PSNR = 15.0dB$, (b) Output of the ALSAF after 10 iterations with $\sigma_n^2 = 359.6$ and $PSNR = 22.6dB$ (c) Output of myriad filter with $\sigma_n^2 = 557.9$ and $PSNR = 20.67dB$ (d) Output of ALMF with $\sigma_n^2 = 234.7$ and $PSNR = 24.42dB$.

5 CONCLUSIONS

Optimal nonlinear filter which maximizes the efficacy under mixed Gaussian noise environment is derived. This filter effectively can be implemented using PM anisotropic diffusion by selecting the appropriate edge stopping function. However, it produces visually grainy output as the amount of impulsive noise increases. Thus, impulse-like signal detection is introduced to process impulsive pixels differently from the remaining pixels. For this process, a myriad filter is selected, which is a maximum log-likelihood estimator of the location parameter for Cauchy density. The filter is known to outperform median-based filters in removing impulsive noise. By combining ALSAF which is a MLE in mixed Gaussian noise with a myriad filter, the resulting filter (ALMF) effectively removes both Gaussian and impulsive noise, preserving the fine details.

REFERENCES

Rabie, T., 2005. Robust Estimation Approach for Blind Denoising. IEEE *Trans. Image Process*., vol. 14, No. 11, pp. 1755-1765.

Black, M. J., Sapiro, G., Marimont, D. H., 2005. Robust Anisotropic Diffusion. IEEE *Trans. Image Process*., vol. 14, No. 11, pp.421-432.

Huber, P., 1981. *Robust Statistics*, New York: Wily.

Perona, P., Malik, J., 1990. Scale-Space and Edge Detection Using Anisotropic Diffusion. IEEE *Trans. PAMI*, vol. 12, No. 7, Jul., pp. 629-639.

Hamza, A. B., Krim, H., 2001. Image Denoising: A Nonlinear Robust Statistical Approach. IEEE *Trans. Signal Process*., vol.49, No.12, pp.3045-3054.

Garnett, R., Huegerich, T., Chui, C., He, W., 2005. A Universal Noise Removal Algorithm With an Impulse Detector. IEEE *Trans. Image Process*., vol. 14, No. 11, pp. 1747-1754.

Gonzalez, J. G., Arce, G. R., 2001. Optimality of the Myriad Filter in Practical Impulsive-Noise Environments. IEEE *Trans. Signal Process*., vol.49, No.2, pp. 438-441.

Zurbach, P., Gonzalez, J. G., Arce, G. R., 1996. Weighted Myriad Filters for Image Processing. ICIP96, pp. 726-728.

TELEOPERATION OF COLLABORATIVE MOBILE ROBOTS WITH FORCE FEEDBACK OVER INTERNET

Ivan Petrović
Faculty of Electrical Engineering and Computing, University of Zagreb, Unska 3, Zagreb, Croatia
ivan.petrovic@fer.hr

Josip Babić
Končar - Institute of Electrical Engineering, Baštijanova bb, Zagreb, Croatia
babic.josip@gmail.com

Marko Budišić
Department of Mechanical Engineering, University of California, Santa Barbara, USA
marko.budisic@ieee.org

Keywords: Teleoperation, mobile robots, force feedback.

Abstract: A teleoperation system has been developed that enables two human operators to safely control two collaborative mobile robots in unknown and dynamic environments from any two PCs connected to the Internet by installing developed client program on them and by using simple force feedback joysticks. On the graphical user interfaces, the operators receive images forwarded by the cameras mounted on the robots, and on the the joysticks they feel forces forwarded by developed obstacle prevention algorithm based on the dynamic window approach. The amount and direction of the forces they feel on their hands depend on the distance and direction to the robot's closest obstacle, which can also be the collaborating robot. To overcome the instability caused by the unknown and varying time delay an event-based teleoperation method is employed to synchronize actions of each robot with commands from its operator. Through experimental investigation it is confirmed that developed teleoperation system enables the operators to successfully accomplish collaborative tasks in complex environments.

1 INTRODUCTION

Teleoperation is often employed in controlling mobile robots navigating in unknown and unstructured environments. This is largely because teleoperation makes use of the sophisticated cognitive capabilities of the human operator (Sheridan, 1992), (Murphy and Rogers, 1996). Conventional teleoperated mobile robots rely on visual contact with the operator, either directly or through video transmissions. Guiding such a robot is a formidable task, often complicated by the limited view from the camera. Under such conditions, a human teleoperator must exercise extreme care, especially in obstacle-cluttered environments. In order to increase the system performance and to reduce the operator stress and the task errors, force feedback from the robot to the human operator is usually employed, see e.g. (Sheridan, 1992), (Lee et al., 2002).

With the rapid development of information technology, the Internet has evolved from a simple data-sharing media to an amazing information world where people can enjoy various services, teleoperation beeing one of them. The use of Internet for teleoperation tasks has become one of the hottest topics in robotics and automation, see e.g. (Munir, 2001)–(Lo et al., 2004). On the other hand, the Internet also entails a number of limitations and difficulties, such as restricted bandwidth, arbitrarily large transmission delays, delay jitter, and packet lost or error, all of which influence the performance of Internet-based telerobotics systems. A number of approaches have been proposed to ensure stability of the force feedback loop closed over Internet, were the majority of them are based on passivity theory (Munir, 2001), (Niemeyer, 1996), (Niemeyer and Slotine, 2004) or on the event based action synchronization using a non-time refernce (Wang and Liu, 2005), (K. Brady, 2002), (Luo and Su, 2003). The later approach is used in this paper because of its simplicity and effectiveness.

There are many complicated and sophisticated tasks that cannot be performed efficiently by a single robot or operator, but require the cooperation of number of them. The cooperation of multiple robots and/or operators is particularly beneficial in cases when robots must operate in unknown and dynamic environments. Teleoperation of multiple robots by multiple operators over Internet has been extensively studied for couple of years. A good survey can be found in (Lo et al., 2004).

In this paper we present a teleoperation system that consists of two mobile robots, where one of them serves as *Scout* and the other one as *Manipulator*. Scout explores the remote site and with its sensory information assists the operator controlling the Manipulator, which executes tasks of direct interaction with working environment by using the robot arm mounted on it. In order to guarantee safe robots navigation in dynamic environments and/or at high-speeds, it is desirable to provide a sensor-based collision avoidance scheme on-board the robots. By having this competence, the robots can react without delay on changes in its surrounding. Usually used methods for obstacle prevention are based on virtual forces creation between the robot and the closest obstacle, see e.g. (Borenstein and Koren, 1990) and (Lee et al., 2006), where the force is inversely proportional to the distance between them. In this paper, we propose a new obstacle avoidance algorithm based on the dynamic window (DW) approach presented in (Fox et al., 1997). Main advantage of our algorithm is that it takes robot dynamic constraints directly into account, which is particularly beneficial for safe navigation at high-speeds as the safety margin depends not only on distances between the robot and the nearby obstacles, but also on the velocity of robot motion. The algorithm is implemented on both robots and each robot considers the other one as the moving obstacle.

The paper is structured as follows. In Section 2, we present overview of developed mobile robot teleoperation system. Force feedback loops implementations are described in section 3. Section 4 describes experimental results. We conclude with a summary and a discussion of future work.

2 OVERVIEW OF THE SYSTEM

The teleoperation system considered in this paper is schematically illustrated in Fig. 1. It consists of two mobile robots (Scout and Manipulator) operating in a remote environment and two operator stations with PCs and force feedback joysticks. While the operators' PCs (clients) are directly connected to the Internet, robots' on-board PCs (servers) are connected to Internet via wireless LAN. Each operator controls a single robot and receives visual and other data from both robots.

Mobile robots that we use are *PIONEER 2DX* (Scout) and *PIONEER 3DX* (Manipulator) manufactured by *ActivMedia Robotics*. Scout is equipped with an on-board PC, a ring with sixteen sonar sensors, laser distance sensor and a *Sony EVI-D30* PTZ camera. Manipulator carries an external laptop computer, a ring with sixteen sonars and a *Cannon VC-C50i* PTZ camera. Additionally, a *Pioneer arm* with five degrees of freedom and a gripper is mounted on the Manipulator. Sonars on each robot are used for obstacle detection.

Any personal computer with an adequate Internet connection, a force feedback joystick and developed client application can serve as an operator station. The client application has graphical user interface (GUI) that enables the operator to preset the operation mode (*driving*, *manipulation*, *observation*) and to supervise both mobile robot actions.

Logitech WingMan Force 3D Joystick used in our experiments has two axes on which it can read inputs and generate force: x (stick up-down) and y (stick left-right) and two additional axes that can only read inputs: z (stick twist) and throttle. GUI provides additional operator input, e.g. when switching between operating modes. Both joystick references and GUI input are collectively referred to as *commands*. When client application is actively connected with the mobile robot and *driving* operating mode is chosen on the GUI, joystick x and y axes are used to input desired translational and rotational velocities, respectively. Force feedback is applied on the same two axis, defying commands that would lead robot toward detected obstacles. If *manipulation* operating mode is chosen two joystick buttons are used to chose one of arm's 6 joints, third button sends the arm in its home position and y axis is used to input chosen joint velocity and to display reflected force. In *observation* operating mode joystick x axis is used to tilt the camera, y axis to pan it, throttle is used for adjusting zoom, and one joystick button sends camera to its home position.

Communication between server applications running on the mobile robots' on-board PCs and client applications running on the operators' PCs is initialized and terminated by client applications. In special cases, e.g. when robot is turned off or malfunctioning or in case of communication failure, a server application can refuse or terminate the connection. Communication is implemented using three independent communication modules: control module, image transfer module and info module. Modules are

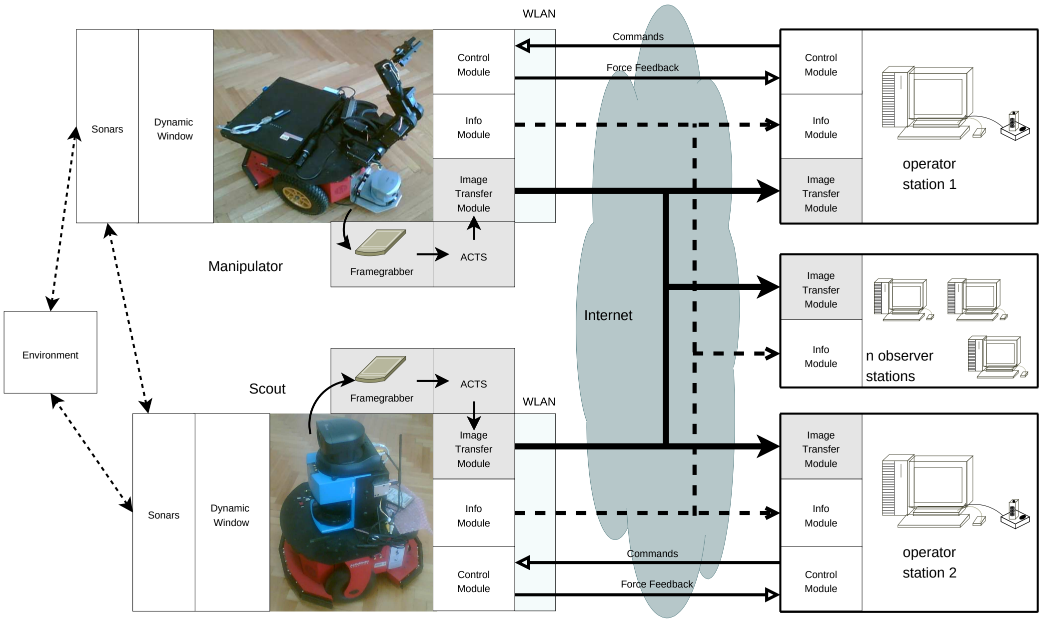


Figure 1: Schematic overview of the teleoperation system.

executed within separate threads of applications and communicate using separate communication sockets. Modular structure decreases individual socket load and enables each module to transfer specific type of data without employing complicated packet scheduling schemes.

Control module is used to transmit commands from the joystick to the robot and force feedback signal from the robot to the joystick. Depending on the operation mode, chosen on the client application GUI, these command can be robot's translational and angular velocities (driving mode), angular velocity of individual joints of the robot arm (manipulation mode) or they could be commands sent to the camera: pan and tilt angular velocity and zoom value (observation mode). If the operator controlls mobile robot's movement or one of robot's arm joints, control module delivers reflected force from the robot to the joystick. In case the operator controlls the camera reflected force is zero and it is only used for action synchronization.

Image transfer module transmits the frames of visual feedback signal from robots' cameras to operators via GUI of the client application. It delivers video signal image at a time. Cameras are connected to the PCs via frame grabber cards and images are fetched using *ActivMedia Color Tracking System* (ACTS) application. ACTS is an application which, in combination with a color camera and frame grabber hardware in a PC, enables custom applications to actively track up to 320 colored objects at a full $30[fps]$ image acquisition rate (ActivMedia, 2003). ACTS serves images to applications connected to it through TCP/IP. This functionality was not used and communication module was developed so that frequency and order in which clients receive images can be directly controlled. Server side communication module periodically requests an image form the ACTS and sends it to the connected clients.

Info module transmits time noncritical information, and is primarily used to synchronize local clocks and transmit information about robot velocities and arm positions to the clients.

3 FORCE FEEDBACK LOOPS

Force feedback from the remote environment gives important information to the human operator. Two force feedback loops have been implemented. One, which is implemented on both mobile robots, forwards the force to the corresponding operator in case of possible collision with the nearby obstacles or with the collaborating robot. Anther one, which is implemented only on Manipulator, forwards the force to its operator's hand when he controls the robot arm.

3.1 Event Based Action Synchronization

Main drawback of closing a control loop over the Internet is the existence of stochastic and unbounded communication delay that can affect system performance and even make it unstable. These problems are usually addressed by ensuring passiveness of the control loop, see e.g. (Munir, 2001), (Niemeyer, 1996) and (Niemeyer and Slotine, 2004) or by using a non-time reference for action synchronization, as presented in (Wang and Liu, 2005), (K. Brady, 2002) and (Luo and Su, 2003). The later approach is used here, because there is no need for additional signal processing and consequently the control system is computationally much simpler. The stability of the control system with event-based action synchronization is ensured if a nondecreasing function of time is used as the action reference (Xi and Tarn, 2000). The number of completed control cycles, which is obviously a monotone increasing function of time, was chosen for this reference. Each control cycle (Fig. 2) is a sequence of the following actions:

1. Server application fetches the most recent force feedback from the buffer and sends it to the client application.
2. Client application receives force feedback.
3. Received force is applied to the joystick.
4. New commands are read from the joystick.
5. The commands are sent to the server application.
6. Server application receives new commands and refreshes the buffer. Depending on the operation mode, DW algorithm or arm controller periodically fetches the command from the buffer and refreshes the force feedback on the buffer after its execution. Commands are refreshed once for every force feedback signal sent to the client application.

Proper order of arrival of information packets is crucial to stability of control system, even more than their timely delivery. For this reason, UDP protocol is used for sending operator commands and force feedback. UDP, unlike TCP, does not resend lost packets and is therefore more suitable for real time applications. Additionally, UDP packets tend to be smaller in size than corresponding TCP packets which yields a smaller load on the communication channel. However, unreliable delivery of control packages could break the control cycle and longer communication delays may destabilize the system. To avoid this, server application monitors control cycle duration and if it exceeds a prescribed maximal value (e.g. 1 second), server application initiates a new control cycle by sending a fresh force feedback packet. All packets that arrive outside their control cycles are simply ignored.

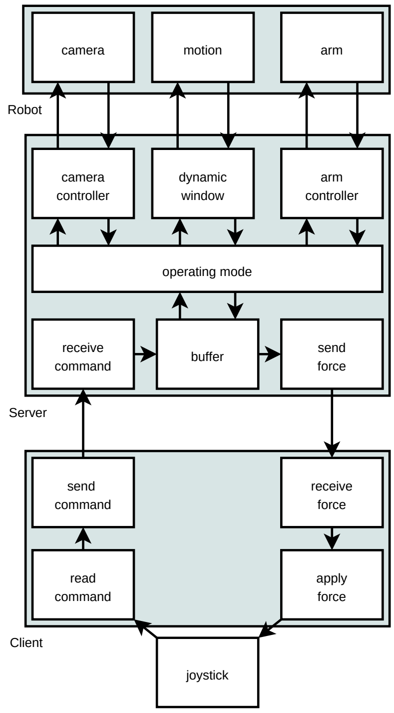


Figure 2: Event-Based Action Synchronization between client and server control modules.

Described event-triggered control ensures that the force applied to the joystick corresponds to the command sent in the previous control cycle and that the buffer state is refreshed with the command that corresponds to the force feedback sent in the current control cycle. Therefore, action synchronization is achieved using the control cycle number as a time independent reference and the system is stable.

Action synchronization between Manipulator and its operator station is executed independently from action synchronization between Scout and its operator station. This arrangement is referred to as decentralized event-based control (Lo et al., 2004) and it assures that control cycle duration of one robot-operator station pair is not affected by communication delays of the other pair.

3.2 Collision Prevention

For safe robots navigation and cooperation in dynamic environments it is necessary to provide a sensor-based collision prevention scheme on-board each mobile robot. Here, we applied the DW algorithm, which is a velocity space based local reactive avoidance technique (Fox et al., 1997). Unlike directional reactive collision avoidance approaches (e.g. potential filed, vector field histograms), the DW algorithm takes robot's kinematic and dynamic constrains directly into account by performing a search in space of translational and rotational velocities. DW produces trajectories that consist of circular and straight line arcs.

The DW algorithm can be integrated with a global path planing algorithm, e.g. FD* algorithm as in (Seder et al., 2005), for executing autonomous tasks in partially unknown environments. While global path planing algorithm calculates optimal path to a specific goal, the DW algorithm takes into account unknown and changing characteristics of the environment based on the local sensory information. We used DW algorithm in a teleoperation system, without global path planing algorithm, just to ensure safe motion of the mobile robot and to help the operator to better perceive obstacles. Therefore some modifications to the original algorithm had to be made.

Operator issues translational and rotational velocities' references. DW algorithm evaluates the given command while taking into account local sonar readings and kinematic and dynamic robot constrains. Commands that are not safe are not executed and force feedback is generated in order to warn the operator.

Proposed DW-based collision prevention algorithm consists of the following steps:

1. Desired velocities (v_d, ω_d) are fetched from the buffer.
2. They are constrained by maximal and minimal achievable velocities:
$$v_d \in [v_{min}, v_{max}],$$
$$\omega_d \in [\omega_{min}, \omega_{max}]. \tag{1}$$
3. Resulting velocities are additionally constrained to values from the set of velocities $V_{nc} = \{v_{nc}, \omega_{nc}\}$ achievable in one robot control cycle.
$$v_d \in v_{nc} = [v_c - T\dot{v}_m, v_c + T\dot{v}_m],$$
$$\omega_d \in \omega_{nc} = [\omega_c - T\dot{\omega}_m, \omega_c + T\dot{\omega}_m], \tag{2}$$
where v_c and ω_c are current velocities, $\dot{v}_m$ and $\dot{\omega}_m$ are maximal accelerations/decelerations and T is robot control cycle duration.
4. Minimal stopping path is calculated. Stopping time and applied translational deceleration must be established first. If condition
$$|\frac{v_d}{\dot{v}_m}| > |\frac{\omega_d}{\dot{\omega}_m}| \tag{3}$$
is satisfied, maximal translational deceleration a_s is applied during stopping time t_s:
$$t_s = |\frac{v_d}{\dot{v}_m}|,$$
$$a_s = \dot{v}_m. \tag{4}$$

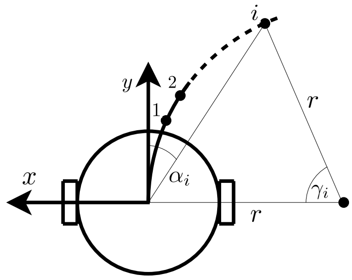


Figure 3: Trajectory of a robot is described as a circular arc.

If (3) is not satisfied it takes more time to stop the rotation of the robot than its translation. Then translational deceleration smaller than maximal is applied:
$$t_s = |\frac{\omega_d}{\dot{\omega}_m}|,$$
$$a_s = |\frac{v_d}{t_s}|. \tag{5}$$
Minimal stopping path s_s is then:
$$s_s = v_d(t_s + t_{stm}) - \frac{a_s t_s^2}{2}, \tag{6}$$
where t_{stm} is additional safety time margin.

5. For the given velocities (v_d, ω_d) coordinates of N_p points on circular or straight line arc (Fig. 3) are calculated using following equations:
$$\gamma_i = 2\alpha_i = 2TN_c\frac{i}{N_p},$$
$$x_i = x_c + sgn(v_d)rsin(\gamma_i),$$
$$y_i = y_c + sgn(v_d)sgn(\omega_d)rcos(\gamma_i), \tag{7}$$
where (x_c, y_c) is the center of the arc, $r = \frac{v_d}{\omega_d}$ its radius, $i = 1, \ldots, N_p$ is the index specifying the point on the trajectory, from 1 to N_p and N_c is the number of cycles for which the algorithm is executed.
6. Minimal allowed distance to obstacle ρ_{min} is:
$$\rho_{min} = r_r + s_{c1} +$$
$$+(s_{c2} - s_{c1})\frac{v_d}{v_{max}} + s_{c\omega}\frac{\omega}{\omega_{max}}, \tag{8}$$
where r_r is robot radius, s_{c1} safety clearance at low translational velocities, s_{c2} safety clearance at high translational velocities and $s_{c\omega}$ rotational safety clearance. Rotational safety clearance is set to $s_{c\omega} = 0$ because it is considered safe to rotate the robot in place. This clearance can be set to a higher value if rotation in place is not considered safe, e.g. when rotating the robot with extended

robot arm. At low translational velocities (i.e. velocities close to zero) s_{c2} contributes little or not at all to ρ_{min} ($\rho_{min} = r_r + s_{c1}, v_d = 0, s_{c\omega} = 0$). At high translational velocities (i.e. velocities close to maximum value) s_{c1} has little or no influence on ρ_{min} value ($\rho_{min} = r_r + s_{c2}, v_d = v_{max}, s_{c\omega} = 0$).

7. For every point on the trajectory (v_d, ω_d), starting with the one closest to the robot, distance to the closest obstacle $\rho_i(v_d, \omega_d)$ is calculated and if the condition:

$$\rho_i(v_d, \omega_d) \geq \rho_{min}(v_d, \omega_d) \tag{9}$$

is true the calculation is executed for the next point on the trajectory. If (9) is satisfied for all N_p points on the trajectory then it is considered clear, force feedback is zero and the algorithm is finished.

8. If the condition (9) is not satisfied, path to the i-th point on the trajectory is calculated:

$$s_p = v_d \frac{i}{N_p} T N_c \tag{10}$$

9. If the path to the i-th point s_p is smaller than stopping path s_s

$$s_p < s_s, \tag{11}$$

i.e. it is possible to stop the robot before it comes closer than ρ_{min} to the obstacle, trajectory is considered safe, force feedback is calculated and the algorithm is finished. Force feedback is calculated as follows:

$$F_{amp} = \beta \sqrt{x_o^2 + y_o^2}, \quad F_{ang} = atan2(y_o, x_o), \tag{12}$$

where F_{amp} and F_{ang} is force feedback amplitude (scaled to $[0, 1]$ with scaling factor β) and direction, respectively, (x_o, y_o) is the closest obstacle position in mobile robots coordinates and *atan2* is arctangent function.

10. If the condition (11) is not met, the investigated trajectory leads to collision, i.e. leads robot closer than ρ_{min} to the obstacle. To prevent collision, desired velocities magnitudes are decreased:

$$v_{d(i+1)} = v_{di} - v_{d1}/N_s, \quad \omega_{d(i+1)} = \omega_{di} - \omega_{d1}/N_s, \tag{13}$$

where $v_{d(i+1)}$ and $\omega_{d(i+1)}$ are velocities for the next iteration of the algorithm, v_{di} and ω_{di} are velocities of the current iteration, v_{d1} and ω_{d1} are original commanded velocities that entered the first iteration of the algorithm and N_s is the number of steps in which velocities are decremented. If the new values are different than zero, algorithm is repeated from step 4) until safe trajectory is found.

3.3 Force Feedback from the Manipulator's Arm

During robot arm movement force is reflected when the joint being controlled rotates to an angle that is less than $10°$ away from its maximum value:

$$F_{arm} = F_{max}(1 - \frac{\xi_{max} - \xi_i}{10}), \quad |\xi_{max} - \xi_i| < 10°, \tag{14}$$

where F_{arm} is the reflected force amplitude corresponding to the current joint angle ξ_i, F_{max} maximal reflected force amplitude, and ξ_{max} maximal joint angle. Feedback force informs the Manipulator's operator that the controlled joint approaches its maximal angle.

4 EXPERIMENTS

In order to validate the developed system a number of experiments have been carried out during which different signals have been recorded. Results of four illustrative experiments are given here.

In the first experiment the operator drove the robot around an obstacle. The commanded velocities on client side (CS), commanded velocities on server side (SS), measured velocities, applied force feedback direction and amplitude were recorded. During the first phase of the experiment (from 0 s to 3 s, Fig. 4) robot moves forward and the obstacle is outside the DW sensitivity region. Measured velocities match the commanded ones and no force is reflected. When the operator starts turning the robot, the obstacle enters DW sensitivity region and force is reflected at $-80°$ informing the operator that the obstacle is on the right from the robot (from 3 s to 6 s), and at the same time DW limits the robot angular velocity, while translational velocity matches the commanded one. At approximately $t = 6s$, the operator stops turning the robot, and the reflected force falls down, DW algorithm limits for a short time first the robot translational velocity and then angular velocity. Then the robot moves away from the obstacle and both velocities match the commanded ones, but the reflected force again slowly increases, which is caused by the next obstacle entering DW sensitivity region.

In the second experiment two operators, located at different places, attempted to drive the Scout and Manipulator robots into a head-on collision. This is the most difficult case for the modified DW algorithm to handle as both robots see the other as a moving obstacle. Velocities of both robots drop rapidly when the

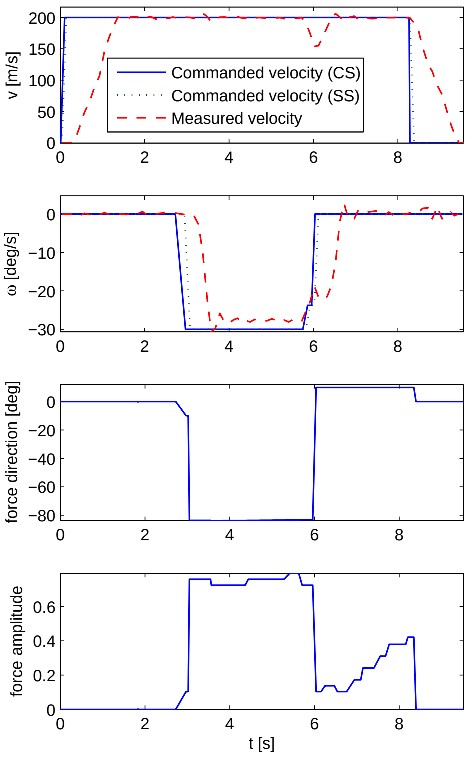


Figure 4: One robot experiment: driving around a corner.

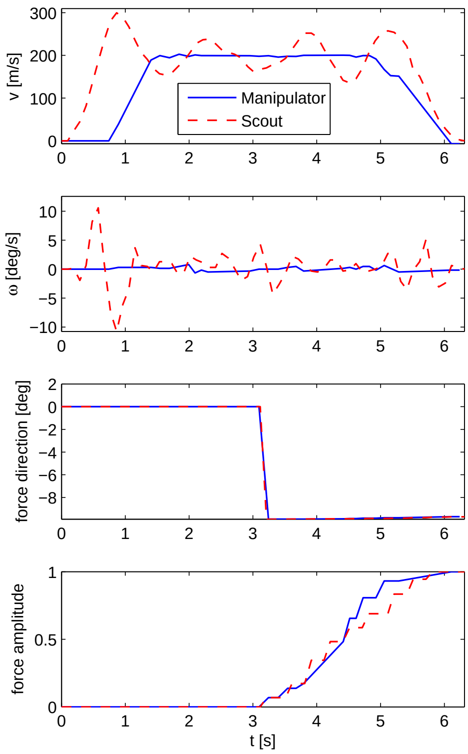


Figure 5: Velocities and force feedback recorded during two robot collision experiment.

distance between them approaches the DW sensitivity range (at approximately $t = 3s$, Fig. 5). At the same time reflected forces sent to both operators rise. When force feedback signals reach their maximal values velocities of both robots drop to zero. Robots stop and the collision is avoided. Fluctuation of the Scouts's rotational and translational velocities is due to difficulties that its low-level velocity control system had while following reference values. In spite of this modified DW algorithm successfully prevented robots from colliding. Force feedback angle should be $0°$ during this experiment as the obstacle is directly in front of the robot. However, this angle varies between $0°$ and $10°$. No stochastic processing was implemented and sonars were treated as rays whose orientation depends on sonar's position on the robot. Such an error is tolerated due to the fact that the force feedback is used only to provide the operator with a general notion about the position and distance to the closest obstacle.

In the last two experiments the task was to pick up a small cardboard box of the floor, place it on the Manipulator's back and put the arm at the home position. Only one operator controlling Manipulator accomplished it after few attempts and with some unnecessary adjustments (Fig. 6). Adjustments were needed as the camera is mounted close to the ground (for better object grasping) not covering the entire workspace of the arm. Another problem is the lack of information about the third dimension which complicates the adjustments of joint angles even when the arm is visible to the operator. The same task can be accomplished much easier and faster (approximately 60 seconds in contrast to approximately 110 seconds) if Scout and Manipulator cooperate (Fig. 7). Scout's assistance gives the Manipulator's operator better view of the distant site enabling him to see the arm during the whole procedure and giving him a feel of the third dimension.

5 CONCLUSION

A teleoperation system has been developed that enables two human operators to safely control two mobile robots in unknown and dynamic environments over the Internet. Each operator receives images displayed on the graphical user interface, which are forwarded by the cameras mounted on the robots, and force feedback on the joystick, which is reflected from

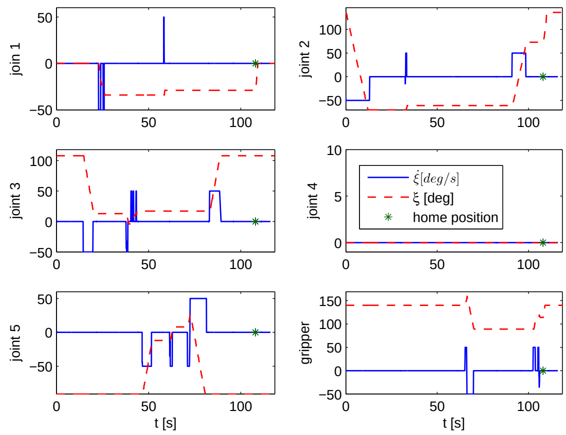


Figure 6: Motion of the robot arm joint angles during the one experiment.

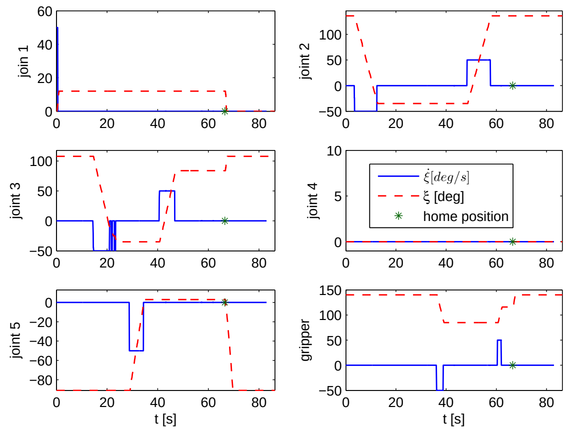


Figure 7: Motion of the robot arm joint angles during the two robots experiment.

the robot controlled by him. To overcome the instability caused by the unknown and varying time delay, event-based teleoperation system is employed to synchronize actions of each robot with command from its operator. Through experimental investigation it is confirmed that developed teleoperation enables the operator to successfully accomplish teleoperation tasks in complex environments.

Developed teleoperation system could be easily adjusted to different robot and sensor types to allow application to different tasks. A possible application could be in missions of finding and rescuing victims from collapsed buildings in cases of natural disasters, fires or terrorist attacks. For example, Scout could be a small flying robot with various sensors that could easily maneuver the site searching for the victims while Manipulator could be stronger robot able to clear its way into the wreckage and carry them out of danger zone.

REFERENCES

ActivMedia (2003). *ACTS ActivMedia Robotics Color Tracking System — User Manual*. ActivMedia Robotics LLC, Amherst.

Borenstein, J. and Koren, Y. (1990). Teleautonomous guidance for mobile robots. In *IEEE Transactions on Systems, Man and Cybernetics*.

Fox, D., Burgard, W., and Thurm, S. (1997). The dynamic window approach to collision avoidance. In *IEEE Robotics & Automation Magazine*.

K. Brady, T. J. T. (2002). *An Introduction to Online Robots*. The MIT Press, London.

Lee, D., Martinez-Palafox, O., and Spong, M. W. (2006). Bilateral teleoperation of a wheeled mobile robot over delayed communication network. In *Proceedings of the 2006 IEEE International Conference on Robotics and Automation*.

Lee, S., Sukhatme, G., Kim, G., and Park, C. (2002). Haptic control of a mobile robot: A user study. In *IEEE/RSJ International Conference on Intelligent Robots and Systems EPFL*.

Lo, W., Liu, Y., Elhajj, I. H., Xi, N., Wang, Y., and Fukada, T. (2004). Cooperative teleoperation of a multirobot system with force reflection via internet. In *TIEEE/ASME Transactions on Mechatronics*.

Luo, R. C. and Su, K. L. (2003). Networked intelligent robots through the internet: issues and opportunities. In *IEEE Proc.*

Munir, S. (2001). *Internet-Based Teleoperation*. PhD thesis, Georgia Instituteof Technology, Atlanta.

Murphy, R. and Rogers, E. (1996). Cooperative assistance for remote robot supervision. In *Presence: Teleoperators and Virtual Environments*.

Niemeyer, G. (1996). *Using Wave Variables in Time Delayed Force Reflecting Teleoperation*. PhD thesis, MIT, Cambridge, MA.

Niemeyer, G. and Slotine, J.-J. E. (2004). Telemanipulation with time delays. In *The International Journal of Robotics Research*.

Seder, M., Maček, K., and Petrović, I. (2005). An integrated approach to real-time mobile robot control in partially known indoor environments. In *IECON*.

Sheridan, T. (1992). *TTelerobotics, Automation, and Human Supervisory Control*. MIT Press, Cambridge, MA.

Wang, M. and Liu, J. N. K. (2005). Interactive control for internet-based mobile robot teleoperation. In *Robotics and Autonomous Systems*.

Xi, N. and Tarn, T. J. (2000). Stability analysis of non-time referenced internet-based telerobotic systems. In *Robotics and Autonomous Systems*.

AN INCREMENTAL MAPPING METHOD BASED ON A DEMPSTER-SHAFER FUSION ARCHITECTURE

Melanie Delafosse, Laurent Delahoche, Arnaud Clerentin
RTeAM IUT Dpt Informatique, Amiens, France
{melanie.delafosse, laurent.delahoche, arnaud.clerentin }@u-picardie.fr

Anne-Marie Jolly-Desodt
GEMTEX, Roubaix, France
anne-marie.jolly-desodt@ensait.fr

Keywords: Mobile robot localization, 2D-mapping, uncertainty and imprecision modeling.

Abstract: Firstly this article presents a multi-level architecture permitting the localization of a mobile platform and secondly an incremental construction of the environment's map. The environment will be modeled by an occupancy grid built with information provided by the stereovision system situated on the platform. The reliability of these data is introduced to the grid by the propagation of uncertainties managed thanks to the theory of the Transferable Belief Model.

1 INTRODUCTION

Localization and mapping are fundamental problems for mobile robot autonomous navigation. Indeed, in order to achieve its tasks, the robot has to determine its configuration in its environment. But, if this result is necessary, it is not sufficient. An estimation of the uncertainty and the imprecision of this position should be determined and taken into account by the robot in order to enable it to act in a robust way and to adapt its behaviour according to these two values.

The two notions of uncertainty and imprecision are distinct ones and they must be clearly defined. The imprecision results from unavoidable imperfections of the sensors, (ie) the imprecision representing the error associated to the measurement of a value. For example, "the weight of the object is between 1 and 1.5 kg" is an imprecise proposition. On the other hand, the uncertainty represents the belief or the doubt we have on the existence or the validity of a data. This uncertainty comes from the reliability of the observation made by the system: this observation can be uncertain or erroneous. In other words, the uncertainty denotes the truth of a proposition. For example, "John is perhaps in the kitchen" is an uncertain proposition.

In a mobile robotics context, these two notions are paramount. Using several tools and several localization algorithms, the mobile robot determines its configuration. Knowing an estimation of the uncertainty and the imprecision of this computed localization, it can adopt an adequate behaviour. For example, if one of these two values is too high, it would try to improve the localization estimation by performing a new localization process.

The key tool used in this purpose is the Transferable Belief Model (TBM) (Smets , 1998), a non-probabilistic variant of the Dempster-Shafer theory (Shafer, 1976). Indeed, this theory enables to easily treat uncertainty since it permits to attribute mass not only on single hypothesis, but also on the union of hypotheses. We can thus express ignorance. So it has enabled us to manage and propagate an uncertainty from low-level data (sensor data) in order to get a global uncertainty about the robot localization. We treat the imprecision independently from the uncertainty because their non-correlation have been proved in (Clerentin and all., 2003)

Our dual approach is particularly adapted to the problem of data integration in an occupancy grid, used as part of SLAM paradigm.

We can principally find two types of mapping paradigm to take into account the notion of distance. The first paradigm consists of computing a cartesian representation of the environment which generally used the Extended Kalman filtering (Leonard and Durrant-Whyte. , 1992). The second approach based on occupancy grid maps allows to manage the metric maps, which were originally proposed in (Elfes,1987.) and which have been successfully employed in numerous mobile robot systems (Boreinstein and Koren , 1991). In (Fox and all,1999) Dieter Fox introduced a general probabilistic approach simultaneously to provide mapping and localization. A major drawback of occupancy grids is caused by their pure sub-symbolic nature: they provide no framework for representing symbolic entities of interest such as doors, desks, etc (Fox and all,1999).

This paper is divided as follows. In a first part, we will detail how our grid occupancy is presented and our uncertain and imprecise sensorial model. Next we will discuss our localization and mapping method based on beacon recognizing . Finally we will present the experimental results.

2 PREAMBLE

2.1 Our Grid Occupancy, Its Initialisation

We choose to model the environment of our mobile platform with the occupancy grid tool in 2D. Thus, the error of sensors measure will be implicitly managed since we will not manipulate a point *(x,y)* but a cell of the grid containing an interval of values *([x],[y])*. We choose to center the grid with the initial position of the platform. Then a cell is defined by its position in the grid . A cell also contains information concerning its occupancy degree by some object of the environment. This latter is defined by a mass function relative to the discernment frame $\Theta 1 = \{yes, no\}$. These two hypotheses respectively correspond to propositions " yes, this cell is occupied by an object of the environment " and " no, this cell is not occupied ". So the mass function of the cell concerning its occupation is composed of the three values in $[0 , 1]$, the mass $m_{cell}(yes)$ on the hypothesis $\{yes\}$, $m_{cell}(no)$ on the hypothesis $\{no\}$ and $m_{cell}(\Theta 1)$ on the hypothesis $\{yes \cup no\}$ representing the ignorance on its occupancy problem. Initially, we have no a-priori knowledge of the situation. So to model our total ignorance, all the cells are initialized with the neutral mass function , that is to say: $m_{cell}\{yes \cup no\} = 1$ and $m_{cell}\{yes\} = m_{cell}\{no\} = 0$.

2.2 Uncertain and Imprecise Sensorial Model

The platform gets a stereovision system composed of two omnidirectionnal sensors (see Figure 1.) distant of about 50 cm. Every acquisition provides two pictures of the environment .

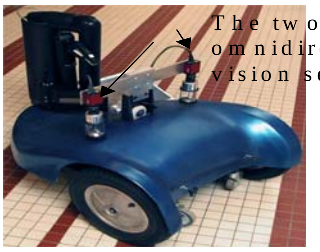


Figure 1: The perception system.

Left sensor	Right sensor

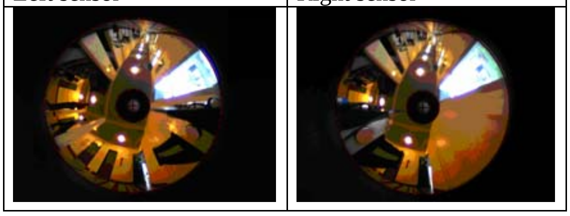


Figure 2 : An example of an acquisition.

On Figure 2 all vertical landmarks of the environment like doors or walls project themselves to the center and form some sectors of different gray levels. The positions of these landmarks will permit to fill the occupancy grid and so to build a map . To get this information, we should associate each sector in the first picture with the one that corresponds to it on the second picture. This stage needs some treatments on the primary data.

First of all, on each omnidirectional picture, we define a signal which represents the mean RGB color from a ring localized around the horizon in the field of view. In fact, what we want to detect are the natural vertical beacons of the environment. Omnidirectional vision system project those vertical parts of the environment according to radial straight lines onto the image. During this computation, it is very important that the rings are centered onto the projection of the revolution axis of the mirror. Otherwise, we will not compute the mean RGB color according to the projection of the vertical elements of the environment. This centering task is automatically done with a circular Hough transform (Ballard,1981). In fact, we look for a circle corresponding to the projection of the black needle situated onto the top of the hyperbolic mirror (see Figure 3) which is situated onto the center of the mirror.

Then, the two 1D mean RGB signals are computed from the ring(Figure 4) and matched together according to a rule of visibility. In fact, if an object is detected from one omnidirectional sensor, it will be visible in a certain area of the other one, according to the distance between the object and the mobile robot.

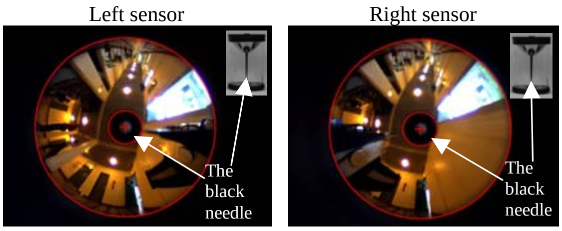


Figure 3: Center location computed with a circular Hough transform.

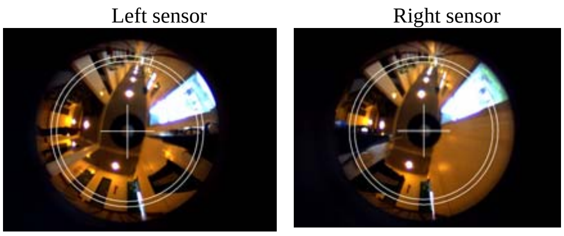


Figure 4: Centered rings to compute the mean RGB signals.

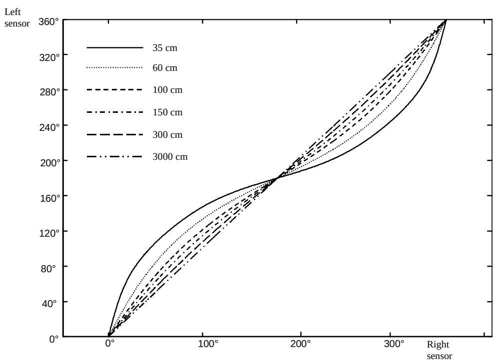


Figure 5: Correspondences between angles from the left sensor to the right sensor for objects situated at different distances from the mobile robot.

Figure 5 shows the correspondences between the angle of the left sensor and the angle of the right sensor according to different distances. We actually notice that the more the object is close to the mobile robot, the more the two angles are different.

The detection algorithm is based upon the derivative of the signal in order to detect sudden changes of color. When we find such value on the left sensor, we look for a similar change in the right sensor signal with a maximum of correlation criteria. In fact, as you can note on the Figure 6, a matching could be close to another one. So, we only keep the most significant matching according to the correlation value.

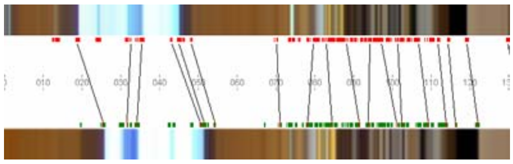


Figure 6 : The two extracted mean color signals from omnidirectional pictures (to 0 from 140°) of Figure 4 and the matching between the left sensor (upper) and the right sensor (bottom).

We choose to use this indicator (in [0-1]) which qualify a very good correlation when is near than 1, to build a degree of uncertainty about the matching by the way of three masses (see Figure 7).

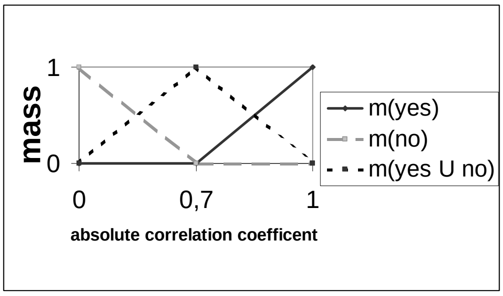


Figure 7: Uncertainty about the matching computed with the correlation coefficient.

When two sectors are matched, we have two pairs of associated angles on the one hand (angles of segments that define borders) and on the other hand a measure of uncertainty on this association. It is directly linked with a landmark which represents it . Therefore the pairs define the position of the landmark and the uncertainty measure its uncertainty on its existence. So this last value is equal to the set of three masses coming from the previous fusion in the discernment frame Θ2= {yes the sectors are associated; no they do not correspond}

- the mass on the hypothesis " yes" $m_{ass}(yes)$
- the mass on the hypothesis " no" $m_{ass}(no)$

the mass on the hypothesis " I can't decide about this matching" $m_{ass}(yes \cup no) = m_{ass}(\Theta 2)$, in other words this mass represents ignorance.

Then the landmark uncertainty is given by the following masses in the discernement frame Θ8= {yes the landmark exist; no it don't exist}:

- the mass on the hypothesis " yes" $m_{land}(yes) = m_{ass}(yes)$
- the mass on the hypothesis " no" $m_{land}(no) = m_{ass}(no)$
- the mass on the ignorance hypothesis ie " I can't decide about the existence "

$m_{land}(\Theta 8) = m_{ass}(\Theta 2)$

Only sectors that have been associated will be used in the continuation of our survey. The measure of uncertainty m_{land} qualifies the landmark but also the segment pairs forming borders. Indeed, the existence of a landmark is linked with the existence of the borders .

Our data are uncertain but they are also imprecise because of sensor measurement errors. This imprecision of measure is managed by the way of intervals. The second result of this fusion, that is to say the matching of two angles (α,β), provides information about extremities (x_i, y_i) of the vertical landmarks in question(Figure 8) thanks to equations of triangulation (1) and (2). So we transform our data in intervals in order to include this imprecision. We create a error domain empirically around our measures of angle (α , β). Then the operations (1) and (2) are computed not between reals but on intervals . We obtain the following equations (3) and (4):

$$x_i = \frac{d \times \tan\beta_i}{\tan\beta_i - \tan\alpha_i} \quad (1)$$

$$y_i = \frac{d \times \tan\beta_i \times \tan\alpha_i}{\tan\beta_i - \tan\alpha_i} \quad (2)$$

$$[x_i] = \frac{d \times \tan[\beta_i]}{\tan[\beta_i] - \tan[\alpha_i]} \quad (3)$$

$$[y_i] = \frac{d \times \tan[\beta_i] \times \tan[\alpha_i]}{\tan[\beta_i] - \tan[\alpha_i]} \quad (4)$$

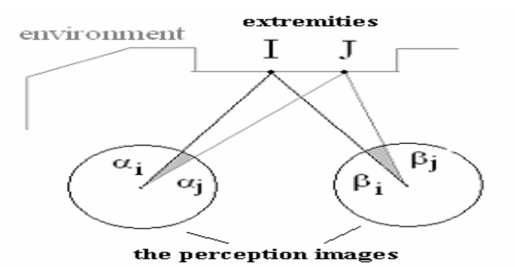


Figure 8 : Sectors matching.

At this level of data exploitation, we have a set of subpaving characterizing the physical extremities of each landmark detected, that is to say the object of which sectors representing it have been matched. These subpavings form the primitive of our sensorial model that we will try to link with the beacons during the time. They are localized by their coordinates *([xi],[yi])* in the frame relative to the platform and they have the same measure of reliability that the landmark ie $m_{prim} = m_{land}$. .

3 LOCALISATION AND MAPPING METHOD

The algorithm consists in matching during the platform displacement the primitives of the sensorial model with information known from the environment that we will call beacons. These matching once achieved will permit both to correct the position of beacons and the estimated position of the platform thanks to the odometry and also to confirm the existence of beacons. In short we will exploit data of these beacons to build our occupancy grid of the surrounding space.

3.1 Definition and Initialisation of Beacon

A beacon is defined by a set of coordinates in the reference frame *(Xe, Ye)* (thus forming a subpaving of localization) and by a degree of uncertainty about its existence composed of three masses as previously shown. This set of masses is established in the discernment frame $\Theta 3=\{yes , no\}$. These two hypotheses respectively correspond to propositions " yes, this beacon exists " and " no, this beacon does not exist". So the function mass concerning its existence is composed of the three values, the mass $m_{bea}(yes)$ on the hypothesis $\{yes\}$, $m_{bea}(no)$ on the hypothesis $\{no\}$ and in short $m_{bea}(\Theta 3)$ on the hypothesis $\{yes \cup no\}$ representing the ignorance about its existence.

A beacon is born from a primitive observed at instant t that cannot be matched with the existing beacons at this instant. This new observation is a landmark not discovered until now or a false alarm. The only information on the existence of a new beacon comes from the existence of the primitive that gave its birth. Then we choose to give the same measure of uncertainty on the beacon, that is to say $m_{bea\ t} = m_{prim}$. Concerning its relative positioning it is equal to the relative localization subpaving of the primitive associated. As thereafter we must associate this beacon to an observation coming from other acquisitions and should use this one in the updating of the occupancy grid. So it is more interesting to manipulate the absolute position . This one is obtained by the change of a frame in relation to the configuration of the platform.

Therefore at each instant, new beacons can appear, and in this case they join the set of the existing beacons to the following acquisition.

3.2 The Association Method between Beacons and Primitives

In looking for these matchings, the aim is on the one hand to get the redundant information permitting to increase the degree of certainty on the existence of the beacons and on the other hand to correct their positioning.

So, at any step, we have several beacons that are characterized by the center of their subpaving *([x],[y])*. Let us call this point the "beacon center". The uncertainty of each beacon is represented by the mass function $m_{bea\,t}$.

In this part, we try to propagate the matchings initialised in the previous paragraph with the observations made during the robot's displacement. In other words, we try to associate beacons with sensed landmarks.

Suppose we manage q beacons at time n. Each beacon is characterized by its "beacon center" (expressed in the reference frame). Let us call this beacon point (x_b, y_b). Suppose the robot gets p observations at time n+1. As we have explained in the previous paragraph, we are able to compute each observation localization subpaving ([x_i], [y_i]) in the reference frame. So, for each observation, we have to search among the q beacons the one that corresponds to it. In other words, we have to match a beacon center (x_b, y_b) with an observation subpaving ([x_i], [y_i]) . The matching criterion we choose is based on the distance between the beacon center and the center of observation subpaving ([x_i], [y_i]).

So at this level, the problem is to match the p observations obtained at acquisition n+1 with the q beacons that exist at acquisition n. To reach this aim, we use the Transferable Belief Model (Smets, 1998) in the framework of extended open word (Shafer, 1976) because of the introduction of an element noted * which represents all the hypotheses which are not modeled, in the frame of discernment.

First we treat the most reliable primitives, that is to say the "strong" primitives by order of increasing uncertainty.

For each sensed primitive *Pj* ($j \in$ [1..p]), we apply the following algorithm:

– The frame of discernment Θ_j is composed of:
  – the q beacons represented by the hypothesis *Qi* ($i \in$ [1..q]). *Qi* means "the primitive *Pj* is matched with the beacon *Qi*")
  – and the element * which means "the primitive *Pj* cannot be matched with one of the q beacons".
  – So: $\Theta_j=\{Q_1, Q_2, \ldots, *\}$
– The matching criterion is the distance between the center of the subpaving of observation *Pj* and one of the beacon centers of *Qi*
– Considering the basic probability assignment (BPA) shown Figure 9, for each beacon *Qi* we compute:
  – $m_i(Qi)$ the mass associated with the proposition "*Pj* is matched with *Qi*".
  – $m_i(\neg Qi)$ the mass associated with the proposition "*Pj* is not matched with *Qi*".
  – $m_i(\Theta_j)$ the mass representing the ignorance concerning the observation *Pi*.
– The BPA is shown on Figure 9.

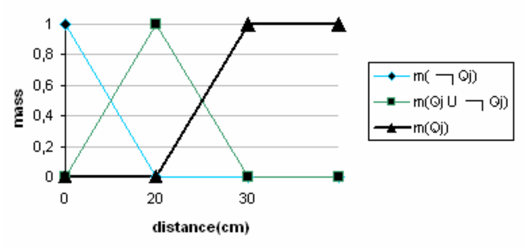


Figure 9: BPA of the matching criterion.

– After the treatment of all the q beacons, we have q triplets :
  – $m_1(Q_1)$ $m_1(\neg Q_1)$ $m_1(\Theta_j)$
  – $m_2(Q_2)$ $m_2(\neg Q_2)$ $m_2(\Theta_j)$
  – ...
  – $m_q(Q_q)$ $m_q(\neg Q_q)$ $m_q(\Theta_j)$

– We fuse these triplets using the disjunctive conjunctive operator built by Dubois And Prade (Dubois and Prade, 1998). Indeed, this operator allows a natural conflict management, ideally adapted for our problem. In our case, the conflict comes from the existence of several potential candidates for the matching, that is to say some near beacons can correspond to a sensed landmark. With this operator, the conflict is distributed on the union of the hypotheses which generate this conflict.

For example, on Figure 10 , the beacon center P_1 and P_2 are candidates for a matching with the primitive subpaving ([x], [y]). So $m_1(P_1)$ is high (the expert concerning P_1 says that P_1 can be matched with ([x], [y])) and $m_2(P_2)$ is high too. If the fusion is performed with the classical Smets operator, these two high values produce some high conflict. But, with the Dubois and Prade operator, the conflict generated by the fusion of $m_1(P_1)$ and $m_2(P_2)$ is rejected on $m_{12}(P_1 \cup P_2)$. This means that both P_1 and P_2 are candidates for the matching.

– So, after the fusion of the q triplets with this operator, we get a mass on each single hypothesis $m_{match}(Qi)$, $i \in$ [1..q], on all the unions of hypotheses $m_{match}(Qi \cup Qj \ldots \cup Qq)$, on the star hypothesis $m_{match}(*)$ and on the ignorance $m_{match}(\Theta_j)$.

– The final decision is the hypothesis which has the maximal pignistic probability (Smets, 1998). If it is the * hypothesis, no matching is achieved. This situation can correspond to two cases: either the primitive *Pj* is an outlier, or *Pj* can be used to initiate a new beacon since any existing track can be associated to it.

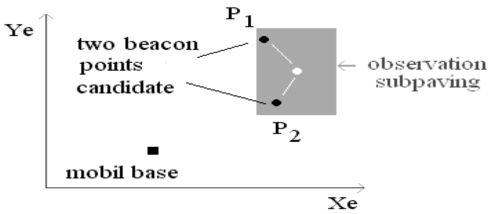


Figure 10: An example of two beacons that generate some conflict.

Once a matching is achieved, the uncertainty of the concerned beacon has to be updated. This uncertainty is denoted by the mass function m_{bea} defined on the frame of discernment Θ_3. This updating has to take the reliability of the matched primitive (mass function m_{prim}) and also the uncertainty of the matching into account. This matching uncertainty is deduced from the pignistic probability of the selected matched primitive by the mass function m2 shown on Figure 11. For example, if the pignistic probability is equal to 0.75, the matching uncertainty is denoted by the following mass function m_2: $m_2(\text{yes})=m_2(\{\text{yes, no}\})=0.5$; $m_2(\text{no})=0$.

Finally, the beacon uncertainty at time t (denoted by the mass function $m_{bea\ t}$) is updated by fusing the beacon uncertainty at the previous time *t*-1, the primitive uncertainty (m_{prim}) and the matching uncertainty (m_2): $m_{bea\ t} = m_{bea\ t\text{-}1} \cap m_{prim} \cap m_2$, where $\cap$ is the fusion operator of Smets.

Let us recall that this mass function is composed of three values: $m_{bea\ t}(\text{yes})$, $m_{bea\ t}(\text{no})$, $m_{bea\ t}(\Theta_3)$.

3.3 The Management of non Associated Beacons

Concerning the beacons that have not been matched at this instant, our first reflection was the following. As no observation can be associated, it implies that our beacon has not been detected to this acquisition. Therefore the first idea was to put in doubt its existence in decreasing its degree of existence. But even if the beacon is no more visible from instant t for example because the mobile platform is moving , the object nevertheless exists. It is necessary not to lose this information at the level of the map. Then we decide not to modify the degree of existence of a beacon which was not matched.

3.4 The Consequences on our Grid

A new beacon or a beacon that have been associated to an observation provide two kinds of information both on the occupied space and on the empty space of our grid. Let us examine the case of one of these beacons at time t to explain this phenomenon. As we have already said, this beacon has a measure of uncertainty on its existence. It is defined by the mass function $m_{bea\ t}$.

3.4.1. The Occupied Space

The existence of a beacon is directly bound to the occupation of cells containing its localization subpaving. Therefore the degree of occupation of these cells must take the degree of existence of the beacon into account. It is achieved thanks to the fusion with the operator of Smets of these two mass functions. So if the mass function of the beacon indicates rather a certain existence then this fusion will increase the degree of occupation of concerned cells. On the contrary, if it indicates an existence which is somewhat unreliable, the fusion will reverberate this doubt on these same cells. A cell is concerned by the fusion if its intersection with the localization subpaving is not empty, they appear in gray on Figure 12a.

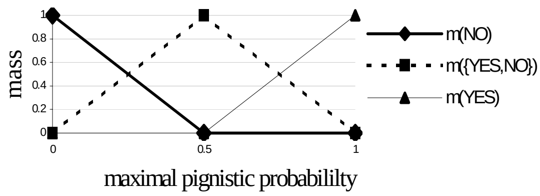


Figure 11: Mass function m2 of the matching uncertainty.

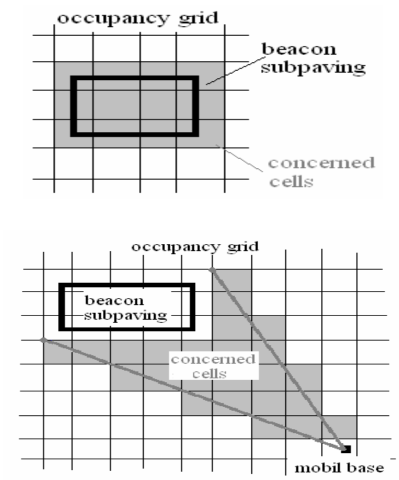


Figure 12: a) Occupied Space, b) Free Space.

The fusion is the following:

$m_{cell\ t} = m_{cell\ t\text{-}1} \cap m_{bea\ t}$

3.4.2 The Free Space

On the other hand, since this beacon has been associated to an observation, it implies that the space between the point of observation in this case the mobile platform and the beacon does not contain any obstacle. This space is therefore free. But it is free in relation with the existence of the beacon.

This operation is achieved in the same way as previously, that is to say merging with the operator of Smets. But this time, we fuse its current occupation degree with a mass function m_3 built as being the "contrary" of the mass function $m_{bea\ t}$. Because the more the beacon is denoted by a high mass on the hypothesis {yes, I exist} , the more the mass on the hypothesis {no, this cell is not occupied} for the cell of the free space (Figure 12 b) must increase . This function m_3 is the next one:

$m_3\{no\} = m_{bea\ t}\{yes\}$, $m_3\{yes \cup no\} = m_{bea\ t}\{yes \cup no\}$ and $m_3\{yes\} = m_{bea\ t}\{no\}$.

And this fusion is given by the following expression : $m_{cell\ t} = m_{cell\ t\text{-}1} \cap m_3$

To resume we get a set of beacons and a occupancy grid of obstacles of the surrounding space.

3.5 The Correction Method

Now we use these data to correct the position of beacons at first and then the estimated position of the mobile platform . These stages are under development. We currently use the correction modules presented below that will be to improve in future works. The beacons are characterized by an error domain of center (x,y). We notice that this center, disposed on the grid, is surrounded with the cells of different occupation levels. To take account of the information we modify the position of the beacon. In fact, we choose the center of gravity of a window 5 x 5 cells around the center pondered by their respective mass $m_{cell}(yes)$, as the point that now characterizes the position of the beacon.

Let us remember that the configuration of the mobile platform is estimated with odometric information. Or we know the classical phenomena of cumulative error if no correction method is achieved (Delafosse and all , 2005). Our correction module is based on the cumulative error minimization. We limit the real possible positions of the platform to centers of cells of a window 3 x 3 around the position estimated by odometry. The kept position among the nine will be the p position that minimizes the accumulated sum of distances between beacons and primitives observed since the p position.

4 EXPERIMENTAL RESULTS

We present experimental results obtained in a structured indoor environment on Figure 13 . The platform is stopped to every stereoscopic acquisition achieved every 30 cms. The managed trajectory is a large boucle represented in yellow on the Figure 14. The natural landmarks mainly observed are the framings of doors, corners , walls and pillars .In Figure 14 we present the obtained map building. The blue cells correspond to the empty space and the red one to the occupied space. The intermediate colors highlight the merging of the occupied and free state. We can notice that the method is robust since most observable landmarks are integrated to the map according to the real map presented in grey on the graphs. We can easily detect for example the corners of the "cross" hall and the free space between each others. We can also notice the certainty of the free space is clearly represented by the color purple. Our approach complementary to the probabilistic one, form an alternative to the SLAM paradigm based on the occupancy grid. We can observe the correlation between the uncertainty of a landmark position estimation and the updating cell values.

5 CONCLUSION

In this article we have presented an architecture of fusion and integration of data for the SLAM paradigm. It is based on a representation of occupancy grid type. The originality of the proposition is on the one hand the propagation of uncertainties on several levels of treatments and on the other hand the management uncoupled of imprecision and uncertainty. The association of these two concepts permits an important reliability in the process of new primitive integrations in the map. This step is crucial since it conditions the global consistency of the cartographic representation on an important number of acquisitions. Moreover our approach permits to solve the problem of "primitive number explosion" which generally implies a divergence of the SLAM process. Besides the precision obtained on the position estimation of observable landmarks is relatively important. So the «symbolic» approach presented constitutes an interesting alternative to methods classically used in this domain that are generally probabilistic.

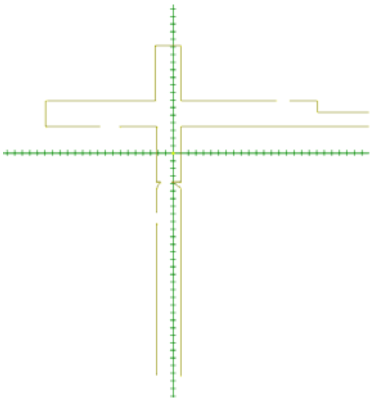


Figure 13: The experimental environment (scale 1m x 1m). We focus on the part of the corridor which represente a cross.

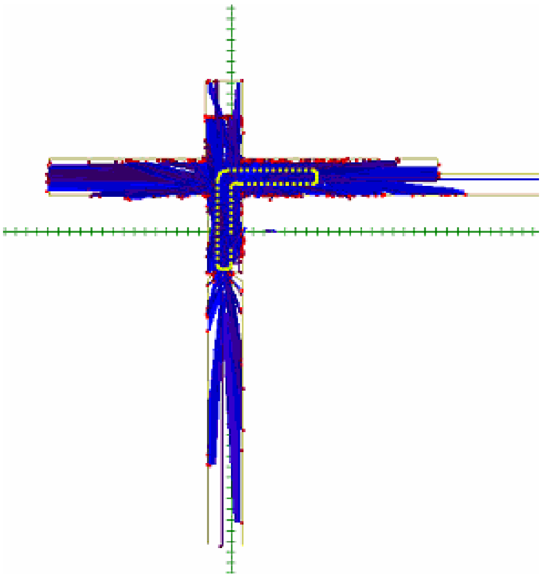


Figure 14 : The resulting map.

REFERENCES

Ballard D. H., "*Generalizing the Hough transform to detect arbitrary shapes." Pattern Recognition*, v. 13, n. 2, pp. 111-122, 1981.

Boreinstein J. , Koren Y., 1991. "*Histogrammic in-motion mapping for mobile robot obstacle avoidance*", IEEE Trans. On rob. and auto., Vol. 7, N°4, pp. 1688-1693

Clerentin A., Delahoche L., Brassart E., Drocourt C., 2003. *"Uncertainty and error treatment in a dynamic localization system"*, proc. of the 11th Int. Conf. on Advanced Robotics (ICAR03), pp.1314-1319, Coimbra, Portugal.

Delafosse M., Clerentin C., Delahoche L., Brassart E., 2005. "*Uncertainty and Imprecision Modeling for the Mobile Robot Localization Problem*" – IEEE international conference on robotics and automation ICRA 05 , Barcelona, Spain.

Dubois D., Prade H., Sandri S., 1998. *Possibilistic logic with fuzzy constants and fuzzily restricted quantifiers.* In: Logic Programming and Soft Computing, (Martin,T.P. et Arcelli-Fontana,F., Eds.), Research Studies Press, Baldock, England, 69-90.

Elfes A. 1987. "*Sonar-based real world mapping and navigation*", IEEE Journal of robotics and automation, Vol. RA-3, N°3, pp. 249-265, June 1987

Fox D., Burgard W. , Thrun S. ,1999. *"Probabilistic Methods for Mobile Robot Mapping"*, Proc. of the IJCAI-99 Workshop on Adaptive Spatial Representations of Dynamic Environments.

Leonard J.J., Durrant-Whyte H.F. , 1992. "*Dynamic map building for an autonomous mobile robot*", The International Journal of Robotics Research, Vol. 11 n°4, August

Royère C, 2002. "*Contribution à la résolution du conflit dans le cadre de la théorie de l'évidence*", thèse de doctorat de l'Université de Technologie de Compiègne.

Shafer GA, 1976. *A mathematical theory of evidence*, Princeton : university press.

Smets Ph ,1998. "*The Transferable Belief Model for Quantified Belief Representation.*", Handbook of Defeasible Reasoning and Uncertainty Management Systems, Kluwer Ed, pp 267-301.

MULTIPLE MODEL ADAPTIVE EXTENDED KALMAN FILTER FOR THE ROBUST LOCALIZATION OF A MOBILE ROBOT

Y. Touati, Y. Amirat, Z. Djama and A. Ali Chérif*
LISSI Laboratory, Paris 12 University, 120-122 rue Paul Armangot
94400 Vitry-sur-Seine, France
touati@univ-paris12.fr, amirat@univ-paris12.fr, djama@univ-paris12.fr, aa@ai.univ-paris8.fr
**Department of Computing, University of Paris 8, 2 rue de la liberté - 93526 Saint-Denis cedex*

Keywords: Mobile robot, Robust Localization, Multiple Model, Hybrid Systems, Kalman Filtering, Data Fusion.

Abstract: This paper focuses on robust pose estimation for mobile robot localization. The main idea of the approach proposed here is to consider the localization process as a hybrid process which evolves according to a model among a set of models with jumps between these models according to a Markov chain. In order to improve the robustness of the localization process, an on line adaptive estimation approach of noise statistics (state and observation), is applied for each mode. To demonstrate the validity of the proposed approach and to show its effectiveness, we've compared it to the standard approaches. For this purpose, simulations were carried out to analyze the performances of each approach in various scenarios.

1 INTRODUCTION

Localization constitutes a key problem in mobile robotics (Borenstein, 1996). It consists of estimating the robot's pose (position, orientation) with respect to its environment from sensor data. Therefore, a better sensory data exploitation is required to increase robot's autonomy. The simplest way to estimate the pose parameters is the integration of odometric data which, however, is associated with unbounded errors, resulting from uneven floors, wheel slippage, limited resolution of encoders, etc. However, such a technique is not reliable due to cumulative errors occurring over the long run. Therefore, a mobile robot must also be able to localize or estimate its parameters with respect to the internal world model by using the information obtained with its external sensors.

The use of sensory data from a range of disparate multiple sensors, is to automatically extract the maximum amount of possible information about the sensed environment under all operating conditions. The main idea of data fusion methods is to provide a reliable estimation of robot's pose, taking into account the advantages of the different sensors (Harris, 1998). The Kalman filter is the best known and most widely applied parameter and state estimation algorithm in data fusion methods (Gao, 2002). Such a technique can be implemented from the kinematic model of the robot and the observation (or measurement) model, associated to external sensors (gyroscope, camera, telemeter, etc.). Basically, the Kalman filter gives a linear, unbiased, and minimum error variance recursive algorithm to optimally estimate the unknown state of a linear dynamic system from Gaussian distributed noisy observations. The Kalman filtering process can be considered as a prediction-update formulation. The algorithm uses a predefined linear model of the system to predict the state at the next time step. The prediction and updates are combined using the Kalman gain which is computed to minimize the Mean Square Error (MSE) of the state estimate. The Extended Kalman Filter (EKF) is a version of the Kalman filter that can handle non-linear dynamics or non-linear measurement equations. Various approaches based on EKF have been developed. These approaches work well as long as the used information can be described by simple statistics well enough. The lack of relevant information is compensated by using models of various processes. However, such model-based approaches require assumptions about parameters which might be very difficult to determine (white Gaussian noise and initial uncertainty over Gaussian distribution). Assumptions that guarantee optimum convergence are often violated and, therefore, the process is not optimal or it can not even converge. In fact, many approaches are based on fixed values of the measurement and state noise covariance matrices. However, such information is not a priori available, especially if the trajectory of the robot is not elementary and if changes occur in the environment. Moreover, it has been demonstrated in the literature that how poor knowledge of noise statistics (noise covariance on state and measurement vectors) may seriously degrade the Kalman filter performance (Jetto, 1999). In the same manner, the filter initialization, the signal-to-noise ratio, the state and observation processes constitute critical parameters, which may affect the filtering quality. The stochastic Kalman filtering techniques were widely used in localization (Gao, 2002) (Chui, 1987) (Arras, 2001)(Borthwick, 1993) (Jensfelt, 2001) (Neira, 1999) (Perez, 1999) (Borges, 2003). Such approaches rely on approximative filtering, which requires ad hoc tuning of stochastic modelling parameters, such as covariance matrices, in order to deal with the model approximation errors and bias on the predicted pose. In order to compensate such error sources, local iterations (Kleeman, 1992), adaptive models (Jetto 1999) and covariance intersection filtering (Julier, 1997)(Xu, 2001) have been proposed. An interesting approach solution was proposed in (Jetto, 1999), where observation of the pose corrections is used for updating of the covariance matrices. However, this approach seems to be vulnerable to significant geometric inconsistencies of the world models, since inconsistent information can influence the estimated covariance matrices.

In the literature, the localization problem is often formulated by using a single model, from both state and observation processes point of view. Such an approach, introduces inevitably modelling errors which degrade filtering performances, particularly, when signal-to-noise ratio is low and noise variances have been estimated poorly. Moreover, to optimize the observation process, it is important to characterize each external sensor not only from statistic parameters estimation perspective but also from robustness of observation process perspective. It is then interesting to introduce an adequate model for each observation area in order to reject unreliable readings. In the same manner, a wrong observation leads to a wrong estimation of the state vector and consequently degrades the performance of localization algorithm. Multiple-Model estimation has received a great deal of attention in recent years due to its distinctive power and great recent success in handling problems with both structural and parametric uncertainties and/or changes, and in decomposing a complex problem into simpler sub-problems, ranging from target tracking to process control (Blom, 1988)(Li, 2000) (Li, 1993)(Mazor, 1996).

This paper focuses on robust pose estimation for mobile robot localization. The main idea of the approach proposed here is to consider the localization process as a hybrid process which evolves according to a model among a set of models with jumps between these models according to a Markov chain (Djama, 1999)(Djama, 2001). A close approach for multiple model filtering is proposed in (Oussalah 2001). In our approach, models refer here to both state and observation processes. The data fusion algorithm which is proposed is inspired by the approach proposed in (Dufour 1994). We generalized the latter for multi mode processes by introducing multi mode observations. We also introduced iterative and adaptive EKFs for estimating noise statistics. Compared to a single model-based approach, such an approach allows the reduction of modelling errors and variables, an optimal management of sensors and a better control of observations in adequacy with the probabilistic hypotheses associated to these observations. For this purpose and in order to improve the robustness of the localization process, an on line adaptive estimation approach of noise statistics (state and observation) proposed in (Jetto, 1999), is applied to each mode. The data fusion is performed by using Adaptive Linear Kalman Filters for linear processes and Adaptive Extended Kalman Filters for nonlinear processes.

The reminder of this article is organized as follows. Section 2 discusses the problem statement of multi-sensor data fusion for the localization of a mobile robot. We develop the proposed robust pose estimation algorithm in section 3 and its application is demonstrated in section 4. Experimental results and a comparative analysis with standard existing approaches are also presented in this section.

2 PROBLEM STATEMENT

This paper deals with the problem of multi sensor filtering and data fusion for the robust localization of a mobile robot. In our present study, we consider a robot equipped with two telemeters placed perpendicularly, for absolute position measurements of the robot with respect to its environment, a gyroscope for measuring robot’s orientation, two drive wheels and two separate encoder wheels attached with optical shaft encoders for odometry measurements (Figure 1). The environment where the mobile robot moves is a rectangular room without obstacles (Figure 2). The aim is not to develop a new method for environment reconstruction or modelling from data sensors; rather, the goal is to propose a new approach to improve existing data fusion and filtering techniques for robust localization of a mobile robot. For an environment with a more complex shape, the observation model, which has to be employed at a given time, will depend on the robot's situation (robot's trajectory, robot's pose with respect to its environment) and on the geometric or symbolic model of environment.

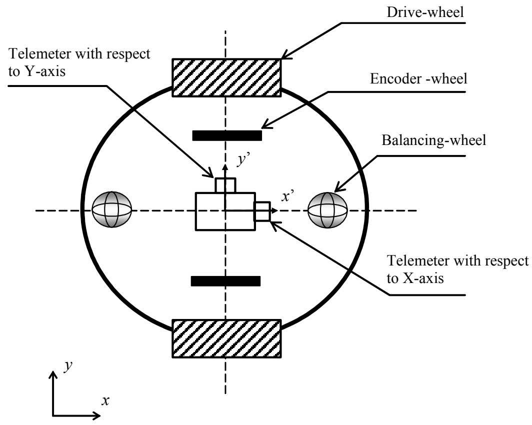


Figure 1: Mobile robot description.

Odometric model: Let $X_e(k)=[x(k)\ y(k)\ \theta(k)]^T$ be the state vector at time k, describing the robot's pose with respect to the fixed coordinate system. The kinematic model of the robot is described by the following equations:

$$x_{k+1} = x_k + l_k \cdot \cos(\theta_k + \Delta\theta_k/2) \quad (1)$$

$$y_{k+1} = y_k + l_k \sin(\theta_k + \Delta\theta_k/2) \quad (2)$$

$$\theta_{k+1} = \theta_k + \Delta\theta_k \quad (3)$$

with: $l_k = (l_k^r + l_k^l)/2$ and $\Delta\theta_k = (l_k^r - l_k^l)/d$. l_k^r and l_k^l are the elementary displacements of the right and the left wheels; d the distance between the two encoder wheels.

Observation model of telemeters: As the environment is a rectangular room, the telemeters measurements correspond to the distances from the robot location to walls (Fig. 2.).

Then, the observation model of telemeters is described as follows:

for $0 \le \theta(k) < \theta^l$:
$d(k) = (d^x - x(k))/\cos(\theta(k))$ with respect to X axis (4)

for $\theta^l \le \theta(k) \le \theta^m$:
$d(k) = (d^y - y(k))/\sin(\theta(k))$ with respect to Y axis. (5)

with:
- d^x and d^y, respectively the length and the width of the experimental site;
- θ^l and θ^m, respectively the angular bounds of observation domain with respect to X and Y axes;
- $d(k)$ is the distance between the robot and the observed wall with respect to X or Y axes at time k .

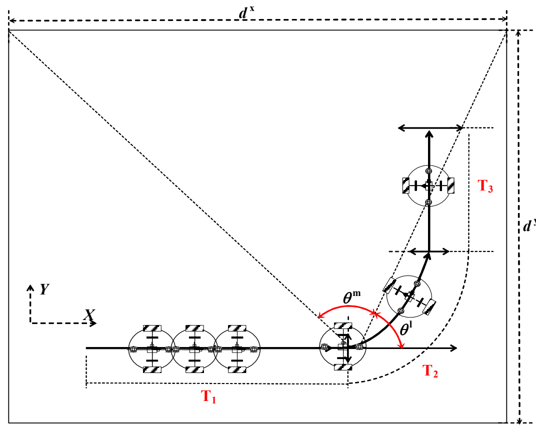


Figure 2: Telemeters measurements –Nominal trajectory composed of sub trajectories T1-T2 and T3.

Observation model of gyroscope: By integrating the rotational velocity, the gyroscope model can be expressed by the following equation:

$$\theta_l(k) = \theta(k) \quad (6)$$

Each sensor described above is subject to random noise. For instance, the encoders introduce incremental errors (slippage), which particularly affect the estimation of the orientation. For a telemeter, let's note various sources of errors: geometric shape and surface roughness of the target, beam width. For a gyroscope, the sources of errors are: the bias drift, the nonlinearity in the scale factor and the gyro's susceptibility to changes in ambient temperature. So, both the odometric and observation models must integrate additional terms representing these noises. Models inaccuracies induce also noises which must be taken into account. It is well known that the odometric model is subject to inaccuracies caused by factors such as: measured wheel diameters, unequal wheel-diameters, trajectory approximation of robot between two consecutive samples. These noises are usually assumed to be Zero-mean white Gaussian with known covariance. This hypothesis is discussed and reconsidered in the proposed approach. Besides, an estimation error of orientation introduces an ambiguity in the telemeters measurements (one telemeter is assumed to measure along X axis while it is measuring along Y axis and vice-versa). This situation is particularly true when the orientation is near angular bounds θ^l and θ^m. This justifies the use of multiple model to reduce measuring errors and efficiently manage robot's sensors. For this purpose, we have introduced the concept of observation domain (boundary angles) as defined in equations (4) and (5).

3 ROBUST MULTIPLE MODEL FILTERING APPROACH

In this section, we present the data fusion and filtering approach for the localization of a mobile robot. In order to increase the robustness of the localization and as discussed in section 2, the localization process is decomposed into multiple models. Each model is associated with a mode and an interval of validity corresponding to the observation domain; the aim is to consider only reliable information by filtering erroneous information. The localization is then considered as a hybrid process. A Markov chain is employed for the prediction of each model according to the robot mode. The multiple model approach is best understandable in terms of stochastic hybrid systems. The state of a hybrid system consists of two parts: a continuously varying base-state component and a modal state component, also known as system mode, that may only jump among points, rather than vary continuously, in a (usually discrete) set. The base state components are the usual state variables in a conventional system. The system mode is a mathematical description of a certain behavior pattern or structure of the system. In our study, the mode corresponds to the robot's orientation. In fact, the latter parameter governs the observation model of telemeters along with observation domain. Other parameters, like velocity or acceleration, could also be taken into account for mode's definition. Updating of mode's probability is carried out either from a given criterion or from given laws (probability or process). In this study, we assume that each Markovian jump (mode) is observable (Djama, 2001)(Dufour, 1994). The mode is observable and measurable from the gyroscope.

3.1 Multiple Model Formulation

Let us consider a stochastic hybrid system. For a linear process, the state and observation processes are given by:

$$X_e(k/k-1,\alpha_k) = A(\alpha_k)\cdot X_e(k-1/k-1,\alpha_k) + B(k,\alpha_k)\cdot U(k-1,\alpha_k) + W(k,\alpha_k) \quad (7)$$

$$Y_e(k,\alpha_k) = C(\alpha_k)\cdot X_e(k/k-1,\alpha_k) + V(k,\alpha_k) \quad (8)$$

For a nonlinear process, the state and observation processes are described by:

$$X_e(k/k-1,\alpha_k) = F(X_e(k-1/k-1),\alpha_k,U(k-1)) + W(k,\alpha_k) \quad (9)$$

$$Y_e(k,\alpha_k) = G_e(X_e(k/k-1),\alpha_k) + V(k,\alpha_k) \quad (10)$$

where: X_e is the base state vector;
Y_e is the noisy observation vector;
U is the input vector;
α_k is the modal state or system mode at time k, which denotes the mode during the k[th] sampling period;
W and V are the mode-dependent state and measurement noise sequences, respectively.

The system mode sequence $\langle \alpha_k \rangle$ is assumed for simplicity to be a first-order homogeneous Markov chain with the transition probabilities:

$$P\{\alpha_{k+1}^j \mid \alpha_k^i\} = \pi_{ij} \qquad \forall \alpha_i,\alpha_j \in S$$

where α_k^j denotes that mode α_j is in effect at time k and S is the set of all possible system modes, called mode space.

The state and measurement noises are of Gaussian white type. In our approach, the state and measurement processes are assumed to be governed by the same Markov chain. However, it's possible to define differently a Markov chain for each process. The Markov chain transition matrix is stationary and well defined.

3.2 Variance Estimation Algorithm

It is well known that how poor estimates of noise statistics may lead to the divergence of Kalman filter and degrade its performance. To prevent this divergence, we apply an adaptive algorithm for the adjustment of the state and measurement noise covariance matrices.

a. Estimation of measurement noise variance

Let $R = \left(\sigma_{v,i}^2(k)\right)\ (i = 1 : n_0)$, be the measurement noise variance at time k for each component of the observation vector. n_0 denotes the number of observers (sensors number).

Let $\hat{\beta}(k)$ the squared mean error for stable measurement noise variance:

$$\hat{\beta}(k) = \frac{1}{n}\sum_{j=0}^{n}\gamma_i^2(k-1) \tag{11}$$

where $\gamma(k)$ represents the innovation.

For $n+1$ samples, the variance of $\hat{\beta}(k)$ can be written as:

$$E\left(\hat{\beta}(k)\right) = \frac{1}{n+1}\sum_{j=0}^{n}\begin{pmatrix} C_i(k-j)\cdot P(k-j,k-j-1)\cdot \\ C_i(k-j)^T + \sigma_{v,i}^2 \end{pmatrix} \tag{12}$$

Then, we obtain the estimation of the measurement noise variance:

$$\hat{\sigma}_{v,i}^2 = \max\left\{\frac{1}{n}\sum_{j=0}^{n}\begin{pmatrix} \gamma_i^2(k-j) - \frac{n}{n+1}\cdot C_i(k-j)\cdot \\ P(k-j,k-j-1)\cdot C_i(k-j)^T \end{pmatrix}, 0\right\} \tag{13}$$

The restriction with respect to zero is related to the notion of variance.

A recursive formulation of the previous estimation can be written:

$$\hat{\sigma}_{v,i}^2(k) = \max\left\{\hat{\sigma}_{v,i}^2(k-1) + \frac{1}{n}\cdot\begin{pmatrix} \gamma_i^2(k) \\ -\gamma_i^2(k-(n+1)) \\ -\frac{n}{n+1}\cdot\Psi \end{pmatrix}, 0\right\} \tag{14}$$

where:

$$\begin{aligned}\Psi = {} & C_i(k)\cdot P(k,k-1)\cdot C_i(k)^T - C_i(k-(n+1))\cdot \\ & P(k-(n+1),k-(n+1)-1)\cdot C_i(k-(n+1))^T\end{aligned} \tag{15}$$

b. Estimation of state noise variance

To estimate the state noise variance, we use the same principle as in subsection a. One can write:

$$\hat{Q}_e(k) = \hat{\sigma}_{n,i}^2(k)\cdot Q_d \tag{16}$$

By assuming that noises on the two encoder wheels measurements obey to the same law and have the same variance, the estimation of state noise variance can be written:

$$\hat{\sigma}_{n,i}^2(k) = \max\left\{\begin{matrix} \dfrac{\gamma_i^2(k-1) - C_i(k+1)\cdot P(k+1,k)\cdot C_i(k+1)^T - \hat{\sigma}_{v,i}^2(k+1)}{C_i(k+1)\cdot Q_d\cdot C_i(k+1)^T}, \\ 0 \end{matrix}\right\} \tag{17}$$

with:

$$\hat{Q}_d(k) = B(k)\cdot B(k)^T \tag{18}$$

By replacing the measurement noise variance by its estimate, we obtain a mean value given by the following equation:

$$\hat{\sigma}_n^2(k) = \max\left\{\frac{1}{(m+1)\cdot n_0}\sum_{j=1}^{m}\sum_{i=1}^{n_0}\hat{\sigma}_{n,i}^2(k-j), 0\right\} \tag{19}$$

Where m represents the sample number.

The algorithm described above carries out an on line estimation of state and measurement noise variances. Parameters n and m are chosen according to the number of samples used at time k. The noises variances are initialized from an "*a priori"* information and then updated on line. In this approach, variances are updated according the robot's mode and the measurement models.

For an efficient estimation of noise variances, an ad hoc technique consisting in a measure selection is employed. This technique consists of filtering unreliable readings by excluding readings with weak probability like the appearance of fast fluctuations. For instance, in the case of Gaussian distribution, we know that about 95% of the data are concentrated in the interval of confidence $[m-2\sigma, m+2\sigma]$ where m represents the mean value and σ the variance.

The sequence in which the filtering of the state vector components is carried out is important. Once the step of filtering completed, the probabilities of each mode are updated from the observers (sensors). One can note that the approach used here is close, on one hand, to the Bayesian filter by the extrapolation of the state probabilities, and on the other to the filter with specific observation of the mode.

4 IMPLEMENTATION AND SIMULATION RESULTS

The approach described above for robust localization was applied for the mobile robot described in section 2. The nominal trajectory of the mobile robot includes three sub trajectories T1, T2 and T3, defining respectively a displacement along X axis, a curve and a displacement along Y axis (Fig. 2.). Note that the proposed approach remains valid for any type of trajectory (any trajectory can be approximated by a set of linear and circular sub trajectories). In our study, we have considered three models. This number can be modified according to the environment's structure, the type of trajectory (robot rotating around itself, forward or backward displacement, etc.) and to the number of observers (sensors). Notice that the number of models (observation and state) has no impact on the validity of the proposed approach.

To demonstrate the validity of the proposed approach (noticed AMM for Adaptive Multiple-Model) and to show its effectiveness, we've compared it to the following standard approaches: Single-Model based EKF without estimation variance (noticed SM), single-model based IEKF (noticed SMI). For this purpose, simulations were carried out to analyze the performances of each approach in various scenarios.

For sub trajectories T1 and T3, filtering and data fusion are carried out by iterative linear Kalman filters due to linearity of the models, and for sub trajectory T2, by iterative and extended Kalman filters. The observation selection technique is applied for each observer before the filtering step in order to control, on one side, the estimation errors of variances, and on the other, after each iteration, to update the state noise variance. If an unreliable reading is rejected at a given filtering iteration, this has for origin either a bad estimation of the next component of the state vector and of the prediction of the corresponding observation, or a bad updating of the corresponding state noise variance. The iterative filtering is optimal when it is carried out for each observer and no reading is rejected. In the implementation of the proposed approach, the state noise variance is updated, for a given mode i, is carried out according to the following filtering sequence: x, y and then θ.

Notation:
- εx, εy and $\varepsilon\theta$: the estimation errors corresponding to x, y and θ respectively;
- Ndx, Ndy and $Nd\theta$: the percentage of selected data for filtering, corresponding to components x, y and θ respectively;
- $Ndxe$, $Ndye$ and $Nd\theta e$: the percentage of selected data for estimation of the variances of state and measurement noises, corresponding to components x, y and θ respectively.
-+:SMI; °: SMI, --:AMM

Scenario 1
-Noise-to-signal Ratio of odometric sensors: right encoder: 8%, left encoder: 8%
-Noise-to-signal Ratio of Gyroscope: 3%
-Noise-to-signal Ratio of telemeter 1: 10% of the odometric elementary step
-Noise-to-signal Ratio of telemeter 2: 10% the odometric elementary step
-"A priori" knowledge on the variance in initial state: Good
-"A priori" knowledge on noise statistics (measurement and state variances): Good

In this scenario, the telemeters measurement noise is higher than state noise. We notice that performances of AMM filter are better that those of SM and SMI filters concerning x and y-components (Table 1; Fig. 3-5). In sub trajectory T3, the orientation's estimation error relating to AMM filter (Table 1) has no influence on filtering quality of the remaining components of state vector. Besides, one can note that this error decreases in this sub trajectory (Figure 6). In this case, only gyroscope is used for the prediction and updating the Markov chain probabilities. In sub trajectory T2, we notice that the estimation error along x-Axis for AMM filter is lightly higher than those relating to other filters. This error is concentrated on first half of T2 sub trajectory (Figure 7) and decreases then on second half of the trajectory. This can be explained by the fact that on one hand, the estimation variances algorithm rejected 0.7% of data, and on the other, the filtering step has rejected the same percentage of data. This justifies that neither the variances updating, nor the x-coordinate correction, were carried out.

Note that unlike filters SM and SMI, filter AMM has a robust behavior concerning pose estimation even when the signal-to-noise ratio is weak. By introducing the concept of observation domain for observation models, we obtain a better modeling of observation and a better management of robot's sensors. The last remark is related to the bad performances of filters SM and SMI when the signal-to-noise ratio is weak. This ratio degrades the estimation of the orientation angle, observation matrices, Kalman filter gain along with the prediction of the observations.

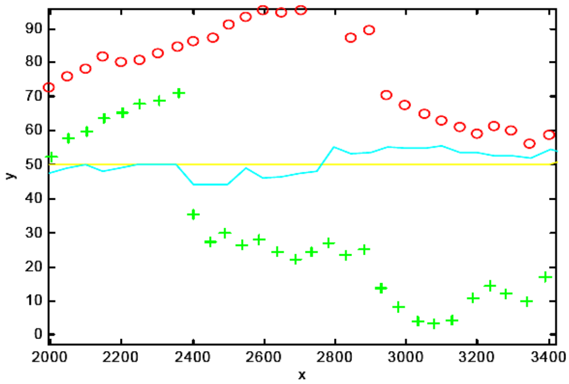


Figure 3: Estimated trajectories (sub trajectory T1).

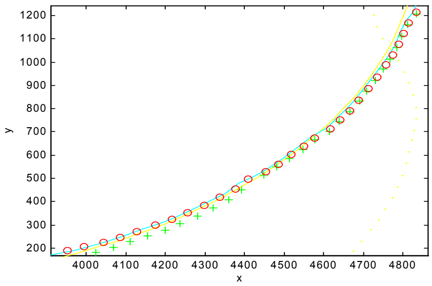


Figure 4: Estimated trajectories (sub trajectory T2).

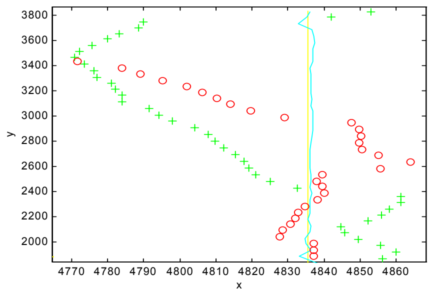


Figure 5: Estimated trajectories (sub trajectory T3).

Table 1: Average estimation errors (Scenario 1).

	T1			T2			T3		
	SM	SMI	AMM	SM	SMI	AMM	SM	SMI	AMM
εx (cm)	6.2 5	3.2 3	2.5	13. 2	10.8	15. 3	31. 9	31. 2	1.2
εy (cm)	13. 6	16. 7	2.3	23. 9	11.9	8.2 5	19. 2	5.7 5	3.2 3
$\varepsilon\theta$ (10^{-3} rad)	81. 1	66. 9	3.8	32. 2	39.9	35. 6	136	125	267 .9

Ndx =99.37%, Ndy = 84.37%, $Nd\theta$ =99.37%,
$Ndxe$ =99.37%, $Ndye$ =97.5%, $Nd\theta e$ =99.37%.

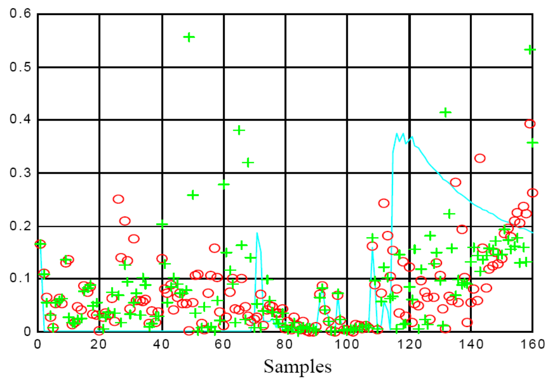


Figure 6: Orientation error.

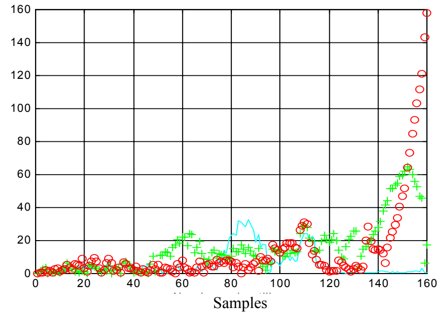


Figure 7: Position error with respect to X axis.

<u>Scenario 2</u>

-Noise-to-signal Ratio of odometric sensors: right encoder: 10%, left encoder: 10%
-Noise-to-signal Ratio of Gyroscope: 3%
-Noise-to-signal Ratio of telemeters: 4% of the odometric elementary step (40% of the state noise)
-"A priori" knowledge on the variance in initial state: Good
-"A priori" knowledge on noise variances (i) telemeters and state: Good; (ii) gyroscope: Bad

The results presented here (Table 2 and Fig. 8-10) show the influence of signal-to-noise ratio and the estimation of noise variances on performances of SM and SMI filters. In this scenario, the initial variance of measurement noise of the gyroscope is incorrectly estimated. Contrary to AMM approach, filters SM and SMI do not carry out any adaptation of this variance, leading to unsatisfactory performance.

Figure 11 illustrates the evolution of state noise variance estimate compared to the average variance. Note that the ratio between variances reaches 1.7 on sub trajectory T1, 3.0 on sub trajectory T2, and 3.3 on sub trajectory T3. It is important to mention that the algorithm proposed for estimation of variances estimates the actual value of state noise variance and not its average value. These results are related to the fact that the signal-to-noise ratio is weak both for the odometer and the telemeters.

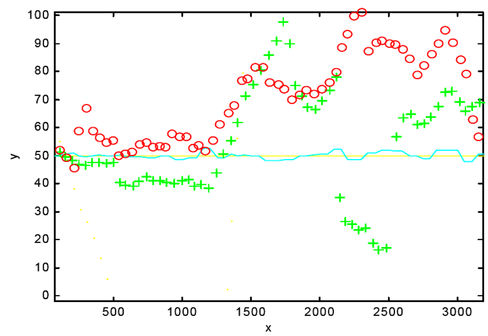


Figure 8: Estimated trajectories (sub trajectory T1).

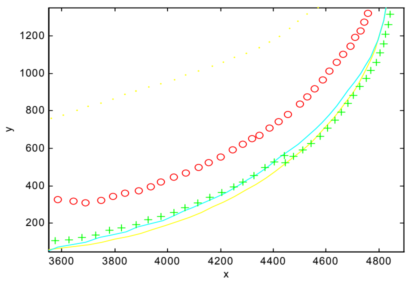


Figure 9: Estimated trajectories (sub trajectory T2).

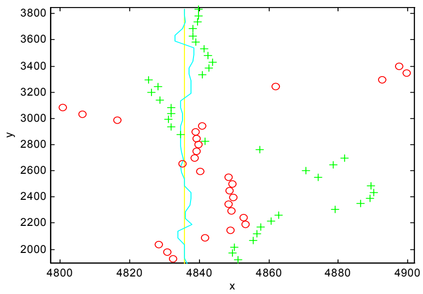


Figure 10: Estimated trajectories (sub trajectory T3).

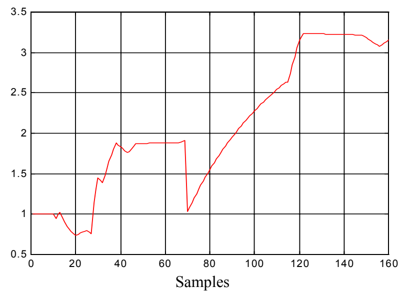


Figure 11: Ratio between the estimate of state noise variance and the average variance.

Table 2: Average estimation errors (Scenario 2).

	T1			T2			T3		
	SM	SMI	AMM	SM	SMI	AMM	SM	SMI	AMM
εx (cm)	11.7	11	1.8	19	75	13.6	17.3	40	1.3
εy (cm)	16.7	21	1	39	179	17.4	15.7	117	1.93
$\varepsilon\theta$ (10^{-3} rad)	99.3	129	1.5	42.9	175	35.4	97.5	167	37.8

Ndx =87.5%, Ndy =66%, $Nd\theta$ =99.37%, $Ndxe$ =87.5%, $Ndye$ =82.5%, $Nd\theta e$ =99.37%.

5 CONCLUSIONS

We presented in this paper a multiple model approach for the robust localization of a mobile robot. In this approach, the localization is considered as a hybrid process, which is decomposed into multiple models. Each model is associated with a mode and an interval of validity corresponding to the observation domain. A Markov chain is employed for the prediction of each model according to the robot mode. To prevent divergence of standard Kalman Filtering, we proposed the application of an adaptive algorithm for the adjustment of the state and measurement noise covariance matrices. For an efficient estimation of noise variances, we used an ad hoc technique consisting of a measure selection for filtering unreliable readings. The simulation results which we obtain in different scenarios show better performances of the proposed approach compared to standard existing filters. These investigations into utilizing multiple model technique for robust localization show promise and demand continuing research.

REFERENCES

J. Borenstein, B. Everett, L. Feng, Navigating Mobile Robots: Systems and Techniques. A.K. Peters, Ltd., Wellesley, MA, 1996.

C. Harris, A. Bailley, T. Dodd, Multi-sensor data fusion in defense and aerospace, *Journal of Royal Aerospace Society*, 162 (1015) (1998) 229-244.

J.B. Gao, C.J. Harris, Some remarks on Kalman filters for the multi-sensor fusion, *Journal of Information Fusion*, 3 (2002) 191-201.

C. Chui, G. Chen, Kalman filtering with real time applications, *Springer Series in Information Sciences, Springer-Verlag*, New-York 17 (1987) 23-24.

K.O. Arras, N. Tomatis, B.T. Jensen, R. Siegwart, Multisensor on-the-fly localization: precision and reliability for applications, *Robotics and Autonomous Systems*, 34 (2001) 131–143.

S. Borthwick, M. Stevens, H. Durrant-Whyte, Position estimation and tracking using optical range data, *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1993, pp. 2172–2177.

P. Jensfelt, H.I. Christensen, Pose tracking using laser scanning and minimalistic environment models, *IEEE Transactions on Robotics and Automation*, 17 (2) (2001) 138–147.

J. Neira, J.D. Tardós, J. Horn, G. Schmidt, Fusing range and intensity images for mobile robot localization, *IEEE Transactions on Robotics and Automation*, 15 (1) (1999) 76–84.

J.A. Pérez, J.A. Castellanos, J.M.M. Montiel, J. Neira, J.D. Tardós, Continuous mobile robot localization: vision vs. laser, *Proceedings of the IEEE International Conference on Robotics and Automation,* 1999, pp. 2917–2923.

G.A. Borges, M.J. Aldon, Robustified estimation algorithms for mobile robot localization based geometrical environment maps, *Robotics and Autonomous Systems,* 45 (2003) 131-159.

L. Kleeman, Optimal estimation of position and heading for mobile robots using ultrasonic beacons and dead-reckoning, *Proceedings of the IEEE International Conference on Robotics and Automation*, 1992, pp. 2582–2587.

L. Jetto, S. Longhi, G. Venturini, Development and experimental validation of an adaptive Kalman filter for the localization of mobile robots, *IEEE Transactions on Robotics and Automation*, 15 (2) (1999) 219–229.

S.J. Julier, J.K. Uhlmann, A non-divergent estimation algorithm in the presence of unknown correlations, *Proceedings of the American Control Conference*, 1997.

X. Xu, S. Negahdaripour, Application of extended covariance intersection principle for mosaic-based optical positioning and navigation of underwater vehicles, *Proceedings of the IEEE International Conference on Robotics and Automation*, 2001, pp. 2759–2766.

Z. Djama, Y. Amirat, Multi-model approach for the localisation of mobile robots by multisensor fusion, *Proceedings of the 32th International Symposium On Automotive Technology and Automation*, 14th-18th june 1999, Vienna, Autriche, pp. 247-260.

Z. Djama, Approche multi modèle à sauts Markoviens et fusion multi capteurs pour la localisation d'un robot mobile. *PhD Thesis, University of Paris XII*, May 2001.

H. A. P. Blom and Y. Bar-Shalom, The interacting multiple model algorithm for systems with Markovian switching coefficients, *IEEE Trans. Automat. Contr.*, vol. 33, pp. 780–783, Aug. 1988.

X. R. Li, Engineer's guide to variable-structure multiple-model estimation for tracking, in Multitarget-Multisensor Tracking: Applications and Advances, *Y. Bar-Shalom and D.W. Blair,* Eds. Boston, MA: Artech House, 2000, vol. III, ch. 10, pp. 499–567.

X. R. Li, Hybrid estimation techniques, Control and Dynamic Systems: *Advances in Theory and Applications, C. T. Leondes*, Ed. New York: Academic, 1996, vol. 76, pp. 213–287.

E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan, Interacting multiple model methods in target tracking: A survey, *IEEE Trans. Aerosp. Electron. Syst.*, vol. 34, no. 1, pp. 103–123, 1996.

F. Dufour, Contribution à l'étude des systèmes linéaire à saut markoviens, *PhD Thesis*, University of Paris Sud University, France, 1994.

M. Oussalah, Suboptimal Multiple Model Filter for Mobile Robot Localization, International Journal of Robotics Research, vol. 20, no. 12, December 2001, pp. 977-989.

COMMUNICATION AT ONTOLOGICAL LEVEL IN COOPERATIVE MOBILE ROBOTS SYSTEM

Lucia Vacariu, Mihai Chintoanu
Department of Computer Science, Technical University of Cluj Napoca, 26 Baritiu str, Cluj Napoca, Romania
Navigon Company Cluj Napoca Subsidiary, Cluj Napoca, Romania
Lucia.Vacariu@cs.utcluj.ro, mchintoanu@rediffmail.com

Gheorghe Lazea, Octavian Cret
Department of Automation, Technical University of Cluj Napoca, Cluj Napoca, Romania
Department of Computer Science, Technical University of Cluj Napoca, Cluj Napoca, Romania
Gheorghe.Lazea@aut.utcluj.ro, Octavian.Cret@cs.utcluj.ro

Keywords: Mobile robots, Multiagents system, Communication, Ontology, Cooperation.

Abstract: Mobile robot applications are faced with the problem of communicating large amounts of information, whose structure and significance changes continuously. A traditional, layered style of communication creates a cooperation problem as fix protocols and message meaning cannot mediate the dynamic, novel types of behaviors mobile robots are acquiring in their environment. We propose by contrast, a non-hierarchical communication control mechanism based on the software paradigm of multiagent systems that have specialized ontologies. It is a communication at ontological level, which allows efficient changes in the content of the messages among robots and a better adaptability and specialization to changes. The intended application is a cooperative mobile robot system for monitoring, manipulating and cleaning in a supermarket. The focus of the paper is to simulate a number of well-defined, controllable and repeatable critical ontological situations encountered in this environment that validate the system cooperation tasks.

1 INTRODUCTION

Research in mobile robotics has achieved a level of maturity that allows applications to move from laboratory into public-level applications. This leads necessarily to a situation where several robots, perhaps with different goals and different internal representations, must cooperate for achieving shared and not reciprocally damaging goals (Arai, Pagello, and Parker, 2002).

A robot control system based on the paradigm of multiagent systems can provide rich, active entities that can adapt, represent and communicate appropriate abstractions (Vacariu, et al, 2004).

Communicating relevant information among heterogeneous units not designed to cooperate in the first place raises several challenges. The communication protocols cannot have a fix hierarchical organization as such solutions have been shown to lack flexibility for expanding and adapting knowledge and behavior for cooperating robots. The precondition is that the communication between robots is enforced by a number of component multiagents that have an efficient exchange of information based on a large spectrum of ontologies.

It has been recognized that an efficient and correct communication will improve the characteristics and the working of whatever multiagent system (Russell and Norvig, 2002).

In the design of multiagent systems, it is possible to introduce an ontological level, where specific concepts for different relevant domains of applications are described (DiLeo, Jacob, and DeLoach, 2002). This level can be an information source for multiagents.

Starting from these premises, we want to demonstrate that the use of communication at the ontological level of information representation is a reliable method to obtain a correct functioning of a system of mobile robots acting in cooperation.

The paper is structured as follows: Section 2 shows briefly the benefits of introducing the ontological level in the multiagent systems design, to achieve the information exchange. Section 3 describes a cooperative mobile robots system for monitoring, manipulating and cleaning in a supermarket and explains the use of ontological level in communication. Section 4 reports the test conditions and results obtained in the simulation. Section 5 presents conclusions and research ideas for future work.

2 ONTOLOGY IN MULTIAGENT SYSTEM DESIGN

Ontology denotes here a common vocabulary with basic concepts and relationships of a domain (Noy and McGuinness, 2001). The domain description requires representations for types and properties of objects, and for relations between them. Software agents use ontologies and knowledge databases on ontologies as information sources.

The ontology classes describe concepts of a domain in a taxonomic hierarchy. The roles describe classes' properties and have different values for different instances. They may be restricted or not to a certain set of values. The knowledge database is created from individual instances of classes, with specific values for properties, and supplemental restrictions.

The aim of ontology integration in a multiagent system design is a description of information using the ontological level of knowledge representation. A common vocabulary of a domain allows definitions of basic concepts and relations between them to be included in the design process of a multiagent system. This creates premises to obtain a new system with new facilities, that is also more robust and adaptable.

We use the framework of MultiAgent System Engineering MASE (DeLoach, Wood, and Sparkman, 2001) that starts from an initial set of purposes and makes the analysis, design and implementation of a functional multiagent system. The ontology can be built during the analysis stage and after that is being used to further create new goals for the system. Purposes often involve parameter transmissions and consequently the ontology classes can be used as parameters.

Objects of the data model are specified as parameters in inter-agent conversations. In role refining and conversation building, that involves meta-message transmissions, that includes the type specification for transmitted parameters. The actions can use information contained in the parameter attributes because the types of parameters and attributes of types are all known. The internal variables representing purposes and conversations can be standardized with respect to the system ontology. The validity of conversations is automatically verified using parameters and variables.

Communication is achieved by inter-agent conversation. A conversation defines the coordination protocol between two agents and uses two communication diagrams between classes: one for the sender and the other one for the receiver. The communication diagram between classes is made of finite state machines, which define the states of conversation between the two participant classes.

The sender always starts a conversation by sending the first message. Any agent who receives a message compares it with all the active conversations from his list. If a match is found, the required transition is made, and the new required activities of the new state are achieved. If there is no match, the agent compares the message with all the other possible messages he might have with the sender agent, and if it finds one that matches, it will start a new conversation.

To be able to participate in conversations, the agents use information from disposition diagrams where the name, address and configurations of agents and stations are saved.

Conversations have no blocking states; all the states have valid transitions from where it is possible to reach the final state.

The transaction syntax uses UML notation:
receive_message(arg1)[condition]/action^transmit_message(arg2).

Robot systems made by autonomous mobile robots execute missions in spatial environments. The environment description using a spatial ontology will highlight entities with respect to space. The spatial ontology defines concepts used to specify space, spatial elements and spatial relations. Therefore, when we create the ontology for the multiagent system used in multirobot system, we insist on physical objects, on concepts that describe the environment using spatial localization of the components. The terms of the concept list are organized in classes and attributes, and an initial model data is elaborated. The necessary concepts of the system are specified for purpose achievement. The communication in multiagent systems uses then the concepts of the developed ontology.

3 COOPERATIVE MOBILE ROBOTS SYSTEM

To verify the effectiveness of using the communication at the ontological level in a mobile multirobot system we designed and implemented a multiagent system. The mobile multirobot system has the purpose to serve, in cooperation, a supermarket. The goals the system must provide are supervising the supermarket, manipulating objects of different types and dimensions, cleaning and giving alarm if necessary. Logging of all events and situations is also necessary.

The multirobot system is cooperative. All the robots have the goal to accomplish activities in collaboration.

3.1 Multiagent System Design

The specification for the cooperative multiagent system gives the following general objectives: to supervise the supermarket; to move in the assigned zone in the market; to identify forms or objects that do not match the knowledge about the supervised environment and notify theirs positions; to manipulate objects (pick up, move and push); to coordinate activities; to exchange information (sending and receiving); to select the appropriate agent with respect to activities (by competencies, positions in space, costs); to clean, and to log events.

The system's objectives are grouped in a hierarchy, based on the importance and connections between them (Figure 1).

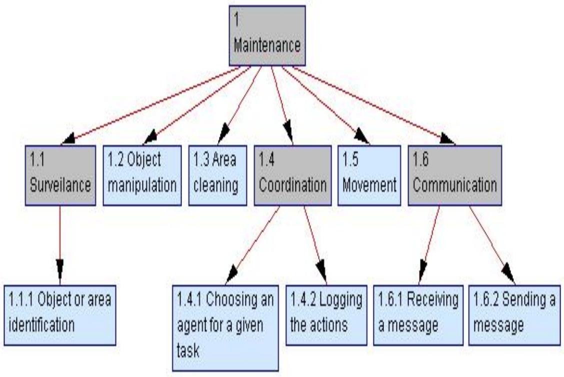


Figure 1: Objectives hierarchy.

From the initial specifications of the system, one defines the base scenarios for identifying the communication ways, like obstacle movement, zone cleaning, and intruder identification.

Using these cases, we build sequence diagrams from where the initial roles are then identified.

In the multiagent system design we highlight the defining elements of the domain described by the ontology and the inter-agent communication methodology based on using ontology concepts.

The taxonomy of the used ontology and part of the concepts are taken from the SUMO (Suggested Upper Merged Ontology) ontology (SUMO, 2006). We insist here on physical entities, with necessary add-ons for multirobot system applications. The ontology has a larger information domain than necessary for this particular application, and allows simple updates. This facility will assure their usage also in futures applications.

The ontology base concept is the *Entity*, the central node of the hierarchy. In the tree we have *PhysicalEntity* and *AbstractEntity*. We consider every concept that may be looked like an entity with spatial and temporal position as a *PhysicalEntity*. In this category, a distinction is made between *Object* and *Process*. The *Object* concept has complete presence in any moment of his life. The ontology tree is developed with concepts down to the multirobot application agent's level. *Object* can be *Group* or *Individual*. We develop the *Individual* concept with *Region*, *Substance* and *ConectedObject*.

The objectives structures and sequence diagrams were converted in roles and goals associated with them. The model of roles includes roles, everyone goals and also information about goals interactions (Figure 2).

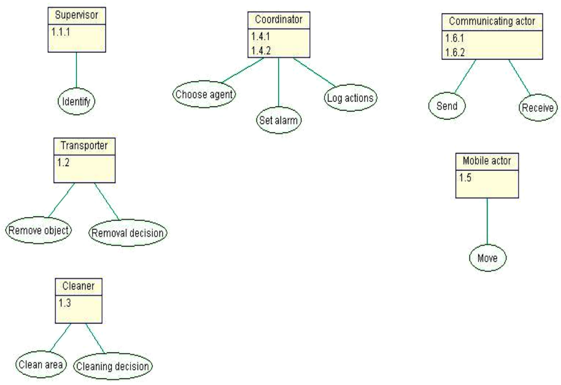


Figure 2: Roles model.

3.2 Multiagent System Conversations

From the role model, the multiagent system implements different agent types: supervisor, coordinator, transporter, cleaner, mobile agent, and communication agent. The agent's class diagram highlights the roles and the conversations between different types of agents. For every identified communication, two conversation diagrams are necessary, each with a sequence of steps. For instance, obstacle movement has conversation diagrams for supervisor, coordinator, and transporter. Every sent message has a name and content. The content can be missing if the message has only control role.

In our approach, the message content is always an ontology object. By this kind of messages we can transmit among agents any concept from the system ontology. Each time when information about a system object is exchanged, an ontology object will be in fact transmitted. The minimum information of content is the type, coordinates and dimensions of an object.

In the multiagent system built to serve a supermarket, the obstacles that must be moved can be different objects from the ontology. For instance, when the coordinator sends a message to the transporter for moving an object, this object can be taken from two conceptual categories, with the same parent *ConnectedObject*. One concept is *Artifact – Product* where the obstacle can be *Desk*, *Table*, *Chair*, *Case*, *Bin*, and the other concept is *OrganicObject*, that can be *Human* or *Plant*. The transporter will send to the supervisor a message with an object that is itself, *TransporterAgent*, and has the coordinates, self-identifier, and other information necessary for mission completion. In the ontology taxonomy, *TransporterAgent* is part of the *MobileAgent* concept, which identifies the mobile robots used in mobile multirobot system, and comes from *Artifact – Device – Robot* concepts.

The agents are instantiated and put into a network diagram. A number, type and location identify them. The built multiagent system is dynamic. Therefore in every moment new agents can be introduced. Agents are placed on the same computer or in remote computers, based on their association with mobile robots. The collective communication is provided by broadcast transmission. Every agent is a separate execution thread and has one's own port for sending and receiving messages.

By using the method of information exchange at the ontological level, it is possible to share knowledge from the entire ontology between any agents from the multiagent system. This improves the capabilities of the system and assures a reliable execution of the missions, a better adaptability and specialization to changes.

4 SIMULATION CONDITIONS AND RESULTS

The multiagent system implementation is made in Java language, with IntelliJ IDEA support (IntelliJ IDEA, 2006). Conversations structure uses agentMom (Message Oriented Middleware), component of agentTool (MASE developer) (agentTool, 2005). The application uses BroadcastServer interface, implemented by every agent or agent component. An agent can share different types of conversations. The agents and conversations are in different threads of execution. The agents run in parallel, independently on each other. All conversations are made at the same time.

The ontology classes are instantiated and transmitted in messages content. The system's agents have different knowledge from the ontology. For instance, the supervisor agent knows all kinds of objects used in conversations because it is the one who identifies the objects and decides what type they have. The coordinator has the same knowledge like the supervisor because it makes the connection with all the agents from the system. The transporter and cleaner agents know only sub-trees of ontology, those in correspondence with their activity areas.

The concepts used in conversations from ontology are *Case* for obstacle, *ConnectedObject* for dirt and *Human* for intruder.

Objects use length, width, and height dimensions. The height is used to compute the volume. The intruders don't use dimensions because only their position is important for the system. The positions of objects are implemented in the *Object* concept.

We implement a WorldInstance class with information about the types and positions of agents and objects, the minimum, maximum, and implicit dimensions, and the work area. A part of this information can be modified.

The multiagent system developed for the mobile multirobot system has been tested in simulation conditions. We built a simulation framework for sets of activities. The agents are associated with robots having different functionalities. Possible problems of synchronization and sharing resources have been solved.

Agents have windows associated with them. These are placed in tabs in main window of the application interface used for messages exchange and actions. These messages show that the ontological information is indeed exchanged. The coordinator has two supplementary windows, to manage the control of supervisors and transporters.

We chose to simulate a number of well-defined, controllable and repeatable critical ontological situations encountered in this environment. The functional situations tested were: detection of obstacles, dirt, and intruders in the supervised area, moving detected obstacle, cleaning detected dirt, moving supervisors in other areas, new transporter or cleaner agent with better capacities insertion, coordination of supervisors, coordination of transporters and cleaners, alarm activation, and actions logging.

In the environment described in Figure 3.a, the system allows detection of two cases, Square and Rectangle, by supervisors S0 and S1. S0 and S1 detect both the Square, but just one transporter is sent to move it. The transporters will move those cases that are the closest from them. The other case and dirt located outside the supervisors areas won't be moved (Figure 3.b). To detect these case and dirt, S1 supervisor is moving in another area (Figure 3.c). After moving the case by T1 and cleaning the dirt by C0, all cases are in the storehouse; transporters are in the waiting area and the cleaner remains where he finished his last action. The environment looks like Figure 3.d.

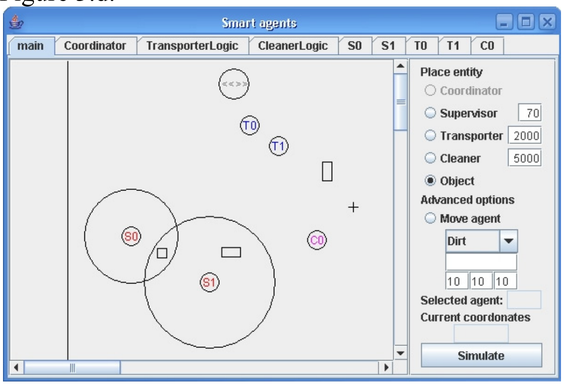


Figure 3.a: Environment.

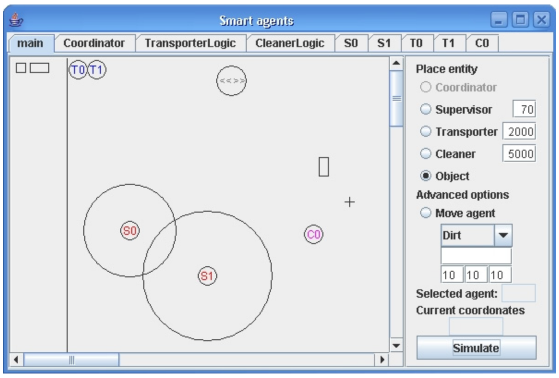


Figure 3.b: Square and Rectangle detection and moving.

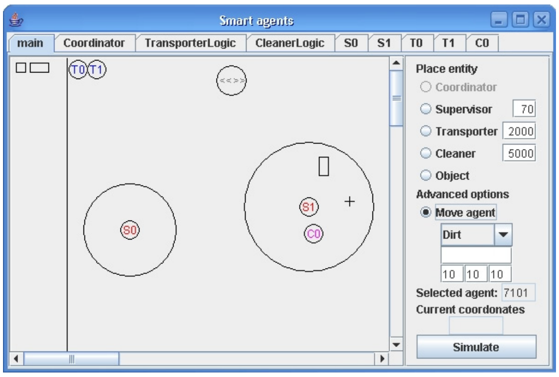


Figure 3.c: Case and dirt detection.

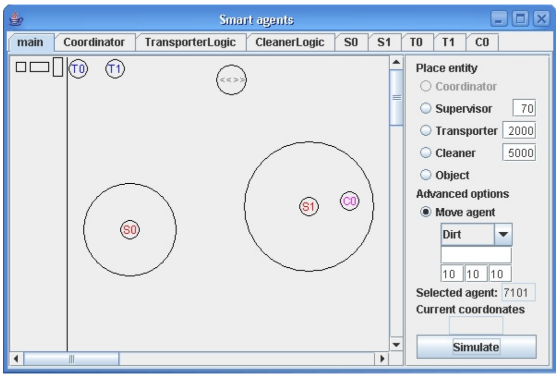


Figure 3.d: Final simulation.

Agents' windows show that the system, as designed, provided all necessary information in communications between coordinator, supervisors, transporters and cleaners. Objects from ontology were successfully exchanged and all agents correctly finished their goals, as seen in the coordinator window (Figure 4).

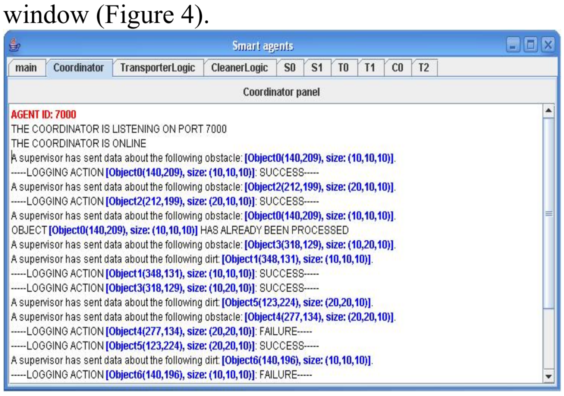


Figure 4: Coordinator window.

The action is not executed and it is logged as failure, if some information differs for some agents (especially the dimensions of objects). This implies that new agents with better capabilities must be introduced to accomplish the goals.

One other environment situation (Figure 5.a.) has the transporter T2 with larger capacity and an intruder that will be detected and after that the coordinator will start the alarm (Figure 5.b).

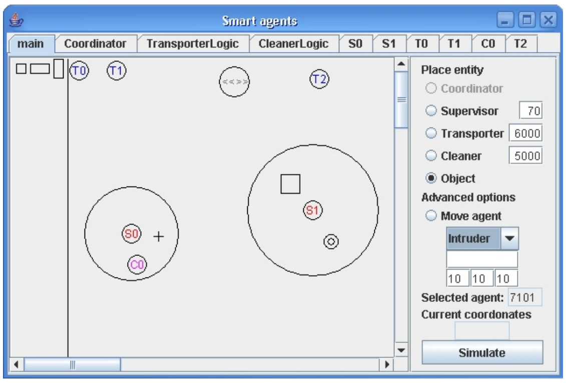


Figure 5.a: New agent and objects.

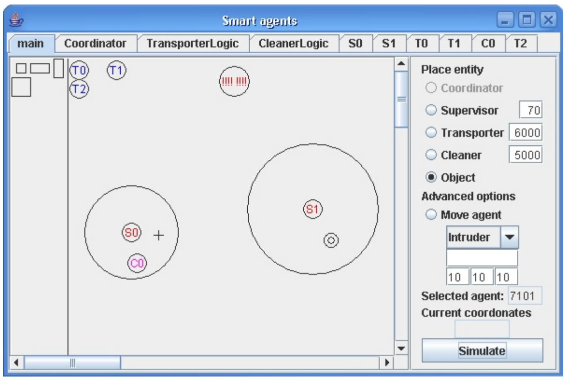


Figure 5.b: Intruder detection and alarm.

The dirt is not removed because the cleaner agent C0 is filled with the previous dirt. The new big case is successfully processed by T2.

The coordinator performs logging of all events and situations arising, for later consultation. For every action we record the time elapsed for its execution, the object that was the goal of the action together with its characteristics, the result of action, and which agent executed the action. When an action is not finished or it was instantaneously accomplished, in the log file will be record only the time of object registration.

5 CONCLUSIONS

The design of multiagent system is valid with respect to the purposes and requests of mobile multirobot system. All agents are able to accomplish successfully the missions and actions assigned to them. The system is robust and flexible.

Using the ontological information in inter-agent communications simplifies the communication process. Conversations made with ontological information are oriented to describe the spatial representation by concepts of the environment, specific to robotic systems.

The system allows efficient changes in the content of the messages among robots and proves that heterogeneous agents, having dissimilar knowledge can, by using ontology information, exchange necessary information to accomplish complex goals.

The system has been tested under simulated conditions. The positive results are leading to our next goal, to make tests under real environments, with mobile robots.

We will also test our approach in other types of missions for mobile multirobot systems.

REFERENCES

AgentTool, 2005. http://macr.cis.ksu.edu/projects/agentTool/

Arai, T., Pagello, E., and Parker, L.E., 2002. Advances in Multi-Robot Systems. In *IEEE Transactions on Robotics and Automation Magazine*. Vol. 18, No. 5, pp. 655-661.

DeLoach, S.A., Wood, M.F., and Sparkman C.H., 2001. Multiagent Systems Engineering. In: *The International Journal of Software Engineering and Knowledge Engineering*. Vol. 11, No. 3, pp. 231-258.

DiLeo, J., Jacob, T., and DeLoach, S., 2002. Integrating Ontologies into Multiagent Systems Engineering. In *AOIS-2002 Proceedings of Fourth International Bi-Conference Workshop an Agent-Oriented Information Systems*. Bologna, Italy.

IntelliJ IDEA, 2006. http://www.jetbrains.com/idea

Noy, N.F., McGuinness, D.L., 2001. Ontology Development 101: A Guide to Creating Your First Ontology. *Stanford Knowledge Systems Laboratory Technical Report KSL-01-05* and *Stanford Medical Informatics Technical Report SMI-2001-0880*.

Russell, S.J., and Norvig, P., 2002. *Artificial Intelligence: A Modern Approach*, Prentice Hall, USA, 2nd edition.

SUMO, 2006. http://protege.stanford.edu/ontologies/sumoOntology/sumo_ontology.html.

Văcariu, L., Blasko, P.C., Leţia, I.A., Fodor, G. and Creţ, O.A., 2004. A Multiagent Cooperative Mobile Robotics Approach for Search and Rescue Missions. In *IAV2004 Proceedings of the 5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles*. Lisbon, Portugal.

INTROSPECTION ON CONTROL-GROUNDED CAPABILITIES
A Task Allocation Study Case in Robot Soccer

Christian G. Quintero M., Salvador Ibarra M., Josep Ll. De la Rosa and Josep Vehí
University of Girona, Campus Montilivi, 17071, Girona, Catalonia, Spain
cgquinte@eia.udg.cat,sibarram@eia.udg.cat, peplluis.delarosa@eia.udg.cat, vehi@eia.udg.cat

Keywords: Machine learning in control applications, Software agents for intelligent control systems, Mobile robots and autonomous systems, Autonomous agents, Reasoning about action for intelligent robots.

Abstract: Our proposal is aimed at achieving reliable task allocation in physical multi-agent systems by means of introspection on their dynamics. This new approach is beneficial as it improves the way agents can coordinate with each other to perform the proposed tasks in a real cooperative environment. Introspection aims at including reliable physical knowledge of the controlled systems in the agents' decision-making. To that end, introspection on control-grounded capabilities, inspired by the agent metaphor, are used in the task utility/cost functions. Such control-grounded capabilities guarantee an appropriate agent-oriented representation of the specifications and other relevant details encapsulated in every automatic controller of the controlled systems. In particular, this proposal is demonstrated in the successful performing of tasks by cooperative mobile robots in a simulated robot soccer environment. Experimental results and conclusions are shown, stressing the relevance of this new approach in the sure and trustworthy attainment of allocated tasks for improving multi-agent performance.

1 INTRODUCTION

In recent years, *A*rtificial *I*ntelligence *(AI)* approaches have been combined with traditional control theory to obtain intelligent systems. In this sense, the advances of the *AI* community, together with the new techniques in the control community, have presented a fresh path for further progress (Halang *et al.*, 2005) (Murray *et al.*, 2003). In particular, complex control systems (Sanz *et al.*, 2003) have been managed using agents. Nowadays, a complex control system is a distributed, asynchronous and networked framework and the whole process must be considered as a multi-agent system that requires coordination and cooperation to perform the proposed tasks (Luck *et al.*, 2005) (Stone and Veloso, 2000). Specifically, agent technology helps to solve task allocation problems in real control scenarios by means of its distributed and cooperative problem-solving paradigm (Jennings and Bussmann, 2003). However, these agents lack appropriate reasoning on knowledge related to the physical features of the controlled system (*physical knowledge mainly related to dynamics*). In addition, such relevant knowledge is not appropriately reflected and communicated by the agents. These lacks do not facilitate more suitable task allocation by the agents in a multi-task scenario. Explicitly, lack of appropriate reasoning on physical knowledge results in a lower number of successful coordinated tasks performed by agents. In fact, this lack is currently a significant impediment to reducing complexity and achieving appropriate levels of control, coordination and autonomy in task allocation problems (Murray *et al.*, 2003). That is to say, agents don't reflect on their control-oriented knowledge and this knowledge is not currently properly taken into account in the utility/costs functions used in the agents' decision-making for allocating tasks. Physical agents are particular examples of controlled systems. In recent years, mobile robots - one typical representation of physical agents - have become progressively more autonomous and cooperative. So we have used mobile robots in this approach without loss of general applicability. Such autonomous cooperating robots must then have reliable self-knowledge if they are to improve the task allocation performance in a multi-robot environment. Specifically, this self-knowledge must be based on an appropriate agent-oriented representation of their automatic controllers in the utility/cost functions used for allocating tasks. With this representation, the physical agents can consider their bodies in a better and more reliable way, whenever it is necessary to allocate and perform tasks in a multi-agent system. In this sense, the paper proposes an introspection-based approach to provide agents a cognitive ability for reasoning on their physical knowledge, aiming at making physically feasible task allocation to improve the cooperative multi-agent performance.

2 RELATED WORK

Several authors (Goldberg and Matarić, 2000) (Gerkey and Matarić, 2002) (Scheutz, 2002) (Balakirsky and Lacaze, 2002) (Goldberg *et al.*, 2003) (Gerkey and Matarić, 2004) (Dahl *et al.*, 2004) (Lagoudakis *et al.*, 2005) (Koenig *et al.*, 2006) (Sariel and Balch, 2006) (Ramos *et al.,* 2006) have studied the problems related to task allocation in multi-robot environments based on utility/cost functions. These approaches present suitable approaches to task/action selection mainly take into account domain knowledge in the agents' decision-making. However, an approach based on control-oriented knowledge has not been completely carried out. In this sense, we want to show how introspection on the physical agents' dynamics contributes to a more suitable task allocation by considering the physical agents' bodies in a better and more reliable way. Such consideration is directly related to the automatic controllers of the physical agents. Thus, reliable information is extracted from the controllers to obtain appropriate control-oriented knowledge of the physical agent's body. In this sense, such knowledge is represented by means of specific attributes (*control-grounded capabilities*) focused mainly on the physical agents' dynamics.

3 OUR APPROACH

Before an agent starts a task, it should make a plan for how to reach a given goal. This planning requires the agent to have knowledge about the environment, knowledge that can be represented in the agent's knowledge base. It is the agent's ability to model its own environment that makes it able to reason about this environment, to plan its actions and to predict the consequences of performing these actions. However, much intelligent behavior seems to involve an ability to model not only the agent's external environment but also itself and the agent's own reasoning. Such ability is called introspection (Bolander, 2003).

Introspection is yet another aspect of human reasoning in artificial intelligence systems (Wilson *et al.,* 2001). To have introspection in an artificial intelligence system means that the system is able to reflect on its own knowledge, reasoning, tasks and planning (Bolander, 2003). For instance, before an agent commits in the execution of a task, the agent should register the fact of knowing if it can or cannot perform the task, this needs introspection, due to the agent has to look introspectively into its own knowledge base and from it to arrive at a suitable decision. In addition, in order to decide how well the agent is doing or will do the proposed task, an agent will also need this self-examination capability (*introspection*) (McCarthy, 1999).

The *agent is non-introspective when no information in the knowledge base expresses facts concerning the agent itself. Any non-introspective agent only models its external environment*. Otherwise, *introspective agents differ from non-introspective ones by modelling not only their external environment but also themselves. It is by also have models of themselves they are given the ability to introspect* (Bolander, 2003).

In particular, introspection on the physical agents' dynamics is a previously unexplored research area. So we have focused our work just on this topic for examining its impact in the performance of cooperative multi-agent systems.

3.1 Introspection on the Physical Agents' Dynamics

Physical agents require a sense of themselves as distinct and autonomous individuals able to interact with others, i.e. they require an identity (Duffy, 2004). A complete concept of identity therefore constitutes the set of internal and external attributes associated with any given physical agent based on introspection of its physical and "mental" states and capabilities. In this work, the notion of internal and external relates to the attributes of a physical agent analogous to Shoham's notion of capabilities in multi-agent systems (Shoham, 1993). Thus, there are two distinct attributes that are local and particular to each physical agent within a cooperative system:

- Internal Attributes: beliefs, desires, intentions, the physical agent's knowledge of self, experiences, a priori and learned knowledge.

• External Attributes: the physical presence of the agent in an environment; its actuator and preceptor capabilities (e.g., automatic controllers) and their physical features.

In particular, a subset of internal attributes (*control-grounded capabilities*) is used to describe the physical agents' dynamics.

Introspection on physical agents' dynamics refers then to the self-examination by a physical agent of the above capabilities to perform tasks. This self-examination mainly considers the agent body's dynamics.

In this sense, an agent's knowledge of its attributes therefore allows a sufficient degree of introspection to facilitate and maintain the development of cooperative work between groups of agent entities (Duffy, 2004). When an agent is "aware" of itself, it can explicitly communicate knowledge of self to others in a cooperative environment to reach a goal. This makes introspection particularly important in connection with multi-agent systems.

In this context, physical agents must reach an agreement in cooperative groups to obtain sure and trustworthy task allocation. Sure and trustworthy task allocation refers to an allocation accepted by the agents only when they have a high certainty about correctly performing the related tasks.

To achieve sure and trustworthy task allocations, each physical agent must introspect, consider and communicate their physical limitations before performing the tasks. Without introspection, physical agents would try to perform actions with no sense, decreasing the number of successful tasks performed by them.

3.2 Formalization Aspects

Let us to suppose that a physical agent A_α is a part of a cooperative group G. A cooperative group must involve more than one physical agent. That is,

$$\exists A_i, A_j \in G \mid A_i \neq A_j \wedge G \subseteq AA$$

Where AA is the set of all possible physical agents in the environment. Let us to define the set of control-grounded capabilities CC to represent the physical agent's dynamics as a subset of the internal attributes IA of a physical agent A_α such that:

$$CC(A_\alpha) \subseteq IA(A_\alpha)$$

A capability is part of the internal state of a physical agent that must be useful to represent the physical agent's dynamics, must allow computational treatment to be accessible and understandable by agents and must be comparable and combined with other capabilities to be exploited as a decision tool by agents.

Let us to define the set of automatic controllers C, whose actions provoke the physical agent's dynamics, as a subset of the external attributes EA of a physical agent A_α such that:

$$C(A_\alpha) \subseteq EA(A_\alpha)$$

The controllers allow and limit the tasks' executions. So they are key at the moment physical agents *introspect* on their control-grounded capabilities to perform tasks.

Let us to consider the domain knowledge DK for a physical agent A_α made up of a set of environmental conditions EC (e.g., agents' locations, targets' locations), a set of available tasks to perform T, and a set of tasks requirements TR (e.g., achieve the target, avoid obstacles, time constraints, spatial constraints,) as is described by (1).

$$DK(A_\alpha) = EC(A_\alpha) \cup T(A_\alpha) \cup TR(A_\alpha) \quad (1)$$
$$DK(A_\alpha) = \{ec_1, \ldots, ec_o, task_1, \ldots task_p, tr_1, \ldots, tr_q\}$$

Each physical agent has associated a subset of controllers for the execution of tasks of the same kind such that:

$$\forall task_k \in T(A_\alpha),\ \exists C_{task_k}(A_\alpha) \subseteq C(A_\alpha)$$

All controllers involve in the same task has associated the same kind on capabilities such that:

$$\forall c_i \in C_{task_k}(A_\alpha),\ \exists CC_{c_i, task_k}(A_\alpha) \subseteq CC(A_\alpha)$$

The capabilities $CC_{c_i, task_k}$ for a controller i in the execution of a particular task k, are obtained, as in (2), taking into account the agent's domain knowledge DK_{task_k} related to the proposed task such that:

$$CC_{c_i, task_k}(A_\alpha) \subseteq CC(A_\alpha) \subseteq IA(A_\alpha)$$
$$DK_{task_k}(A_\alpha) \subseteq DK(A_\alpha) \mid$$
$$CC_{c_i, task_k}(A_\alpha) = \Psi_{c_i, task_k}(DK_{task_k}(A_\alpha)) \quad (2)$$

$\Psi_{c_i,task_k}$ is a self-inspection function that allows physical agents *introspect* on their capabilities using the controller *i* for the task *k*.

A self-evaluation function $\Phi_{c_i,task_k}$ uses the capabilities in an appropriate way to allow physical agents *know* a certainty index $ci_{c_i,task_k}$ related to the correct execution of the proposed task *k* using the controller *i* as is described in (3).

$$ci_{c_i,task_k}(A_\alpha) = \Phi_{c_i,task_k}(CC_{c_i,task_k}(A_\alpha)) \quad (3)$$

The set of all certainty indexes for a specific task *k* is constituted by all $ci_{c_i,task_k}$ of the controllers in this task:

$$\forall c_i \in C_{task_k}(A_\alpha),\ \exists ci_{c_i,task_k}(A_\alpha) \subseteq CI_{task_k}(A_\alpha)$$
$$\text{Where } CI_{task_k}(A_\alpha) \subseteq CI(A_\alpha)$$

CI constitutes the set of all certainty indexes related to the available tasks *T* for the agent A_α. A certainty index provides physical agent a degree of conviction concerning the truth of its knowledge and ability to perform a particular task.

In summary, the functions (Ψ,Φ) provide physical agents powerful tools for introspection-level reasoning and suitable model of themselves.

Currently, there are several alternatives to implement independently or jointly the above functions. Thus, soft-computing techniques (e.g., neural networks, case-based reasoning and fuzzy logic) and control techniques (e.g., model-predictive control, multiple model adaptive controllers and switching control systems) are commonly used.

3.3 Task Allocation Algorithm

There are several coordination parameters to take into account in the utility/cost functions for allocating tasks. Our approach considers jointly with the introspection, the proximity and the trust.

The introspection parameter $I_{task_k}(A_\alpha)$ refers to a comparative analysis between all possible certainty indexes CI_{task_k} of the controllers in a specific task that allows physical agent, if it is possible, to select a controller for the execution of this task as is described in (4).

$$I_{task_k}(A_\alpha) = \max(CI_{task_k}(A_\alpha)) \quad (4)$$
$$I_{task_k}(A_\alpha) \in [0,1]$$

A high $I_{task_k}(A_\alpha)$ value represents that the agent A_α can perform the task *k* correctly. A low $I_{task_k}(A_\alpha)$ value indicates that the agent cannot perform the task.

Proximity represents the physical situation of each agent in the environment. The proximity parameter $P_{task_k}(A_\alpha)$ is related to the distance between the current location of the agent A_α and the location of the target as is described in (5).

$$P_{task_k}(A_\alpha) = (1 - d_{task_k}(A_\alpha)/d\max) \quad (5)$$
$$P_{task_k}(A_\alpha) \in [0,1]$$

Where $d_{task_k}(A_\alpha)$ is the distance between the physical agent A_α and the target $task_k$ and *dmax* establishes a fixed maximal radius limit according to the target's location.

Trust represents the social relationship among physical agents that rule the interaction and behavior of them. The trust parameter $T_{task_k}(A_\alpha)$ takes into account the result of the past interactions of a physical agent with others. The performance of the proposed task is then evaluated based on $T_{task_k}(A_\alpha)$. Equation (6) shows the reinforcement calculus if goals are correctly reached by the agent. Otherwise, the agent is penalized if goals are not reached using (7).

$$T_{task_k}(A_\alpha) = T_{task_k}(A_\alpha) + \Delta A_{task_k}(A_\alpha) \quad (6)$$
$$T_{task_k}(A_\alpha) = T_{task_k}(A_\alpha) - \Delta P_{task_k}(A_\alpha) \quad (7)$$
$$T_{task_k}(A_\alpha) \in [0,1]$$

Where $\Delta A_{task_k}(A_\alpha)$ and $\Delta P_{task_k}(A_\alpha)$ are the awards and punishments given to A_α in the task *k* respectively. A high $T_{task_k}(A_\alpha)$ value represents a more trusted physical agent in the task.

The utility/cost function $U_{task_k}(A_\alpha)$ is therefore constituted as a proper average of the element-by-element multiplication of the tuples as in (8).

$$U_{task_k}(A_\alpha) = \frac{\sum\left(Th_{task_k} \cdot Ok_{task_k} \cdot Pa_{task_k}(A_\alpha)\right)}{\sum Ok_{task_k}} \quad (8)$$

Where $Pa_{task_k}(A_\alpha)$ is a tuple formed by the coordination parameters such that:

$$Pa_{task_k} = [I_{task_k}(A_\alpha)\ P_{task_k}(A_\alpha)\ T_{task_k}(A_\alpha)]$$

Th_{task_k} is a set of flags (1 or 0) that establishes if the above coordination parameters fulfill their respective decision thresholds such that:

$$Th_{task_k} = [I\_th_{task_k}\ P\_th_{task_k}\ T\_th_{task_k}]$$

We have selected appropriate decision thresholds to set or not the above flags.

Ok_{task_k} is a set of flags (1 or 0) that establishes if the above coordination parameters are currently taking into account in the utility/cost function such that:

$$Ok_{task_k} = [I\_ok_{task_k}\ P\_ok_{task_k}\ T\_ok_{task_k}]$$

Thus, the tasks are allocated to physical agents according to the value of their utility/cost functions (see Equation 8).

4 CASE STUDY

We have used a simulated robot soccer environment to evaluate our approach. Here, task allocation is related to achieve targets with different time and spatial constraints.

In particular, the environment is divided is several *scenes* S = {scene$_1$, scene$_2$, scene$_3$,…, scene$_N$}, each one with a specific set of tasks to allocate $T = \{task_1, task_2, task_3, \ldots, task_{M(scene_j)}\}$. Here, scenes refer to the spatial regions where agents must meet and work jointly to perform the proposed tasks. Physical agents are represented by nonholonomic mobile robots. The robots have just one controller to control its movements in the environment. These physical agents must therefore coordinate their moves to increase the system performance by means of a suitable task allocation in each scene. At the beginning of each simulation, the physical agents are not moving. In addition, the agents' locations are set randomly in each simulation.

5 IMPLEMENTATION

In our implementation, we have designed a heterogeneous team such that G = {A$_1$, A$_2$, A$_3$, A$_4$, A$_5$} where each agent has a different movement controller such that: C(A$_1$) = {c$_1$}, C(A$_2$) = {c$_2$}, C(A$_3$) = {c$_3$}, C(A$_4$) = {c$_4$} and C(A$_5$) = {c$_5$}. There are three scenes S = {attack, midfield, defense} in the environment as is shown in Figure 1. The current scene is established taking into account the current ball's location. For the sake of simplicity, the main task to allocate is to kick the ball in each scene toward the opposite goal. In this sense, for each physical agent is calculated its utility/cost function $U_{task_k}(A_\alpha)$ in the current scene. Such function allows selecting the most suitable agent for that task while the other remaining agents follow a fixed strategy. Specifically, the introspection approach was implemented by using feed-forward back-propagation neural networks. Similarly, the awards and punishments of the trust parameter are different in each scene.

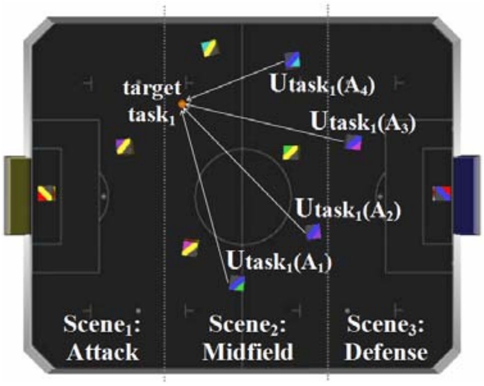


Figure 1: Robot soccer simulator environment.

6 EXPERIMENTAL RESULTS

We established empirical experiments featuring simulated robot soccer tournaments to test system performance when introspection on physical agents' dynamics is used. Our tests used models of the *MiroSot* robots simulator. The selected simulation experiments consist of a predefined number of championships (10), each one with a predefined number of games (10). The performance is measured as a ratio between the total points (*won game: 3 points, tied game: 1 point*) achieved by our team in each championship and the all possible points (30) in this championship where our team play versus a blind opponent robotic team. The initial state of each physical agent was randomly set at every game.

We have compared the system performance highlighting the influence of the introspection in the decisions of our team. In particular, we compared the following teams R vs. I, P vs. P + I, T vs. T + I and P + T vs. P + T + I by modifying the set of flags $\mathrm{Ok}_{\mathrm{task}_k}$ such that, e.g., $\mathrm{Ok}_{\mathrm{task}_k} = [0\ 0\ 0]$ for R and $\mathrm{Ok}_{\mathrm{task}_k} = [1\ 0\ 0]$ for I, where R: random, I: introspection, P: proximity and T: trust.

Figure 2 illustrates how the best system performance is achieved by using introspection in all cases. Here follows a preliminary conclusion: the composition of any parameters with introspection increase the performance as the result of most suitable task allocation in the system. The system performance always improves when the physical agents take into account their physical capabilities based on introspection. The figure also confirms that successful decisions related to task allocation increase when agents use introspection: agents can make better decisions and can consequently make more sure and trustworthy task allocations. In addition, it should be noted that the improvement rate of the introspection approach over the other approaches is a result of the possibility of including the misses in the agents' decisions. In fact, this is an advantage of introspection. Agents can discriminate between the trials in which they have a chance of successfully performing the tasks and those in which they have no chance. In summary, all the above results show that a good decision tool based on introspective reasoning can increase the autonomy and self-control of agents and obtain reliable utility/costs functions in task allocation problems. Introspection and decisions based on capabilities give a trustworthy indication of the real reliability with which each agent performs tasks in cooperative systems.

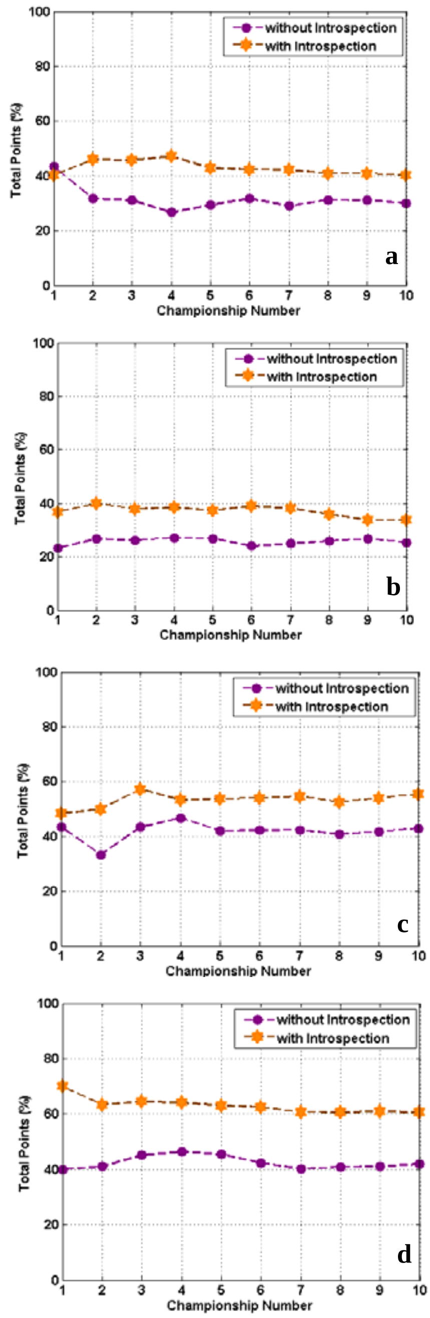


Figure 2: Performance comparison a) R vs. I, b). P vs. P + I, c). T vs, T + I, d). P + T vs. P + T + I.

7 CONCLUSIONS

We argue the need for introspective skills in relation to control-oriented knowledge in physically grounded agents to improve the physical agents' decision-making performance in task allocation problems. Introspection allows physical agents to achieve sure and trustworthy task allocations in cooperative systems, thereby improving the performance of agents in a multi-task environment.

We considered a representation based on capabilities related to the agent body's dynamics. These capabilities were managed in an introspective manner when agents were required to select the most suitable task to perform. Nevertheless, it is still difficult to choose the necessary information to include in the capabilities to represent control-oriented knowledge. In spite of this, our experimental results show that introspection on control-oriented features helps physical agents to make a reliable task allocation and to form sure, achievable and physically grounded commitments for these tasks. Here, introspection on control-oriented features is closely related to the automatic controllers of physical agents. From the controllers, suitable information was extracted to obtain reliable control-oriented knowledge of the physical agent's body. There is still much to explore about how to take advantage of this approach. In the future, we want to extend the contribution to other controlled systems with a larger number of tasks, physical agents, controllers and capabilities, as well as to include introspection-based approaches in auction-based methods for coordination. Furthermore, selection of a paradigm for the implementation of these concepts is not at all trivial, and its development is still an open question.

ACKNOWLEDGEMENTS

This work was supported in part by the Grant TIN2006-15111/ "Estudio sobre Arquitecturas de Cooperación" from the Spanish government.

REFERENCES

Halang W., Sanz R., Babuska R. and Roth H., 2005. "Information and Communication Technology Embraces Control," *Status Report by the IFAC Coordinating Committee on Computers, Cognition and Communication*.

Murray R., Astrom K.J., Boyd S., Brockett R.W. and Stein G., 2003. "Future Directions in Control in an Information-Rich World," *IEEE Control Systems Magazine*, vol. 23, no. 2, pp. 20-33.

Sanz R., Yela A. and Chinchilla R., 2003. "A Pattern Schema for Complex Controllers". *IEEE Emergent Technologies for Factory Automation (ETFA)*, pp. 101-105.

Luck M., McBurne P., Shehory O. and Willmott S., 2005. "Agent Technology: Computing as Interaction," *A Roadmap for Agent Based Computing*, pp. 11-12.

Stone P. and Veloso M., 2000. "Multiagent Systems: A Survey from a Machine Learning Perspective," *Autonomous Robots*, vol. 8, no. 3, pp. 345–383.

Jennings N.R. and Bussmann S., 2003. "Agent-Based Control Systems. Why Are They Suited to Engineering Complex Systems?" *IEEE Control Systems Magazine*, vol. 23, no. 3, pp. 61-73.

Goldberg, D. and Matarić, M. J., 2000. "Robust Behavior-Based Control for Distributed Multi-Robot Collection Tasks". Technical Report IRIS-00-387, USC Institute for Robotics and Intelligent Systems.

Gerkey, B. P. and Matarić, M. J., 2002. "Sold!: Auction Methods for Multi-robot Coordination". *IEEE Transactions on Robotics and Automation*, 18(5):758-768.

Scheutz, M., 2002. "Affective Action Selection and Behavior Arbitration for Autonomous Robots". In *Proc. of IC-AI 02.*

Balakirsky, S. and Lacaze, A., 2002. "Value-Driven Behavior Generation for an Autonomous Mobile Ground Robot". In *Proc. of the SPIE 16th Annual International Symposium on Aerospace/Defense Sensing, Simulation and Controls*.

Goldberg, D., Cicirello, V., Dias, M. B., Simmons, R., Smith, S., and Stentz, A. T., 2003. "Market-based Multi-robot Planning in a Distributed Layered Architecture". *Proceedings from the 2003 International Workshop on Multi-Robot Systems*, vol. 2, pp. 27-38.

Gerkey, B. P. and Matarić, M. J., 2004. "A Formal Analysis and Taxonomy of Task Allocation in Multi-robot Systems. *Int. J. of Robotics Research*, 23(9):939-954.

Dahl, T. S., Matarić, M. J., and Sukhatme, G. S., 2004. "Emergent Robot Differentiation for Distributed Multi-Robot Task Allocation". In *Proc. of the 7th International Symposium on Distributed Autonomous Robotic Systems (DARS)*, pages 191-200.

Lagoudakis, M. G., Markakis, E., Kempe, D., Keskinocak, P., Kleywegt, A., Koenig, S., Tovey, C., Meyerson, A., and Jain, S., 2005. "Auction-based Multi-robot Routing". In *Proc. of Robotics: Science and Systems*, Cambridge, USA.

Koenig, S., Tovey, C., Lagoudakis, M., Markakis, V., Kempe, D., Keskinocak, P., Kleywegt, A., Meyerson, A., and Jain, S., 2006. "The Power of Sequential Single-Item Auctions for Agent Coordination". In *Proc. of the Twenty-First National Conference on Artificial Intelligence.*

Sariel, S. and Balch, T, 2006. "Efficient Bids on Task Allocation for Multi Robot Exploration". In *Proc. of the 19th International FLAIRS Conference*.

Ramos, N., Lima, P., and Sousa, J. M., 2006. "Robot Behavior Coordination Based on Fuzzy Decision-Making". In *Actas do Encontro Cientifico do Robotica*.

Bolander, T., 2003. "Logical Theories for Agent Introspection". Ph.D. thesis, Informatics and Mathematical Modelling (IMM), Technical University of Denmark.

Wilson R. A. and Keil F. C. (Eds.), 2001. "The MIT Encyclopedia of the Cognitive Sciences". Cambridge, MA: The MIT Press, ISBN 0-262- 73144-4.

McCarthy, J., 1999. "Making Robots Conscious of Their Mental States". *Machine Intelligence 15, Intelligent Agents*, St. Catherine's College, Oxford, pp.3-17.

Duffy, B. , 2004. "Robots Social Embodiment in Autonomous Mobile Robotics", *Int.. Journal of Advanced Robotic Systems*, vol. 1, no. 3, pp. 155 – 170, ISSN 1729-8806.

Shoham, Y., 1993. "Agent-oriented programming". *Artificial Intelligence*, vol. 60, no. 1, pp. 51-92.

FURTHER STUDIES ON VISUAL PERCEPTION FOR PERCEPTUAL ROBOTICS

Ozer Ciftcioglu, Michael S. Bittermann
Delft University of Technology
o.ciftcioglu@tudelft.nl, m.s.bittermann@tudelft.nl

I. Sevil Sariyildiz
Delft University of Technology
i.s.sariyildiz@tudelft.nl

Keywords: Vision, visual attention, visual perception, perception measurement.

Abstract: Further studies on computer-based perception by vision modelling are described. The visual perception is mathematically modelled, where the model receives and interprets visual data from the environment. The perception is defined in probabilistic terms so that it is in the same way quantified. At the same time, the measurement of visual perception is made possible in real-time. Quantifying visual perception is essential for information gain calculation. Providing virtual environment with appropriate perception distribution is important for enhanced distance estimation in virtual reality. Computer experiments are carried out by means of a virtual agent in a virtual environment demonstrating the verification of the theoretical considerations being presented, and the far reaching implications of the studies are pointed out.

1 INTRODUCTION

Visual perception, although commonly articulated in various contexts, it is generally used to convey a cognition related idea or message in a quite fuzzy form and this may be satisfactory in many instances. Such usage of perception is common in daily life. However, in professional areas, like computer vision, robotics, or design, its demystification or precise description is necessary for proficient executions. Since the perception concept is soft and thereby elusive, there are certain difficulties to deal with it. For instance, how to quantify it or what are the parameters, which play role in visual perception. Visual perception is one of the important information sources playing role on human's behavior. Due to the diversity of existing approaches related to perception, which emerged in different scientific domains, we provide a comprehensive introduction to be explicit as to both, the objectives, and the contribution of the present research.

Perception has been considered to be the reconstruction a 3-dimensional scene from 2-dimensional image information (Marr, 1982; Poggio, Torre et al., 1985; Bigun, 2006). This *image processing* approach attempts to mimic the neurological processes involved in vision, with the retinal image acquisition as starting event. However, modeling the sequence of brain processes is a formidable endeavor. This holds true even when advanced computational methods are applied for modeling of the individual brain-components' behavior (Arbib, 2003). The reason is the brain processes are complex.

Brain researchers trace visual signals as they are processed in the brain. A number of achievements are reported in the literature (Wiesel, 1982; Hubel, 1988; Hecht-Nielsen, 2006). However, due to complexity there is no consensus about the exact role of brain regions, sub-regions and individual nerve-cells in vision, and how they should be modeled (Hecht-Nielsen, 2006; Taylor, 2006). The brain models are all different due to the different focus of attention that refers to uncountable number of modalities in the brain. Therefore they are inconclusive as to understanding of a particular brain process like perception on a common ground. As a state of the art, they try to form a firm clue for perception and attention beyond their verbal accounts. In modeling the human vision the involved brain process components as well as their interactions should be known with certainty if a deterministic approach, like the image processing approach, is to be successful. This is currently not the case. Well-known observations of visual effects, such as *depth from stereo disparity* (Prince, Pointon et al., 2002)*, Gelb effect* (Cataliotti and Gilchrist, 1995)*, Mach bands* (Ghosh, Sarkar et al., 2006), *gestalt principles* (Desolneux, Moisan et al., 2003), *depth from defocus* (Pentland, 1987) etc. reveal components of the vision process, that may be algorithmically mimicked. However, it is unclear how they interact in human vision to yield the mental act of perception. When we say that we perceived something, the meaning is that we can recall relevant properties of it. What we cannot remember, we cannot claim we perceived, although we may suspect that corresponding image information was on our retina. With this basic understanding it is important to note that the act of perceiving has a characteristic that is *uncertainty*: it is a common phenomenon that we overlook items in our environment, although they are visible to us, i.e., they are within our visual scope, and there is a possibility for their perception. This everyday experience has never been exactly explained. It is not obvious how some of the retinal image data does not yield the perception of the corresponding objects in our environment. Deterministic approaches do not explain this common phenomenon.

The psychology community established the probable "overlooking" of visible information experimentally (Rensink, O'Regan et al., 1997; O'Regan, Deubel et al., 2000), where it has been shown that people regularly miss information present in images. For the explanation of the phenomenon the concept of *visual attention* is used, which is a well-known concept in cognitive sciences (Treisman and Gelade, 1980; Posner and Petersen, 1990; Itti, Koch et al., 1998; Treisman, 2006). However, it remains unclear what attention exactly is, and how it can be modeled quantitatively. The works on attention mentioned above start their investigation at a level, where basic visual comprehension of a scene must have already occurred. An observer can exercise his/her bias or preference for certain information within the visual scope only when he/she has already a perception about the scene, as to where potentially relevant items exist in the visible environment. This early phase, where we build an overview/initial comprehension of the environment is referred to as *early vision* in the literature, which is omitted in the works on attention mentioned above. While the early perception process is unknown, identification of attention in perception, that is due to a task specific bias, is limited. This means, without knowledge of the initial stage of perception its influence on later stages is uncertain, so that the later stages are not uniquely or precisely modeled and the attention concept is ill-defined. Since attention is ill-defined, ensuing perception is also merely ill-defined. Some examples of definitions on perception are *"Perception refers to the way in which we interpret the information gathered and processed by the senses,"* (Levine and Sheffner, 1981) and "*Visual perception is the process of acquiring knowledge about environmental objects and events by extracting information from the light they emit or reflect,*" (Palmer, 1999). Such verbal definitions are helpful to understand what perception is about; however they do not hint how to tackle the perception beyond qualitative inspirations. Although we all know what perception is apparently, there is no unified, commonly accepted definition of it.

As a summary of the previous part we note that visual perception and related concepts have not been exactly defined until now. Therefore, the perception phenomenon is not explained in detail and the perception has never been quantified, so that the introduction of human-like visual perception to machine-based system remains as a soft issue.

In the present paper a newly developed theory of perception is introduced. In this theory visual perception is put on a firm mathematical foundation. This is accomplished by means of the well-established probability theory. The work concentrates on the early stage of the human vision process, where an observer builds up an unbiased understanding of the environment, without involvement of task-specific bias. In this sense it is an underlying fundamental work, which may serve as basis for modeling later stages of perception, which may involve task specific bias. The probabilistic theory can be seen as a unifying theory as it unifies synergistic visual processes of human, including physiological and neurological ones. Interestingly this is achieved without recourse to neuroscience and biology. It thereby bridges from the environmental stimulus to its mental realization.

Through the novel theory twofold gain is obtained. Firstly, the perception and related phenomena are understood in greater detail, and reflections about them are substantiated. Secondly, the theory can be effectively introduced into advanced implementations since perception can be quantified. It is foreseen that modeling human visual perception can be a significant step as the topic of perception is a place of common interest that is shared among a number of research domains, including cybernetics, brain research, virtual reality computer graphics, design and robotics (Ciftcioglu, Bittermann et al., 2006). Robot navigation is one of the major fields of study in autonomous robotics (Oriolio, Ulivi et al., 1998; Beetz, Arbuckle et al., 2001; Wang and Liu, 2004). In the present work, the human-like vision process is considered. This is a new approach in this domain, since the result is an autonomously moving robot with human-like navigation to some extent. Next to autonomous robotics, this belongs to an emerging robotics technology, which is known as perceptual robotics (Garcia-Martinez and Borrajo, 2000; Söffker, 2001; Burghart, Mikut et al., 2005; Ahle and Söffker, 2006; Ahle and Söffker, 2006). From the human-like behaviour viewpoint, perceptual robotics is fellow counterpart of emotional robotics, which is found in a number of applications in practice (Adams, Breazeal et al., 2000). Due to its merits, the perceptual robotics can also have various applications in practice.

From the introduction above, it should be emphasized that, the research presented here is about to demystify the concepts of perception and attention as to vision from their verbal description to a scientific formulation. Due to the complexity of the issue, so far such formulation is never achieved. This is accomplished by not dealing explicitly with the complexities of brain processes or neuroscience theories, about which more is unknown than known, but incorporating them into perception via probability. We derive a vision model, which is based on common human vision experience explaining the causal relationship between vision and perception at the very beginning of our vision process. Due to this very reason, the presented vision model precedes all above referenced works in the sense that, they can eventually be coupled to the output of the present model.

Probability theoretic perception model having been established, the perception outcome from the model is implemented in an avatar-robot in virtual reality. The perceptual approach for autonomous movement in robotics is important in several respects. On one hand, perception is very appropriate in a dynamic environment, where predefined trajectory or trajectory conditions like occasional obstacles or hindrances are duly taken care of. On the other hand, the approach can better deal with the complexity of environments by processing environmental information selectively.

The organization of the paper is as follows. Section two gives the description of the perception model developed in the framework of ongoing perceptual robotics research. Section three describes a robotics application. This is followed by discussion and conclusions.

2 A PROBABILISTIC THEORY OF VISUAL PERCEPTION

2.1 Perception Process

We start with the basics of the perception process with a simple and special, yet fundamental orthogonal visual geometry. It is shown in figure 1.

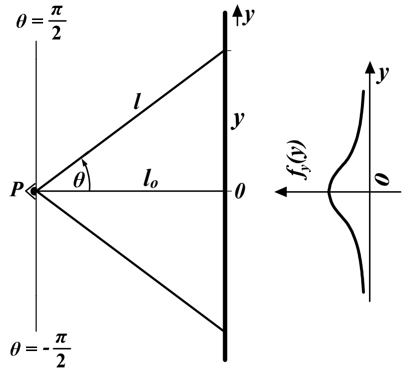


Figure 1: The geometry of visual perception from a top view, where P represents the position of eye, looking at a vertical plane with a distance l_o to the plane; $f_y(y)$ is the probability density function in y-direction.

In figure 1, the observer is facing and looking at a vertical plane from the point denoted by P. By means of looking action the observer pays visual attention equally to all locations on the plane in the first instance. That is, the observer visually experiences all locations on the plane without any preference for one region over another. Each point on the plane has its own distance within the observer's scope of sight which is represented as a cone. The cone has a solid angle denoted by θ. The distance of a point on the plane and the observer is denoted by l and the distance between the observer and the plane is denoted by l_o. Since visual perception is associated with distance, it is straightforward to proceed to express the distance of visual perception l in terms of θ and l_o. From figure 1, this is given by

$$l = \frac{l_o}{\cos(\theta)} \qquad (1)$$

Since we consider that the observer pays visual attention equally to all locations on the plane in the first instance, the probability of getting attention for each point on the plane is the same so that the associated probability density function (pdf) is uniformly distributed. This positing ensures that there is no visual bias at the beginning of visual perception as to the differential visual resolution angle $d\theta$. Assuming the scope of sight is defined by the angle $\theta = \pm \pi/2$, the pdf f_θ is given by

$$f_\theta = \frac{1}{\pi} \qquad (2)$$

since θ is a random variable, the distance x in (1) is also a random variable. The pdf $f_l(l)$ of this random variable is computed as (Ciftcioglu, Bittermann et al.)

$$f_l(l) = \frac{2}{\pi} \frac{l_o}{l\sqrt{l^2 - l_o^2}} \qquad (3)$$

for the interval $l_o \leq l \leq \infty$.

Considering that

$$tg(\theta) = \frac{y}{l_o} \qquad (4)$$

and by means of pdf calculation similar to that to obtain $f_x(x)$ one can obtain $f_y(y)$ as (Ciftcioglu, Bittermann et al.).

$$f_y(y) = \frac{l_o}{\pi(l_o^2 + y^2)} \qquad (5)$$

for the interval $-\infty \leq y \leq \infty$. (9) and (11) are dual representation of the same phenomenon. The probability density functions $f_l(l)$ and $f_y(y)$ are defined as *attention* in the terminology of cognition.

By the help of the results given by (9) and (11) two essential applications in design and robotics are described in a previous research (Bittermann, Sariyildiz et al.). In this research the fundamental orthogonal visual geometry is extended to a general visual geometry to explore the further properties of the perception phenomenon. In this geometry the earlier special geometry the orthogonality condition of the infinite plane is relaxed. This geometry is shown in figure 2 where the attentions at the points O and O' are subject to computation, with the same axiomatic foundation of the probabilistic theory, as before. Since the geometry is symmetrical with respect to x axis, we consider only the upper domain of the axis without loss of the generality.

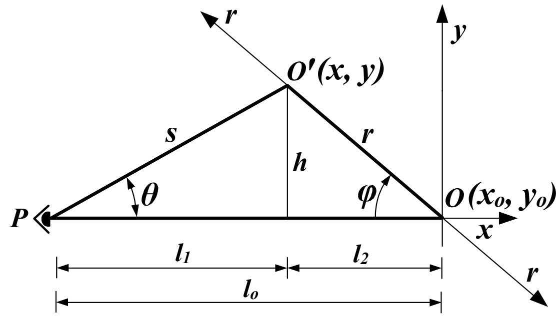


Figure 2: The geometry of visual perception where the observer has the position at point P with the orientation to the point O. The x,y coordinate system has the origin placed at O and the line defined by the points P and O coincides with the x axis for the computational convenience.

In figure 2, an observer at the point P is viewing an infinite plane whose intersection with the plane of page is the line passing from two point designated as O and O'. O represents the origin. The angle between OO' and OP is designated as θ. The angle between OP and OO' is defined by φ. The distance between P and O' is denoted by s and the distance of O' to the OP line is designated as h. The distance of O' to O is taken as a random variable and denoted by r. By means of looking action the observer pays visual attention equally in all directions within the scope of vision. That is, in the first instance, the observer visually experiences all locations on the plane without any preference for one region over another. Each point on the plane has its own distance within the observer's scope of sight which is represented as a cone. The cone has a solid angle denoted by θ. The distance between a point on the plane and the observer is denoted by l and the distance between the observer and the plane is denoted by l_o. Since we consider that the observer pays visual attention equally for all directions within the scope of vision, the associated probability density function (pdf) with respect to θ is uniformly distributed. Positing this ensures that there is no visual bias at the beginning of visual perception as to the differential visual resolution angle $d\theta$. Assuming the scope of sight is defined by the angle $\theta = + \pi/2$, the pdf f_θ is given by

$$f_\theta = \frac{1}{\pi/2} \qquad (6)$$

Since θ is a random variable, the distance r in (1) is also a random variable. The pdf $f_r(r)$ of this random variable is computed as follows.

To find the pdf of the variable r denoted $f_r(r)$ for a given r we consider the theorem on the *function of random variable* and, following Papoulis (Papoulis), we solve the equation

$$r = g(\theta) \tag{7}$$

$$r\,tg\theta + r\,tg\varphi = \frac{l_o\,tg\varphi}{\sin\varphi} tg\theta \tag{18}$$

$$\left(r - l_o \frac{tg\varphi}{\sin\varphi}\right) tg\theta = -r\,tg\varphi \tag{19}$$

$$tg\theta_1 = \frac{r\;\;tg\varphi}{\frac{l_o\,tg\varphi}{\sin\varphi} - r} \tag{20}$$

for θ in terms of r. If θ_1, θ_2 ,...., θ_n , .. are all its *real* roots,

$r=g(\theta_1) = g(\theta_2) = \ldots\ldots = g(\theta_n) = \ldots.$
Then

$$f_r(r) = \frac{f_\theta(\theta_1)}{|g'(\theta_1)|} + \ldots + \frac{f_\theta(\theta_2)}{|g'(\theta_2)|} + \ldots + \frac{f_\theta(\theta_n)}{|g'(\theta_n)|} + .. \tag{8}$$

Aiming to determine $f_r(r)$ given by (8), from figure 2 we write

$$tg\theta = \frac{h}{l_1} \qquad l_1 = \frac{h}{tg\theta} \tag{9}$$

$$tg\varphi = \frac{h}{l_2} \qquad l_2 = \frac{h}{tg\;\;\varphi} \tag{10}$$

$$l_1 + l_2 = \frac{h}{tg\theta} + \frac{h}{tg\varphi} = l_o \tag{11}$$

From above, we solve h, which is

$$h = \frac{l_o}{\frac{1}{tg\theta} + \frac{1}{tg\varphi}} \tag{12}$$

From figure 2, we write

$$r(\varphi) = \frac{h}{\sin\varphi} \tag{13}$$

Using (12) in (13), we obtain

$$r(\varphi) = \frac{l_o}{\sin\varphi} \frac{1}{\frac{1}{\mathrm{tg}\theta} + \frac{1}{\mathrm{tg}\varphi}} = g(\theta) = \frac{l_o\,\mathrm{tg}\varphi}{\sin\varphi} \frac{tg\theta}{tg\theta + tg\varphi} \tag{14}$$

We take the derivative w.r.t. θ, which gives

$$g'(\theta) = \frac{l_o\,tg\varphi}{sin\,\varphi} \frac{\frac{1}{cos^2\theta}(tg\theta + tg\varphi) - \frac{1}{cos^2\theta} tg\theta}{(tg\theta + tg\varphi)^2} \tag{15}$$

$$= \frac{l_o\,tg\varphi}{sin\,\varphi} \frac{tg\varphi \frac{1}{cos^2\theta}}{(tg\theta + tg\varphi)^2} \tag{16}$$

Substituting tg(θ) from (14) into (16) yields

$$\begin{aligned} g'(\theta_1) &= \frac{sin\varphi}{l_o} \frac{1}{\sin^2\theta} r^2 \\ &= \frac{\sin\varphi}{l_o} r^2 \frac{1 + tg^2\theta_1}{tg^2\theta_1} \\ &= \frac{\sin\varphi}{l_o} r^2 \left(1 + \frac{1}{tg^2\theta_1}\right) \end{aligned} \tag{17}$$

Above, $tg(\theta_1)$ is computed from (14) as follows.

We apply the theorem of *function of random variable* (Papoulis):

$$f_r(r) = \frac{f_r(\theta_1)}{|g'(\theta_1)|} \tag{21}$$

$$\mathrm{f_r(r)} = \frac{1}{\frac{\pi}{2}\sin\varphi} \frac{\mathrm{l_o}}{\mathrm{r}^2\left(1 + \frac{1}{\mathrm{tg}^2\theta_1}\right)} \tag{22}$$

where

$$\mathrm{tg}\theta_1 = \frac{r}{\frac{l_o}{\sin\varphi} - \frac{r}{tg\varphi}} \tag{23}$$

Substitution of (23) into (22) gives

$$f_r(r) = \frac{1}{\frac{\pi}{2}|\sin\varphi|} \frac{l_o}{\left| r^2 + \left(\frac{l_o}{\sin\varphi} - \frac{r}{tg\varphi}\right)^2 \right|} \tag{24}$$

or

$$f_r(r) = \frac{l_o}{\frac{\pi}{2}} \frac{\sin(\varphi)}{r^2 - 2l_o r\cos\varphi + l_o^{\;2}} \tag{25}$$

for $$0 < r < \frac{l_0}{\cos(\varphi)}. \tag{26}$$

To show (25) is a pdf, we integrate it in the interval given by (26). The second degree equation at the denominator of (25) gives
$b^2 - 4ac = -4l_o^2 \sin^2(\varphi)$

where $b=2l_or$ and $a=1$ and $c=l_o^2$, so that $b^2 < 4ac$ which means, for this the integral

$$I = \int_0^{l_0/\cos(\varphi)} f_r(r,\varphi)\,dr \tag{27}$$

gives (Korn and Korn)

$$I = \frac{1}{\pi/2} arctg\{\frac{r - l_o\cos(\varphi)}{l_o\sin(\varphi)}\}\Big|_o^{l_o/\cos(\varphi)} = 1 \tag{28}$$

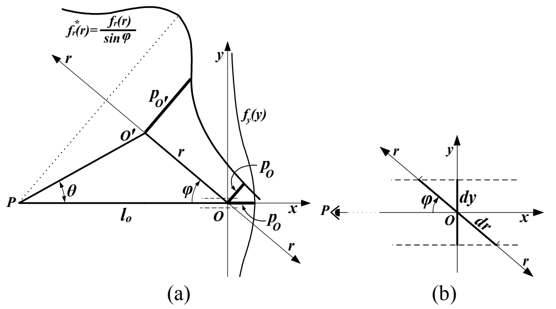
as it should verify as pdf. Since attention is a scalar quantity per unit, it has to be the same for different geometries subjected to computation meaning that it is measured with the same units in both cases. In the same way we can say that since perception is a scalar quantity, the perceptions have to be correspondingly the same. Referring to both the orthogonal geometry and the general geometry, the density functions are shown in figure 3a. The same attention values at the origin O are denoted by p_o. Since the attentions are the same, for the perception comparison, the attention values have to be integrated within the same intervals in order to verify the same quantities at the same point. Figure 3b is the magnified portion of figure 3a in the vicinity of origin. In this magnified sketch the infinitesimally small distances dy and dr are indicated where the relation between dy and dr is given by $dy = dr\sin(\varphi)$ or

$$dr = dy/\sin(\varphi) \tag{29}$$

and

$$f_r(r)dr = \frac{l_o}{\frac{\pi}{2}} \frac{\sin(\varphi)}{r^2 - 2l_o r\cos\varphi + l_o^2} dr \tag{30}$$

substitution of (29) into (30) yields



(a) (b)

Figure 3: Illustration of the perception in the orthogonal geometry and the general geometry indicating the relationship between the infinitesimally small distances dy and dr. The geometry (a) and zoomed region at the origin (b).

$$f_r^*(r) = \frac{l_o}{\frac{\pi}{2}} \frac{1}{r^2 - 2l_o r\cos\varphi + l_o^2} \tag{31}$$

which is the attention for a general geometry. It boils down into the orthogonal geometry for all conditions; for a general position of O' within the visual scope, this is illustrated in figure 4 as this was already illustrated in figure 3 for the origin O.

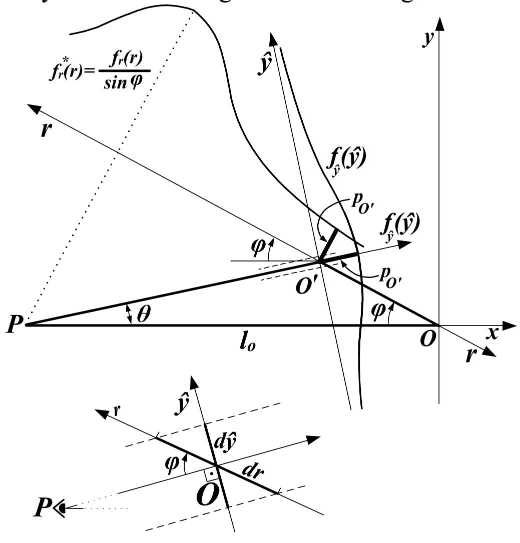


Figure 4: Illustration of the perception in the orthogonal geometry and the general geometry indicating the relationship between the infinitesimally small distances $d\hat{y}$ and dr. This is the same as figure 3 but the zoomed region is at a general point denoted by O'.

The pdf has several interesting features. First, for $\varphi=\pi/2$, it boils down

$$f_y(r) = \frac{l_o}{\frac{\pi}{2}} \frac{1}{\left(r^2 + l_o^2\right)} \tag{32}$$

An interesting point is that when $\varphi \to 0$ but $r \neq 0$. This means O' is on the gaze line from P to O. For the case O' is between P and O, $f_r(r)$ becomes

$$f_r(r) = \frac{l_o}{\pi} \frac{1}{(l_o - r)^2} \tag{33}$$

or otherwise

$$f_r(r) = \frac{l_o}{\pi} \frac{1}{(l_o + r)^2} \tag{34}$$

In (33) for $r \to l_0$ $f_r(r) \to \infty$. This case is similar to that in (3) where $l \to l_0$ $f_l(l) \to \infty$.

The variation of $f_r^*(r,\varphi)$ is shown in figure 5 in a 3-dimensional plot where φ is a parameter.

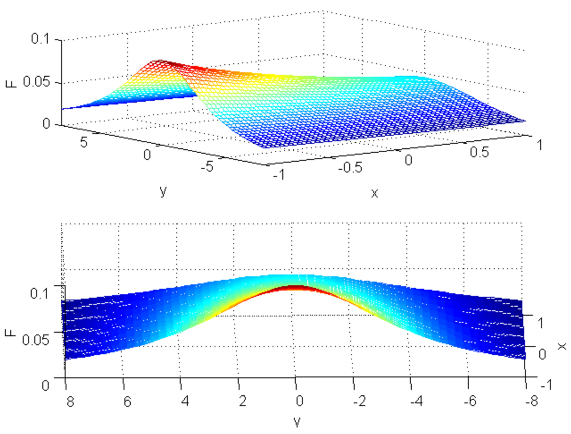


Figure 5: The variation of $f_r^*(r,\varphi)$ is shown as a 3-dimensional plot for $l_o=5$.

The actual $f_r(r)$ is obtained as the intersection of a vertical plane passing from the origin O and the surface. The analytical expression of this intersection is given by (25) and it is shown in figure 4 where φ is a parameter; for the upper plot $\varphi=\pi/4$ and for the lower plot $\varphi=\pi/2$. The latter corresponds to the vertical cross section of the surface shown in figure 3 as lower plot.

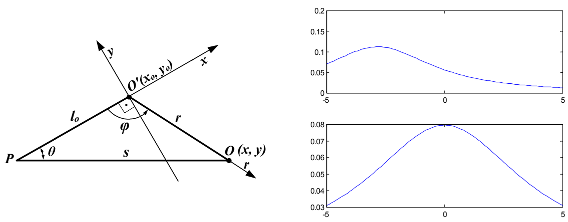


Figure 6: The pdf $f_r(r,\varphi)$ where φ is a parameter; for the upper plot $\varphi=\pi/4$ and for the lower plot $\varphi=\pi/2$.

The pdf in (25) indicates the attention variation along the line r in figure 2 where the observer faces the point O.

3 APPLICATION

Presently, the experiments have been done with the simulated measurement data since the multiresolutional filtering runs in a computationally efficient software platform which is different than the computer graphics platform of virtual reality. For the simulated measurement data, first the trajectory of the virtual agent is established by changing the system dynamics from the straight ahead mode to bending mode for a while, three times. Three bending modes are seen in figure 7 with the complete trajectory of the perceptual agent. The state variables vector is given by

$$X = [x, \dot{x}, y, \dot{y}, \omega]$$

where ω is the angular rate and it is estimated during the move. When the robot moves in a straight line, the angular rate becomes zero.

In details, there are three lines plotted in figure 7. The green line represents the measurement data set. The black line is the extended Kalman filtering estimation at the highest resolution of the perception measurement data. The outcome of the multiresolutional fusion process is given with the blue line. The true trajectory is indicated in red. In this figure they cannot be explicitly distinguished. For explicit illustration of the experimental outcomes the same figure with a different zooming range and the zooming power are given in figures 8 and 9 for bending mode and 10 for a straight-ahead case. From the experiments it is seen that, the Kalman filtering is effective for estimation of the trajectory from perception measurement. Estimation is improved by the multiresolutional filtering. Estimations are relatively more accurate in the straight-ahead mode.

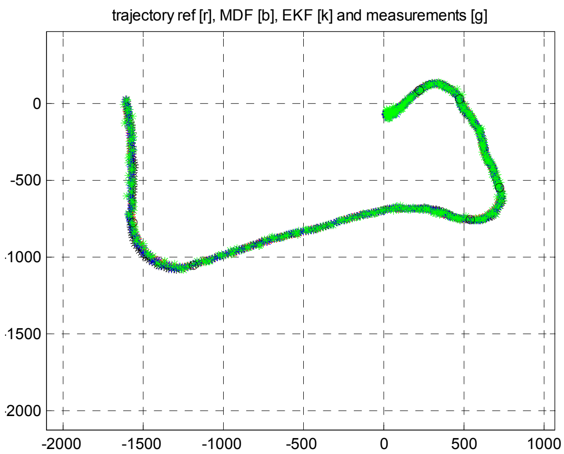


Figure 7: Robot trajectory, measurement, Kalman filtering and multiresolutional filtering estimation.

It is noteworthy to mention that, the multiresolutional approach presented here uses *calculated* measurements in the lower resolutions. In general case, each sub-resolution can have separate perception measurement from its own dedicated perceptual vision system for more accurate executions. The multiresolutional fusion can still be improved by the use of different data acquisition provisions which play the role of different sensors at each resolution level and to obtain independent information subject to fusion.

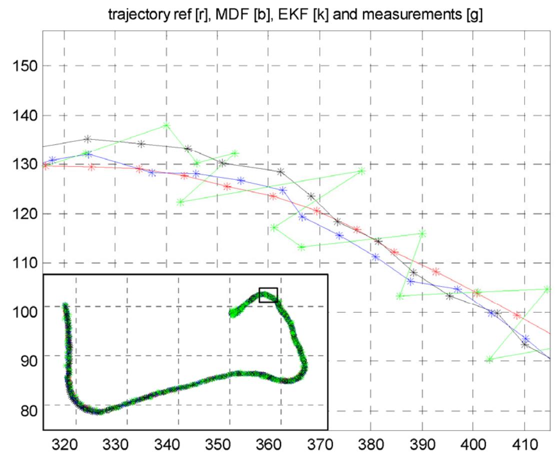


Figure 8: Enlarged Robot trajectory, measurement Kalman filtering and multiresolutional filtering estimation, in bending mode (light grey is for measurement, the smooth line is the trajectory).

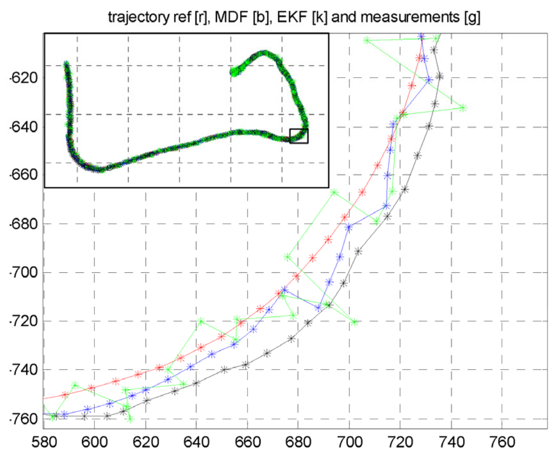


Figure 9: Enlarged Robot trajectory, measurement Kalman filtering and multiresolutional filtering estimation, in bending mode (light grey is for measurement, the smooth line is the trajectory).

4 DISCUSSION AND CONCLUSION

Although, visual perception is commonly articulated in various contexts, generally it is used to convey a cognition related idea or message in a quite fuzzy form and this may be satisfactory in many instances. Such usage of perception is common in daily life. However, in professional areas, like architectural design or robotics, its demystification or precise description is necessary for proficient executions. Since the perception concept is soft and thereby elusive, there are certain difficulties to deal with it. For instance, how to quantify it or what are the parameters, which play role in visual perception. The positing of this research is that perception is a very complex process including brain processes. In fact, the latter, i.e., the brain processes, about which our knowledge is highly limited, are final, and therefore they are most important. Due to this complexity a probabilistic approach for a visual perception theory is very much appealing, and the results obtained have direct implications which are in line with our common visual perception experiences, which we exercise every day.

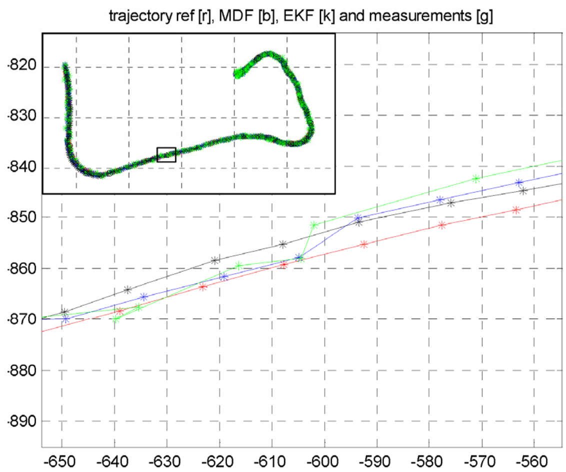


Figure 10: Enlarged Robot trajectory, measurement Kalman filtering and multiresolutional filtering estimation in straight-ahead mode (light grey is for measurement, the smooth line is the trajectory).

In this work a novel theory of visual perception is developed, which defines perception in probabilistic terms. The probabilistic approach is most appropriate, since it models the complexity of the brain processes, which are involved in perception and result in the characteristic uncertainty of perception, e.g., an object may be overlooked although it is visible. Based on the constant differential angle in human vision, which is the minimal angle humans can visually distinguish, vision is defined as the ability to see, that is, to receive information, which is transmitted via light, from different locations in the environment, which are located within different differential angles. This ability is modeled by a function of a random variable, namely the viewing direction, which has a uniform probability density for the direction, to model unbiased vision in the first instance. Hence vision is defined as probabilistic act. Based on vision, visual attention is defined as the corresponding probability density with respect to obtaining information from the environment. Finally, the visual perception is the intensity of attention, which is the integral of attention over a certain unit length, yielding a probability that the environmental information from a region in the environment is realized in the brain.

It is noteworthy to emphasize that perception is to be expressed in terms of intensity, which is the integral of a probability density. This is not surprising since perception, corresponding to its commonly understood status as a mental event, should be a dimensionless quantity, as opposed to a concept, which involves a physical unit, namely a probability density over a unit length, like visual attention. The definitions are conforming to common perception experience by human. The simplicity of the theory in terms of understanding its result together with its explanatory power, indicates that a fundamental property of perception has been identified.

In this theory of perception a clear distinction is made between the act of *perceiving* and *seeing*. Namely, seeing is a definitive process, whereas perception is a probabilistic process. This distinction may be a key to understand many phenomena in perception, which are challenging to explain from a deterministic viewpoint. For example the theory explains the common experience, that human beings may overlook an object while searching for it, although such an overlooking is not justified, and it is difficult to explain the phenomenon. This can be understood from the viewpoint that vision is a probabilistic act, where there exists a chance that corresponding visual attention is not paid sufficiently for the region in the environment, which would provide the pursued information. An alternative explanation, which is offered by an information theoretic interpretation of the theory, is that through the integration of the visual attention over a certain domain some information may be lost, so that, although attention was paid to a certain item in the environment, pursued information is not obtained. The theory also explains how it is possible, that different individuals have different perceptions in the same environment. Although similar viewpoints in the same environment have similar visual attention with unbiased vision, the corresponding perception remains a phenomenon of probability, where a realization in the brain is not certain, although it may be likely.

The theory is verified by means of extensive computer experiments in virtual reality. From visual perception, other derivatives of it can be obtained, like visual openness perception, visual privacy, visual color perception etc. In this respect, we have focused on visual openness perception, where the change from visual perception to visual openness perception is accomplished via a mapping function and the work is reported in another publication (Ciftcioglu, Bittermann et al.). Such perception related experiments have been carried out by means of a virtual agent in virtual reality, where the agent is equipped with a human-like vision system (Ciftcioglu, Bittermann et al.).

Putting perception on a firm mathematical foundation is a significant step with a number of far reaching implications. On one hand vision and perception are clearly defined, so that they are understood in greater detail, and reflections about them are substantiated. On the other hand tools are developed to employ perception in more precise terms in various cases and even to measure perception. Applications for perception measurement are architectural design, where they can be used to monitor implications of design decisions, and autonomous robotics, where the robot moves based on perception (Ciftcioglu, Bittermann et al.).

REFERENCES

Adams, B., C. Breazeal, et al., 2000. Humanoid robots: a new kind of tool, *Intelligent Systems and Their Applications, IEEE [see also IEEE Intelligent Systems]* 15(4): 25-31.

Ahle, E. and D. Söffker, 2006. A cognitive-oriented architecture to realize autonomous behaviour – part I: Theoretical background. *2006 IEEE Conf. on Systems, Man, and Cybernetics*, Taipei, Taiwan.

Ahle, E. and D. Söffker, 2006. A cognitive-oriented architecture to realize autonomous behaviour – part II: Application to mobile robots. *2006 IEEE Conf. on Systems, Man, and Cybernetics*, Taipei, Taiwan.

Arbib, M. A., 2003. *The Handbook of Brain Theory and Neural Networks*. Cambridge, MIT Press.

Beetz, M., T. Arbuckle, et al., 2001. Integrated, plan-based control of autonomous robots in human environments, *IEEE Intelligent Systems* 16(5): 56-65.

Bigun, J., 2006. *Vision with direction*, Springer Verlag.

Bittermann, M. S., I. S. Sariyildiz, et al., 2006. Visual Perception in Design and Robotics, *Integrated Computer-Aided Engineering* to be published.

Burghart, C., R. Mikut, et al., 2005. A cognitive architecture for a humanoid robot: A first approach. *2005 5th IEEE-RAS Int. Conf. on Humanoid Robots*, Tsukuba, Japan.

Cataliotti, J. and A. Gilchrist, 1995. Local and global processes in surface lightness perception, *Perception & Psychophysics* 57(2): 125-135.

Ciftcioglu, Ö., M. S. Bittermann, et al., 2006. Autonomous robotics by perception. *SCIS & ISIS 2006, Joint 3rd Int. Conf. on Soft Computing and Intelligent Systems and 7th Int. Symp. on advanced Intelligent Systems*, Tokyo, Japan.

Ciftcioglu, Ö., M. S. Bittermann, et al., 2006. Studies on visual perception for perceptual robotics. *ICINCO 2006 - 3rd Int. Conf. on Informatics in Control, Automation and Robotics*, Setubal, Portugal.

Ciftcioglu, Ö., M. S. Bittermann, et al., 2006. Towards computer-based perception by modeling visual perception: a probabilistic theory. *2006 IEEE Int. Conf. on Systems, Man, and Cybernetics*, Taipei, Taiwan.

Desolneux, A., L. Moisan, et al., 2003. A grouping principle and four applications, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25(4): 508-513.

Garcia-Martinez, R. and D. Borrajo, 2000. An integrated approach of learning, planning, and execution, *Journal of Intelligent and Robotic Systems* 29: 47-78.

Ghosh, K., S. Sarkar, et al., 2006. A possible explanation of the low-level brightness-contrast illusions in the light of an extended classical receptive field model of retinal ganglion cells, *Biological Cybernetics* 94: 89-96.

Hecht-Nielsen, R., 2006. The mechanism of thought. *IEEE World Congress on Computational Intelligence WCCI 2006, Int. Joint Conf. on Neural Networks*, Vancouver, Canada.

Hubel, D. H., 1988. *Eye, brain, and vision*, Scientific American Library.

Itti, L., C. Koch, et al., 1998. A model of saliency-based visual attention for rapid scene analysis, *IEEE Trans. on Pattern Analysis and Machine Intelligence* 20(11): 1254-1259.

Korn, G. A. and T. M. Korn, 1961. *Mathematical handbook for scientists and engineers*. New York, McGraw-Hill.

Levine, M. W. and J. M. Sheffner, 1981. *Fundamentals of Sensation and Perception*. London, Addison-Wesley.

Marr, D., 1982. *Vision*, Freeman.

O'Regan, J. K., H. Deubel, et al., 2000. Picture changes during blinks: looking without seeing and seeing without looking, *Visual Cognition* 7: 191-211.

Oriolio, G., G. Ulivi, et al., 1998. Real-time map building and navigation for autonomous robots in unknown environments, *IEEE Trans. on Systems, Man and Cybernetics - Part B: Cybernetics* 28(3): 316-333.

Palmer, S. E., 1999. *Vision Science*. Cambridge, MIT Press.

Papoulis, A., 1965. *Probability, Random Variables and Stochastic Processes*. New York, McGraw-Hill.

Pentland, A., 1987. A new sense of depth, *IEEE Trans. on Pattern Analysis and Machine Intelligence* 9: 523-531.

Poggio, T. A., V. Torre, et al., 1985. Computational vision and regularization theory, *Nature* 317(26): 314-319.

Posner, M. I. and S. E. Petersen, 1990. The attention system of the human brain, *Annual Review of Neuroscience* 13: 25-39.

Prince, S. J. D., A. D. Pointon, et al., 2002. Quantitative analysis of the responses of V1 Neurons to horizontal disparity in dynamic random-dot stereograms, *J Neurophysiology* 87: 191-208.

Rensink, R. A., J. K. O'Regan, et al., 1997. To see or not to see: The need for attention to perceive changes in scenes, *Psychological Science* 8: 368-373.

Söffker, D., 2001. From human-machine-interaction modeling to new concepts constructing autonomous systems: A phenomenological engineering-oriented approach., *Journal of Intelligent and Robotic Systems* 32: 191-205.

Taylor, J. G., 2006. Towards an autonomous computationally intelligent system (Tutorial). *IEEE World Congress on Computational Intelligence WCCI 2006*, Vancouver, Canada.

Treisman, A. M., 2006. How the deployment of attention determines what we see, *Visual Cognition* 14(4): 411-443.

Treisman, A. M. and G. Gelade, 1980. A feature-integration theory of attention, *Cognitive Psychology* 12: 97-136.

Wang, M. and J. N. K. Liu, 2004. Online path searching for autonomous robot navigation. *IEEE Conf. on Robotics, Automation and Mechatronics*, Singapore.

Wiesel, T. N., 1982. Postnatal development of the visual cortex and the influence of environment (Nobel Lecture), *Nature* 299: 583-591.

KAMANBARÉ[1]
A Tree-climbing Biomimetic Robotic Platform for Environmental Research

Reinaldo de Bernardi
Genius Institute of Technology, São Paulo, Brazil
Department of Telecommunications and Control, São Paulo University, São Paulo, Brazil
rbernardi@genius.org.br

José Jaime da Cruz
Department of Telecommunications and Control, São Paulo University, São Paulo, Brazil
jaime@lac.usp.br

Keywords: Climbing robots, legged robots, environmental research.

Abstract: Environmental research is an area where robotics platforms can be applied as solutions for different problems to help or automate certain tasks, with the purpose of being more efficient or also safer for the researchers involved. This paper presents the Kamanbaré platform. Kamanbaré is a bioinspired robotic platform, whose main goal is to climb trees for environmental research applications, applied in tasks such as gathering botanical specimen, insects, climatic and arboreal fauna studies, among others. Kamanbaré is a platform that provides flexibility both in hardware and software, so that new applications may be developed and integrated without the need of extensive knowledge in robotics.

1 INTRODUCTION

Robotic platforms can be used in countless applications and in the most varied branches of activities. Such results, presented over the last two decades, can be verified in (Armada et al., 2003) and (Virk, 2005).

It deals specifically with robots provided with legs and with the capability, or ability, to climb vertical surfaces, while many other applications/solutions can be found. As an example, one can mention: robots to climb lower parts of bridges (Abderrahim et al., 1999), to crawl inside pipes for inspection purposes (Galvez, Santos, Pfeiffer, 2001), implemented to perform solder inspection works in nuclear plants (White et al., 1998), to climb metallic structures for inspection purposes (Armada et al., 1990). More examples can be found in marine industry applications, such as walking robots to check internal parts solders in ship hulls, climbing robots for parts solders (Santos, Armada, Jiménez, 2000) and (Armada et al., 2005), climbing robots for paint cleaning, underwater robots for ballast tank inspection, and underwater robots for hull cleaning (Santos, Armada, Jiménez, 1997a, 1997b).

Robots were demonstrated to be the ideal option for many such applications due to the fact that the working environment is difficult to access or even hazardous or risky for human beings, such as exposure to hazardous substances or environments and risk conditions. Productivity increase and quality issues are also extremely relevant and are considered.

However, besides the varied applications and areas mentioned above, there still remains a little explored area: environmental research. As in any other area, different applications or problems can be addressed or solved with the help of a robotics platform. As an example, one can mention the activities:

Gathering of Botanical Specimens: gathering flower and plant specimen is fundamental for biodiversity studies. Several host species are found in high trees, and their collection is actually risky. Thus, this is an application where a robotics platform with tree climbing capability can be used to minimize risks for the researchers involved in this type of activity.

[1] Kamanbaré means chameleon, on the language of the Brazilian Tupi indians.

Gathering of vegetable material: collection of vegetable material is a fundamental activity for phytochemistry. Every pharmaceutical preparation that uses as raw material plant parts such as leaves, stems, roots, flowers, and seeds, with known pharmacological effect is considered phytotherapic (extracts, dyes, ointments and capsules). Thus, just as in botanical collection, this is an activity that can be accomplished by a robot operating in large-sized trees, therefore minimizing risk for the humans involved in the collection.

Gathering of Insect Specimens: usually, for collecting insects in higher levels, nets spread around the tree and a source of smoke below it are used. The great discussion about this technique is that one not only captures the desired insects, but one ends up killing most of the insects that inhabit that tree as well. Thus, one proposes to use a trap, positioned in the robot, containing a pheromone or equivalent substance as bait, specific for the insect species one wants to capture. One can even adapt cameras and sensors in the trap to gather images of the moment of capture. With a trap thus implemented, one reduces the negative environmental impact of the capture. Another relevant issue would be the possibility that the robot moves along the day to capture varied insect specimens at different heights in different hours, to check for variations in insect populations according to the height, and the hour of the day.

Climatic studies: climatic or microclimatic studies refer to works on microenvironments in native vegetation and/or reforested areas. It is important to study the different energy flows: horizontal and vertical. The vertical directly reflects the results of solar radiation, which decisively influences the horizontal energy flows: air masses, hot and cold fronts, action centers. Solar radiation determines the whole system, and may be analized according to its elements: temperature, pressure, and humidity, greatly influencing biogeographic characteristics, geomorphologic and hydrologic phenomena etc. Thus, the robot can be equipped with sensors for such measures, to collect data on the desired elements.

Studies on biosphere/atmosphere interaction: biosphere environments are the group of biotic or abiotic factors that interfere in the life conditions on a certain region of the biosphere. In the case of forests the aerial environment is studied, and the most important elements to consider are: light, oxygen, ice formation, winds, humidity and carbon gas. In order to register all this information, the robot can be endowed with specific sensors for each kind of required measure, to collect data regarding the elements at hand, and to provide information on them regarding both height and time variations.

Studies on arboreal fauna: fauna studies are hampered by the existence of many leaves, or very dense treetops, as the lack of existing natural light hinders the observation of the species. Other usually relevant points are the difficulty to obtain a proper angle due to the great heights involved, and the very presence of human beings in that specific environment, easily detected by their movements, noise and odors. For this type of task, the robot can be fitted with cameras to capture both static and dynamic images. These can then be stored locally in some type of memory card, or transmitted via communication interface to a base station.

Sensors Network: robots carrying a group of sensors and fitted with a communication interface, for instance Wi-fi or other similar technology can be dispersed in the forest to capture data regarding the ecological behavior in the area at hand. Measurements such as the ones already mentioned in climatic studies and biosphere/atmosphere interaction can be shared among robots or even retransmitted among robots to reach the base station, without the need for the researcher to “pay visits” to the reading points.

2 GOAL AND PURPOSE

Thus, considering this poorly explored area, one proposes the Kamanbaré robotics platform. Kamanbaré is a biomimetic robot, i.e., inspired in nature, with the purpose of climbing trees for environmental research applications. The proposed work represents a progress in robot applications, both for the fact of environmental research applications, and for its tree-climbing ability, with computer-controlled devices configured to be used as paws.

The project’s main application is climbing trees for non-invasive search purposes, reaching points (at high altitudes) that may offer risk to humans.

The development was driven mainly to seek for robot stability and efficiency regarding the paws. The adopted line of research is an implementation inspired in nature. One must stress here that the system doesn't mimic nature, by copying an animal or insect to accomplish the desired work, but rather, a robot development project that combines the climbers' best characteristics (insects and lizards) considering the available technologies and the associated cost/benefit ratio, in other words, some parts were inspired in certain solutions found in nature, while other parts were inspired in different elements.

Ensemble stability, contact stability (paws), contact position, and contact force (pressure) were defined and used to implement the strategies and techniques for dynamic and static control of the system.

3 BIOLOGICAL INSPIRATION: THE CHAMELEON

Differently from any other reptile or lizard, chameleons have paws that were created, or adapted, to adequately clasp the different types of branches or shrubs existing in its environment.

The chameleon paws evolved to provide them with the best possible maneuvering and grip capabilities on trees. The paws are actually forked, with three fingers on one side and two on the other. Frequently they also have powerful and sharp nails (or claws) that can grip the surface which they hold on to. This unique arrangement allows them to position their paws completely around the branches or shrubs on which they move around, giving them an amazingly strong clasping capability.

Chameleon paws have five fingers each, divided in two groups (internal and external), one side composed of three fingers while the other side has two, as seen in Figure 1. Front paws have two fingers pointing to the external side, and three towards the inner side, while rear paws are configured in an opposite arrangement, i.e., two inside fingers and three outside fingers. This provides the chameleon with the same number of fingers on each side of the branch, considering all the paws, allowing a balanced and stable clasping.

Figure 1: Details of the chameleon paw, presenting the bifurcation and configuration of the fingers.

4 GEOMETRY OF THE MECHANICAL MODEL

The mechanical structure of the Kamanbaré platform consists of a central rigid body with four legs, identical and symmetrically distributed. Each leg comprises three links connected by two rotating joints, and is connected to the central body by a third rotating joint, while each joint has 1 DOF (degrees of freedom). Identical motor and reduction groups make the rotary movements. Figure 2 shows the kinematic configuration of a leg. Each leg also has a paw, which is forked just as the chameleons, however with only two fingers, one on each side, for simplification purposes. The paw connects to the leg by a rotating joint, and also has another motor and reduction group that provides for its opening and closing movements.

Potentiometers and microswitches were inserted in the joints to supply the signal corresponding to the opening and closing angle, or position, for the links. In the paws, more precisely in their plant, a microswitch was also added to supply the contact signal with the surface of the object to be climbed during the movements of the legs.

As each leg has four joints, this means 4 DOF. Considering the four legs, the platform will have a total of 16 DOF. Therefore, sixteen motor and reduction groups were necessary to produce the global movements.

The prototype of the Kamanbaré platform presented in this work was developed considering certain capabilities (abilities), such as: locomotion in irregular environments (unpredictability of the branch complexity that compose a tree), surmounting obstacles (nodes and small twigs), tree climbing and descent without risking stability, and keeping low structural weight (mechanics + electronics + batteries).

Platform development began by considering the mechanical requirements (structures, materials, implementation complexity, costs etc.) in parallel with the electronic requirements (hardware and software tools, development time, knowledge of the tools, components availability, costs etc.).

Figure 2: Kinematic configuration of a leg.

The main dimensions of the platform are: length of 250 mm, and width varying between 260 and 740 mm, depending on the positioning of the legs. Its height also depends on the posture and positioning of the legs, and can vary between 190 and 310 mm.

This configuration, with all the parts implemented in aluminum, comprises an approximate total weight of 0.6 kg, not including the batteries. The geometric model obtained can be seen in Figure 3. This model was used to generate the basic locomotion behavior.

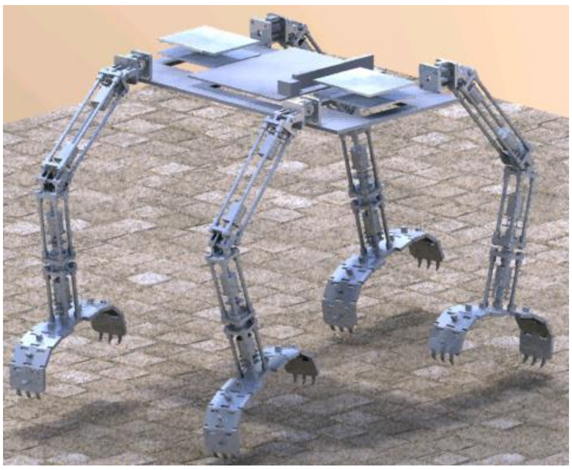


Figure 3: Mechanical structure of the Kamanbaré platform.

5 BASE STATION ARCHITECTURE

The Base Station provides mission control functions, sending and receiving data to/from the Kamanbaré platform via a communication interface such as Wi-fi.

The station also provides mission start and end parameters, as well as commands for moving the robot.

A graphic interface was implemented for a better display of the data received from the platform, including a window for image reception, when appropriate.

Data and command inputs are enabled via keyboard or, for the robot's control, through a joystick.

The possibility of using speech recognition commands was also considered. As the platform has a motherboard with enough processing capacity and a Linux operating system, a speech recognition module can be integrated and commands sent directly, without the Base Station, via microphone with a Bluetooth-type serial interface.

6 KAMANBARÉ PLATFORM ARCHITECTURE

An architecture was implemented for local control of the Kamanbaré platform. This architecture corresponds to the robot's functional organization.

Based on the hardware architecture to be presented in the following section, the development of the following systems was accomplished according to Figure 4. This model is based on the architecture implemented for the MARIUS robot (PASCOAL et al., 1997).

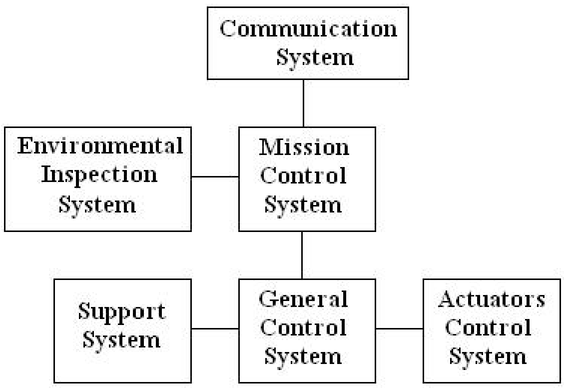


Figure 4: Kamanbaré Architecture.

Support system: this system controls energy distribution to the platform's electronic and electromechanic hardware, and monitors energy consumption as well. This system is also responsible for the startup of other subsystems and, during operation, for detecting hardware failures, and for starting and controlling the emergency modes.
Actuators control system: this system is responsible for controlling the motors, and also for controlling the movements of the legs. Information on legs positioning is received from the general control system. Data regarding joint positions, as well as the values of the electric currents involved, are sent to the general control system.
General control system: this system receives trajectory reference information from the mission control system. It controls all the robot's movements, sending the necessary commands to the actuators control system. Problems occurring in the path, such as obstacles and absence of support points for the paws, are handled in this system.
Mission control system: this system is the main module, the highest hierarchical level of the platform. It is responsible for receiving commands via the communications system, and for distributing them to the systems involved. It also stores information on the general status of the platform (battery voltage, position of the legs, angles etc.) keeping them available in case the Base Station (described in the following topics) requests them. This system gathers information from the Environmental inspection system to be subsequently forwarded to the Base Station.
Communication system: this system is the module responsible for the communication interfaces existing in the platform, managing communications via Wi-fi and Bluetooth, and exchanging data with the Mission control system.
Environmental inspection system: this system is responsible for gathering data from the installed sensors, and for controlling any additional hardware necessary for that purpose as well. Every data acquired are sent to the Mission control system.

7 ELECTRONIC ARCHITECTURE

Considering the implementation of control algorithms, processing information from sensors, the necessary calculations for control and locomotion strategies, interfacing and communication with the base station, added to the need of real time control with position and force feedback, the involvement of high computing complexity was ascertained, thus requiring a processor of advanced architecture.

Eventually, it was selected then a development kit containing a processor based on the ARM9 core, and the deployment of a Linux operating system.

Other motor control boards were also developed using a Texas Instruments microcontroller of the MSP430 family and specific integrated circuits to implement the power actuation section, based on the so-called H-bridge technique.

To implement the control systems for the Kamanbaré platform, an electronic architecture was defined. Initially considering only one joint, it can be represented in Figure 5, where the main components are seen: a DC motor, a potentiometer and a microswitch.

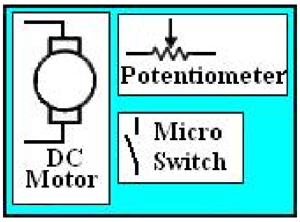


Figure 5: Representation of a joint.

Thus, for control purposes, the need of a PWM output (motor control), an analog input (potentiometer reading, indicating the joint angle), and a digital input (reading the end, or beginning, of the joint course) was ascertained.

As already mentioned, the platform has 16 DOF, corresponding to the need of sixteen copies of the joint control system described above.

As the robot has four legs, one opted for distributing the control individually to each one of them. Thus, each leg control module needs four groups as mentioned, namely, three for the joints, and one for controlling the opening and closing of the claw.

One then developed a motor control board for this specific purpose, Figure 6, based on the MSP430F149 Texas Instruments microcontroller, and the L298 integrated circuit (H-bridge). The board dimensions are: 60 x 50 mm.

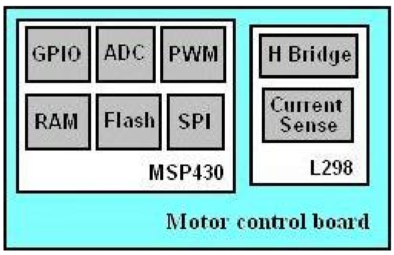


Figure 6: Motor control board diagram.

Due to the control complexity existing in a robotics platform, it was necessary to adopt a main board where the highest hierarchical level control activities were executed.

As a solution for the main board, the model selected was the TS-7250 by Technologic Systems. It was selected because it's compact, contains different standard interfaces, and is based on the EP9302 Cirrus Logic processor, with an ARM9 core, Figure 7. The EP9302 implements an advanced processor core: 200 MHz ARM920T with support for a memory management unit (MMU). This ensemble allows the use of a high-level operating system, in this case Linux. The ARM920T core has a 32-bit architecture with a 5-stage pipeline, offering high performance and low energy consumption levels. With a 200 MHz clock, the TS-7250 module offers a performance approximately twice as fast as other boards based on 586-core processors.

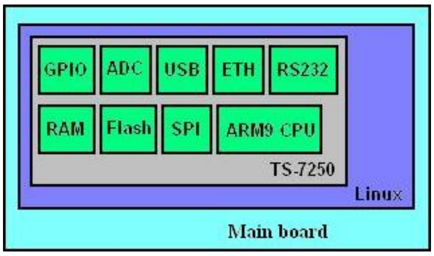


Figure 7: Main board diagram.

Thus, the general electronic architecture for the Kamanbaré platform was deployed according to the diagram in Figure 8.

8 IMPLEMENTATION

The control structure described in the previous section was implemented for the Kamanbaré platform. The robot has contact and force sensors in each joint (actually, in the case of the force sensor, the electric current of the corresponding motor will be used for this purpose), contact sensors in the paws, tilt sensor, sensors to measure the distance or proximity of the body from the surface, and possibly an accelerometer, intended to measure the displacement speed, and also to identify a potential fall.

C language was used for the development of all software, mainly for reasons of code execution speed, and easiness of portability in case some other control board is needed.

A concerning point is the contact detection via force sensors in the joints. The issue here is that these virtual sensors only present readings when the motors are in motion. Thus, whenever a reading is necessary, a movement must be initiated.

Regarding the movement generation module, it was initially implemented only the simplest way, i.e., the robot follows a straight path at the highest

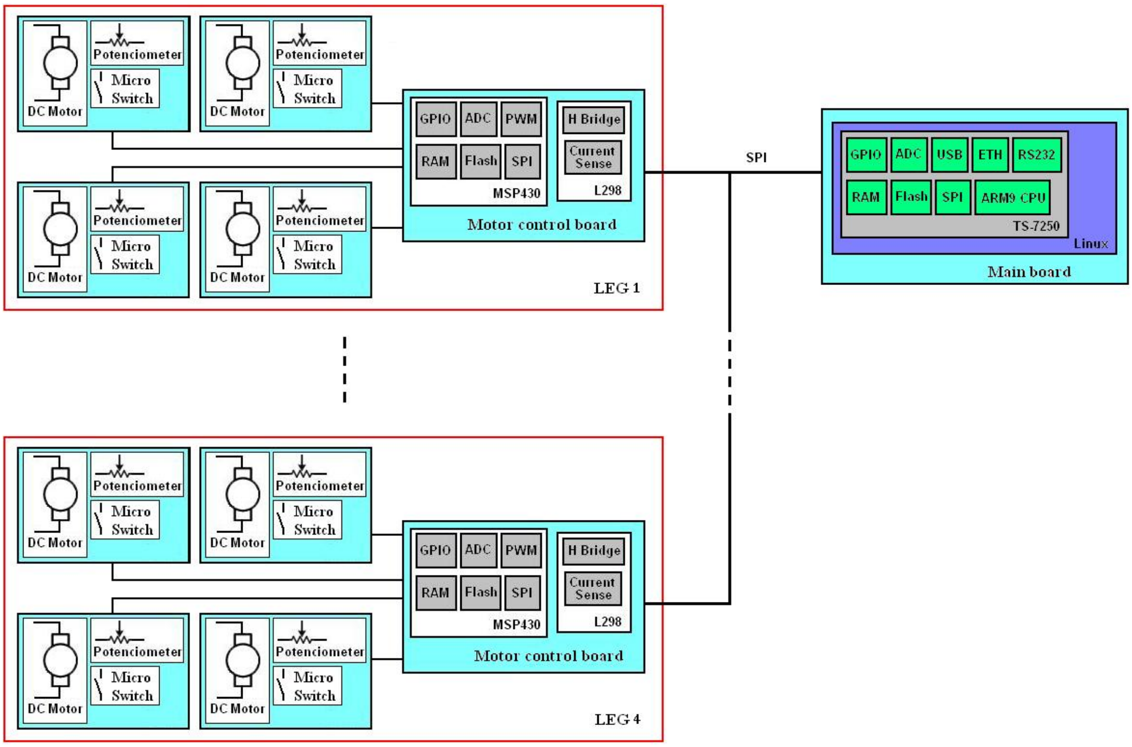


Figure 8: Electronic architecture of the Kamanbaré platform.

speed allowed by the surface (no waiting time will be introduced, besides those produced by the legs when searching for support points).

Steering and speed commands are provided via base station and, in the future, via speech commands over a microphone with Bluetooth technology.

9 CONCLUSION

This work presented the Kamanbaré robot, which is a four-legged bionspired platform with the main purpose of climbing trees, for environmental research applications. The mechanical and electronics structure, and the software architecture, were presented.

REFERENCES

Abderrahim, M., Balaguer, C., Giménez, A., Pastor, J. M., Padrón, V. M., 1999. *ROMA: A Climbing Robot for Inspection Operations*. Proceedings of the IEEE International Conference on Robotics and Automation, Volume 3, 10-15, Detroit, USA.

Armada, M., Santos, P. G. de, Nieto, J., Araujo, D. *On the design and control of a self-propelling robot for hazardous environments*. 1990. In Proc. 21st Int. Symposium on Industrial Robots, IFS, pp. 159–166.

Armada, M., Santos, P. G. de, Jiménez, M. A., Prieto, M. *2003. Application of CLAWAR Machines*. The International Journal of Robotics Research. Vol. 22, nº 3–4, pp. 251-264.

Armada, M., Prieto, M., Akinfiev, T., Fernández, R., González, P., Garcia, E., Montes, H., Nabulsi, S., Ponticelli, R., Sarria, J., Estremera, J., Ros, S., Grieco, J., Fernandez, G. 2005. *On the Design and Development of Climbing and Walking Robots for the Maritime Industries*. Journal of Maritime Research, Vol. II. nº 1, pp. 9-32.

Galvez, J. A., Santos, P. G. de, Pfeiffer, F. 2001. *Intrinsic Tactile Sensing for the Optimization of Force Distribution in a Pipe Crawling Robot*. IEEE/ASME Transactions on Mechatronics, Vol. 6, nº 1.

Pascoal, A., Oliveira, P., Silvestre, C., Bjerrum, A., Ishloy, A., Pignon, J. P., Ayela, G., Petzelt, C. 1997. *MARIUS: An Autonomous Underwater Vehicle for Coastal Oceanography*. IEEE Robotics & Automation Magazine. pp. 46-57.

Santos, P. G. de, Armada, M., Jiménez, M. A. 1997. *An industrial walking machine for naval construction*. In IEEE Int. Conf. on Robotics and Automation, Albuquerque, NM.

Santos, P. G. de, Armada, M., Jiménez, M. A. 1997. *Walking machines. Initial testbeds, first industrial applications and new research*. Computing and Control Engineering Journal, pp. 233-237.

Santos, P. G. de, Armada, M., Jiménez, M. A. 2000. *Ship building with ROWER*. IEEE Robotics and Automation Magazine 7(4):35–43.

Virk, G. S. 2005. *The CLAWAR Project: Developments in The Oldest Robotics Thematic Network*. IEEE Robotics & Automation Magazine, pp. 14-20.

White, T. S., Hewer, N., Luk, B. L., Hazel, J. 1998. *The Design and Operational Performance of a Climbing Robot Used for Weld Inspection in Hazardous Environments*. Proceedings of IEEE International Conference on Control Applications, Trieste, Italy.

KINEMATICS AND DYNAMICS ANALYSIS FOR A HOLONOMIC WHEELED MOBILE ROBOT

Ahmed El-Shenawy, Achim Wagner and Essam Badreddin
Computer Engineering, University of Mannheim , B6. 23-27 part B, Mannheim, Germany
ashenawy@rumms.uni-mannheim.de

Keywords: Modeling, Robot Kinematics, Robot Dynamics, Analysis.

Abstract: This paper presents the kinematics and the dynamics analysis of the holonomic mobile robot C3P. The robot has three caster wheels with modular wheel angular velocities actuation. The forward dynamics model which is used during the simulation process is discussed along with the robot inverse dynamics solution. The inverse dynamics solution is used to overcome the singularity problem, which is observed from the kinematic modeling. Since both models are different in principle they are analyzed using simulation examples to show the effect of the actuated and non actuated wheel velocities on the robot response. An experiment is used to illustrate the performance of the inverse dynamic solution practically.

1 INTRODUCTION

Wheeled mobile robots became an important tool in our daily life. They are found in a multitude of application such as guiding disabled people in museums (Burgard et al., 2002; Steinbauer and Wotawa, 2004) and hospitals (Kartoun et al., 2006), transporting goods in warehouses, manipulation of army explosives (Bruemmer et al., 1998), or securing important facilities (Chakravarty et al., 2004).

Wheeled mobile robots are categorized into two main types: *holonomic* and *non-holonomic*, which are the mobility constraints of the mobile robot platform (Yun and Sarkar, 1998). A holonomic configuration implies that the number of robot velocities DOF (**D**egrees **O**f **F**reedom) is equal to the number of position coordinates. The main advantage of holonomic mobility is the ability of efficient maneuvering in narrow places. In the last two decades, many considerable research efforts addressing the mobility of holonomic wheeled mobile robots were done (Fulmer, 2003; Moore and Flann, 2000) and (Yamashita et al., 2001).

Usually, kinematic modeling is used in the field of WMRs (**W**heeled **M**obile **R**obot) to obtain stable motion control laws for trajectory following or goal reaching (Khatib et al., 1997; Ramírez and Zeghloul, 2000). Using the dynamics modeling during the simulation process results in a better control design. By comparing its results to the practical implementation, a better precision and more variables assumption are achieved in comparison to the kinematics model.

The dynamic modeling is much more complex than the kinematic modeling. Furthermore, the kinematic modeling is required for deriving the dynamic model. Hence, it is assumed that the velocity and acceleration solutions can be solved without any difficulty. Generally, the main property of the dynamic model is that it involves the forces that act on the multibody system and its inertial parameters such as : mass, inertia, with respect to the center of gravity (Albagul and Wahyudi, 2004; Asensio and Montano, 2002).

The holonomic mobile robot "C3P" (**C**aster **3** wheeled **P**latform) is described geometrically in section 2. The singularity problem found in the C3P configuration is illustrated in section 3 through the kinematic analysis. The forward dynamic model and the inverse dynamic solution are presented and analyzed in section 4 to show their different structure. Few simulation examples are shown to illustrate the effect of the actuated and non-actuated variables on the robot behavior using the inverse dynamic solution. In section 5, a lab experiment is presented to show the performance of the solution on the practical implemented module.

2 THE PLATFORM CONFIGURATION

The C3P WMR is a holonomic mobile robot, which is previously discussed in (Peng and Badreddin, 2000). The C3P has three caster wheels attached to a triangular shaped platform with smooth rounded edges as shown in Fig. (1). Each caster wheel is attached to each hip of the platform. The platform origin coordinates are located at its geometric center, and the wheels are located with distance h from the origin and $\alpha_1 = 30^o$, $\alpha_2 = 150^o$, and $\alpha_3 = 270^o$ shifting angles,

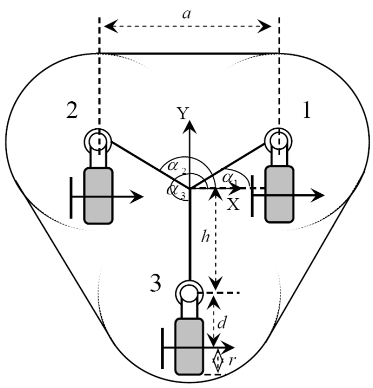


Figure 1: C3P platform configuration.

where

X, Y, ϕ : WMR translation and rotation displacement.

θ_{s_i} : the steering angle for wheel i.

r, d : the wheel radius and the caster wheel offset.

The conventional Caster wheel has three DOF due to the wheel angular velocity $\dot{\theta}_{x_i}$, the contact point angular velocity $\dot{\theta}_{c_i}$ and the steering angular velocity $\dot{\theta}_{s_i}$. The unique thing about the C3P is that it has wheels angular velocities actuation only ($\dot{\theta}_{x_i}$) with no steering angular actuation, which makes the model modular and more challenging.

3 KINEMATICS MODELING AND ANALYSIS

For WMRs' mobility analysis and control, the kinematics modeling is needed. Furthermore, calculating the velocity and acceleration variables are important in the dynamics modeling procedures. In order to analyze and derive the C3P mathematical models, some variables are assigned, such as the following: the robot position vector $p = [X\ Y\ \phi\ ]^T$, the wheel angles vector $q_x = [\theta_{x_1}\ \theta_{x_2}\ \theta_{x_3}]^T$, the steering angles vector $q_s = [\theta_{s_1}\ \theta_{s_2}\ \theta_{s_3}]^T$, and the contact angles vector $q_c = [\theta_{c_1}\ \theta_{c_2}\ \theta_{c_3}]^T$. By differentiating the robot and wheel vectors with respect to time, the robot and wheel velocities are

$$\dot{p} = \frac{dp}{dt},\ \dot{q}_x = \frac{dq_x}{dt},\ \dot{q}_s = \frac{dq_s}{dt},\ \dot{q}_c = \frac{dq_c}{dt}. \quad (1)$$

From the generalized inverse kinematic solution described in (Muir, 1987), the wheel angular velocity inverse kinematic solution is

$$\dot{q}_x = J_{in_x}\dot{p}$$

$$J_{in_x} = \frac{1}{r}\begin{bmatrix} -S(\theta_{s_1}) & C(\theta_{s_1}) & h\,C(\alpha_1-\theta_{s_1}) \\ -S(\theta_{s_2}) & C(\theta_{s_2}) & h\,C(\alpha_2-\theta_{s_2}) \\ -S(\theta_{s_3}) & C(\theta_{s_3}) & h\,C(\alpha_3-\theta_{s_3}) \end{bmatrix} \quad (2)$$

while the steering angular actuation is

$$\dot{q}_s = J_{in_s}\dot{p}$$

$$J_{in_s} = \frac{-1}{d}\begin{bmatrix} C(\theta_{s_1}) & S(\theta_{s_1}) & -h\,S(\alpha_1-\theta_{s_1})+d \\ C(\theta_{s_2}) & S(\theta_{s_2}) & -h\,S(\alpha_2-\theta_{s_2})+d \\ C(\theta_{s_3}) & S(\theta_{s_3}) & -h\,S(\alpha_3-\theta_{s_3})+d \end{bmatrix} \quad (3)$$

and the contact angular velocity inverse solution is

$$\dot{q}_c = J_{in_c}\dot{p}$$

$$J_{in_c} = \frac{-1}{d}\begin{bmatrix} -S(\theta_{s_1}) & C(\theta_{s_1}) & -h\,C(\alpha_1-\theta_{s_1}) \\ -S(\theta_{s_2}) & C(\theta_{s_2}) & -h\,C(\alpha_2-\theta_{s_2}) \\ -S(\theta_{s_3}) & C(\theta_{s_3}) & -h\,C(\alpha_3-\theta_{s_3}) \end{bmatrix} \quad (4)$$

where "C" stands for "cos" and "S" stands for "sin". The solution (2) shows singularities for some steering angles configurations. The singularity appears only when the steering angles are equal. For example, when the steering angles are -90^o, the robot velocity $\dot{Y}$ is not actuated (Fig. (2-a)), and when they are 0^o the velocity $\dot{X}$ is not actuated (Fig. (2-c)). The steering configuration in Fig. (2-b) gives singular determent for the matrix J_{in_x} with -45^o steering angles although all the robot DOFs are actuated.

Obviously, the direction of $[-1\ 1\ 0]^T$ is not actuated, which concludes the following; if all steering angles yield the same value, then the robot is not actuated in the direction parallel to the wheel axes. Fig. (2-d) represents a non-singular steering wheels configuration condition.

Solutions (3) and (4) can be used to overcome the singularity practically by adding practical actuation

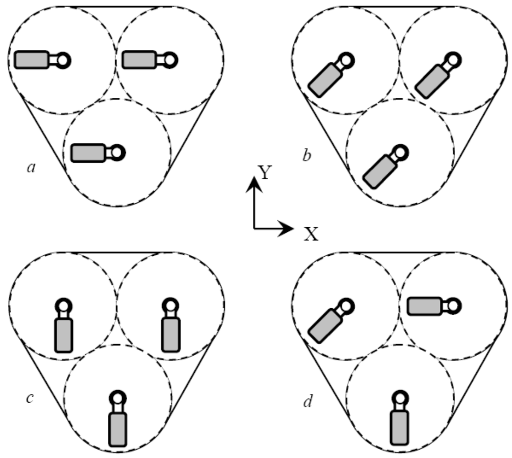


Figure 2: Different steering angles configurations.

to the steering angular velocities, or theoretically by virtually actuating the steering angular velocities (El-Shenawy et al., 2006c).

The forward sensed kinematics used in (El-Shenawy et al., 2006a) shows that the sensed variables are sufficient for robust sensing and slippage detection through the following equation

$$\dot{p} = J_{f_x}\dot{q}_x + J_{f_s}\dot{q}_s, \tag{5}$$

where J_{f_x} and J_{f_s} are the sensed forward solutions for the wheel angular and steering angular velocities respectively. The derivative of equation (5), yields the robot accelerations,

$$\ddot{p} = J_{f_x}\ddot{q}_x + J_{f_s}\ddot{q}_s + g(q_s, \dot{q}_x, \dot{q}_s), \tag{6}$$

from equations (2), (3) and the inversion method proposed in (Muir, 1987) the inverse actuated kinematic accelerations is

$$\begin{bmatrix} \ddot{q}_x \\ \ddot{q}_s \end{bmatrix} = \begin{bmatrix} J_{in_x} \\ J_{in_s} \end{bmatrix} \ddot{p} - g_{cs}(q_s, \dot{p}) \tag{7}$$

4 DYNAMIC MODELING AND ANALYSIS

4.1 Euler Lagrange

The dynamic equations of motion are derived using the Euler-Lagrangian method (Naudet and Lefeber, 2005) based on the Lagrangian function

$$L = K - P, \tag{8}$$

where K denotes the kinetic energy and P denotes the potential energy. Since the C3P is assumed to drive on a planar surface, P is zero.

The Lagrangian dynamic formulation is described as

$$\tau = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q}, \tag{9}$$

where τ is the vector of actuated torques.

The C3P is considered as a closed chain multi-body system. To derive the dynamic model of the C3P, the system is converted into an open chain structure. First, the dynamic model of each serial chain is evaluated using the Lagrangian dynamic formulation (9). Second, the platform constraints incorporate the open chain dynamics into a closed chain dynamics. The robot consists of 7 parts; 3 identical wheels, 3 identical links and one platform (Fig.(3)).

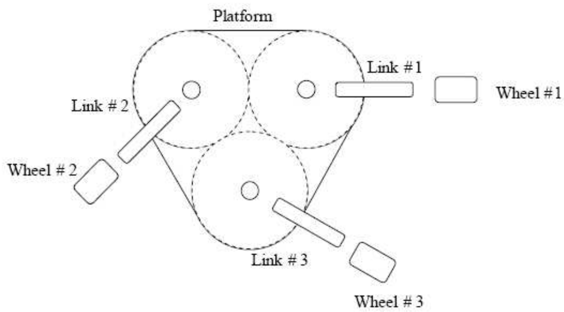


Figure 3: The C3P parts structure.

The kinetic energy of the rigid body depends on its mass, inertia , linear and angular velocities as described by the following equation

$$K = \frac{1}{2}\, m\, V^T\, V + \frac{1}{2}\, \Omega^T\, I\, \Omega \tag{10}$$

where,

m, I : mass and inertia of the rigid body.

V, Ω : linear and angular velocity at the center of gravity of the rigid body.

The sum of the platform kinetic energies equations results in the following Lagrangian function

$$L = \sum_{i=1}^{3} K_{wi} + \sum_{i=1}^{3} K_{li} + K_{pl}, \tag{11}$$

where K_{wi}, K_{li} are the wheel and link i kinetic energies respectively, while K_{pl} is the platform kinetic energy. The wheel co-ordinates q can be considered as the actuated displacements and $\dot{q}$ as the actuated velocities, while τ is the external torque/force vector. The overall dynamics of the robot can be formulated as a system of ordinary differential equations whose solutions are required to satisfy the WMR constraints through the following force/torques vector equation

$$\tau = M(q)\ddot{q} + G(q,\dot{q}) \tag{12}$$

where, $M(q)$ is the inertia matrix, $G(q,\dot{q})$ contains the centripetal and Coriolis terms.

4.2 Control Structure Design

The C3P control structure contains three main models (Fig.(4)): the forward kinematics for calculating the C3P velocities, the C3P forward dynamic model which models the C3P practical prototype on the simulation level, and the inverse dynamics solution.

The velocity controller (V.C) calculates a control signal u from the velocity error $\dot{e} = \dot{p}_r - \dot{p}_o$, which is added to the reference acceleration signal $\ddot{p}_r$. The reference robot velocities $\dot{p}_r$ and accelerations $\ddot{p}_r$ are used in the inverse dynamic solution to deliver the actuated wheels torques τ_{xa}.

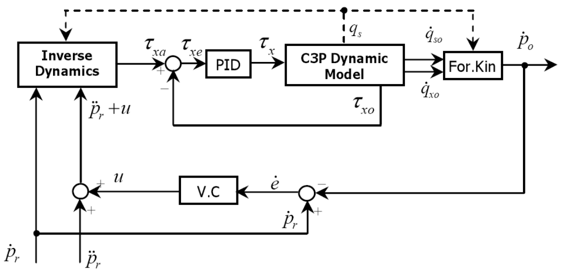


Figure 4: Robot closed-loop structure.

The forward dynamics consists of two main equations; the Wheels Torques Dynamics (WTD) and the Dynamic Steering Estimator (DSE), which were proposed in (El-Shenawy et al., 2006b). The wheel angular velocities are calculated using the wheels torques dynamic equation of motion

$$\tau_x = M(q_s)\ddot{q}_x + G_x(q_s,\dot{q}_x), \tag{13}$$

with respect to the actuated wheels torques, where $\tau_x = [\tau_{x_1}\ \tau_{x_2}\ \tau_{x_3}]^T$. The steering angles and the steering angular wheel velocities are recursively calculated by the steering dynamic estimator

$$\ddot{q}_s = M_{ss}^{-1} M_{xx}\ddot{q}_x + M_{ss}^{-1} G_{ssx}(q_s,\dot{q}_s,\dot{q}_x) \tag{14}$$

corresponding to the angular wheels velocities and accelerations generated due to the applied wheel torque resulting from equation (13).

The C3P dynamic model shown in Fig. (5) has the actuated wheel torques τ_x as an input, while the outputs are the sensed wheel velocities $\dot{q}_x$, the steering angular velocities $\dot{q}_s$, and the steering angles q_s. Since the steering angular velocities are actuated by the angular wheel velocities, the angular wheel velocities $\dot{q}_x$ and accelerations $\ddot{q}_x$ are the main inputs of the steering dynamic estimator. The steering angles q_s and steering angular velocities $\dot{q}_s$ are delayed by unity time interval because the steering dynamic model is calculated recursively according to (14).

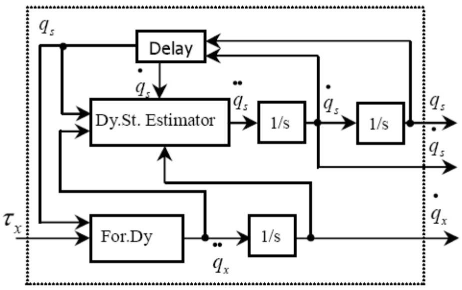


Figure 5: The C3P Dynamic Model.

The inverse dynamic solution proposed is implemented in the velocity control loop to overcome the singularity with simpler velocity controller and better performance. The inverse dynamic equation depends on the platform constraints, which are described in the forward kinematic solution. They are combined using Lagrangian formulation and the dynamic torque equation (9) to obtain the described wheel torques equation

$$\tau_{x_a} = \begin{bmatrix} M_{x_a} & M_{s_a} \end{bmatrix} \begin{bmatrix} \ddot{q}_x \\ \ddot{q}_s \end{bmatrix} + G_{sx_a}(\dot{q}_x,\dot{q}_s,q_s) \tag{15}$$

The matrix M_{x_a} is the inverse dynamic solution for actuating the wheels torques τ_{x_a}, while the matrix M_{s_a} is the inverse dynamic solution for actuating the steering angular acceleration $\ddot{q}_s$ using the wheel torques τ_{x_a}. The inverse dynamics solution is a relation between the desired robot velocities and accelerations ($\dot{q}$, $\ddot{q}$) as an input and the actuated applied torques of the wheels (τ_{x_a}) as an output. However, the dynamic torque equation (15) is a function of $\ddot{q}_x$, $\ddot{q}_s$, $\dot{q}_x$, and $\dot{q}_s$. By using the velocity and acceleration inverse kinematic solutions(2, 3, 4 and 7), the desired torque equation is achieved and the actuation characteristics of the steering angular velocities and accelerations are included in the inverse dynamic solution as well. As a result the actuated torques equation will have the robot velocities $\dot{p}$ and accelerations $\ddot{p}$ as input variables

$$\tau_{x_a} = M_x(q_s)\ddot{p} + G_{xi}(q_s,\dot{p}) \tag{16}$$

$$M_x = \left[\begin{array}{cc} M_{x_a} & M_{s_a} \end{array} \right] \left[\begin{array}{c} J_{in_x} \\ J_{in_s} \end{array} \right], \quad (17)$$

$$G_{xi} = G_{sx_a}(q_s, \dot{p}) - \left[\begin{array}{cc} M_{x_a} & M_{s_a} \end{array} \right] g_{cs}(q_s, \dot{p}) \quad (18)$$

4.3 Model Analysis

The proposed inverse and forward dynamic solutions are different in structure and mathematical representation as well. However, both models should yield the inversion of each other. Therefore some simulation examples are done and analyzed in this section to illustrate the performance of the model. The simulations are done using the structure shown in Fig. (4) with zero values for the velocity (V.C.) and the axes level control parameters to disable their effects. The C3P parameters are set to be exactly like the practical prototype, which are described in Table 1.

Table 1: The C3P parameters.

C3P Parameters	Value	Units
h	0.343	m
d	0.04	m
r	0.04	m
M_p (Platform mass)	25	Kg
I_p (Platform inertia)	3.51	$Kg\ m^2$

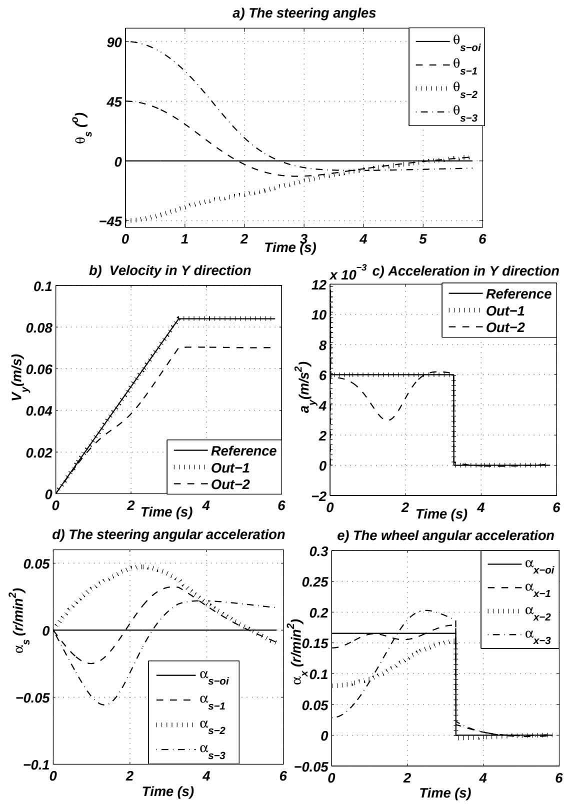


Figure 6: Simulation comparison for ramp input.

Fig.(6) shows a comparison between two different examples. The first example is a non singular condition, where the steering angles are $\theta_{s_1} = \theta_{s_2} = \theta_{s_3} = 0^o$ as shown in Fig. (2c). The input signal is a ramp input in Y direction while the X translational and Φ velocities are zeros. The input V_y or $\dot{Y}$ is a ramp signal till 3.2 seconds then it is constant. The second example is a non singular condition but with different steering angles, where $\theta_{s_1} = 45^o$,$\theta_{s_2} = -45^o$ and $\theta_{s_3} = 90^o$ with the same ramp input signal. Fig. (6-a) shows the trajectory of the steering angles, which are indicated as (θ_{s-oi}) for example one and (θ_{s-i}) for example two.

The reference input velocities maintain zero steering angles value. For the first example, the steering angles keep their initial value, while in the second example the steering angles were adjusted from their initial value to the zero value. The steering wheel adjustment took place due to the the step acceleration input in Y direction. The robot output velocity and acceleration $Out-1$ result from the first example (Fig. (6-b) &(6-c)), which follow the input signal as well as the steering and wheel angular accelerations (α_{s-oi},α_{x-oi}) as shown in Fig. (6-d) &(6-e).

For the second example, the input acceleration and the initial steering angles produce disturbances and oscillations in the wheel angular acceleration (($\alpha_{x_i} = 0$ for i $\in \{1,2,3\}$ (Fig. (6-e)))). Such disturbances produce oscillations in the steering angular accelerations (($\alpha_{s_i} = 0$ for i $\in \{1,2,3\}$ (Fig. (6-d)))), which results negative overshoot in the robot Y acceleration ($Out-2$) (Fig. (6-c))). In addition to the presence of the dynamic delay in the forward dynamics solution (Fig. (5)) and some simulation numerical errors the overshoot appears.

When the robot velocity takes constant value the desired wheel acceleration suddenly change from value 0.17 (r/min^2) to 0 (r/min^2). Such input does not cause multiple oscillations or high overshoots in output signal (Fig. (6-d)).

The next simulation shows the responses of the robot velocities and accelerations after enabling the axes and robot level controllers. The robot starts from the same initial steering angles but the input velocity is ramp signal in X direction and Zero value in the Y and the rotational velocities (Fig.(7-d)). Such input yields the steering angles to reach -90^o (Fig.(7-a)), which is adjusted due to the wheel angular accelerations oscillations resulted in Fig. (7-b).

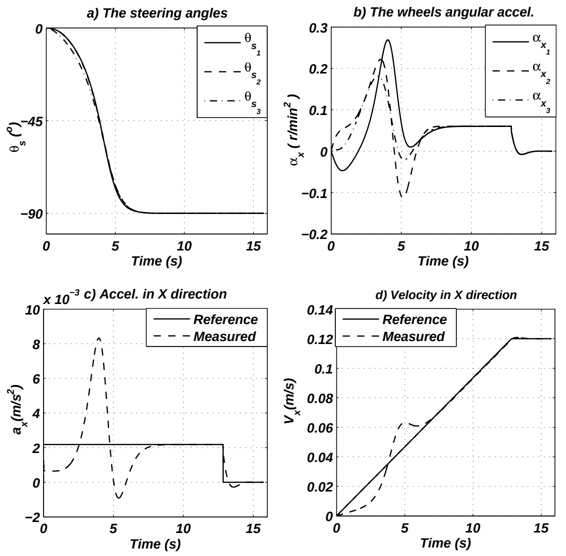


Figure 7: simulation responses for ramp input with singularity configuration.

In the first seven seconds during the steering angles adjustment, the output acceleration of the robot X direction oscillates but it reach the desired value as the steering angles settle with their desired values (Fig.(7-c)). These oscillations affect the robot velocity output as well by oscillating around the desired signal (Fig.(7-d)). In case of having Y as adesired direction, the output signal will be exactly the same like the input, because the effect of the delay unit in the forward dynamics solution is negligible.

5 LAB EXPERIMENT

After analyzing the inverse and forward dynamic solutions, the inverse dynamic solution is implemented on the C3P prototype (Fig. (8)) to test its performance practically. Therefore, an experiment was implemented in the lab with the following initial steering angles $\theta_{s_1} = \theta_{s_2} = \theta_{s_3} = 135^o$ and input robot velocities $\dot{p}_r = [0.5(m/s)\ \ 0.5(m/s)\ \ 0(r/min)]^T$. Such input velocities yields the steering angles to flip 180^o to reach -45^o or 315^o values (Fig.(9-a)).

The steering angles θ_{s_2} and θ_{s_3} flipped after 2 seconds in different directions (Fig.9c), causing oscillations in the robot velocities (Fig.(9-d), (9-e) & (9-f)). After the fourth second, the first steering angle θ_{s_1} flipped producing another overshoots in the robot velocities. The robot velocities (Fig.(9-d), (9-e) & (9-f))

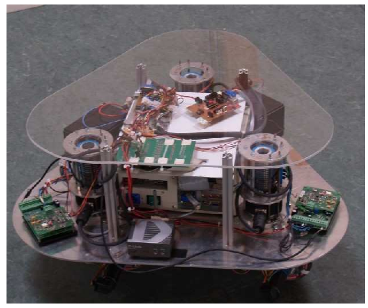


Figure 8: The C3P practical prototype.

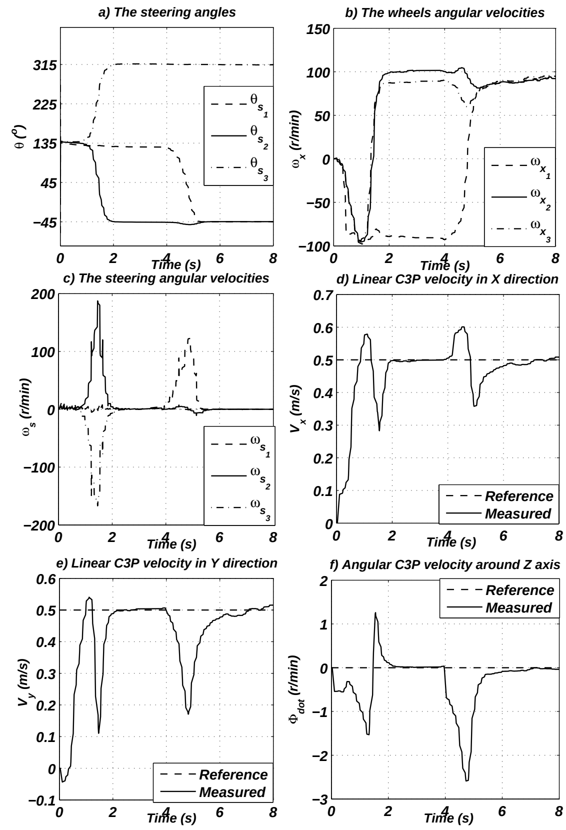


Figure 9: Practical Lab results for the C3P prototype.

are measured with respect to the floor frame of coordinates, which illustrates how the responses follow the reference value in the steady state.

The main advantage and main problem of the C3P configuration are the same. This is the non direct actuation of the steering angular velocities of the wheel. Such a problem is very challenging from the theoretical and practical point of view. Since the steering angular velocities are virtually actuated, their behavior can not be exactly predicted. However, the forward dynamic model is the main factor in deriving the inverse dynamic solution and designing its control structure along with tuning its parameters. These information are very useful during the practical implementation takes place, as in the last experiment.

6 CONCLUSION

The kinematics and the dynamics models of the holonomic mobile robot C3P are presented in this paper. The singularity problem found in such actuation configuration is described by the inverse kinematic solution. The dynamic analysis showed the effect of the wheel and steering angular accelerations of the caster wheels on the inverse dynamic behavior with and without velocity controllers. Although the inverse and forward dynamic models are different in structure, they yield the inversion of each other. The steering angular acceleration plays a very important rule in calculating the reference actuated signal to the platform to overcome the robot singularities and to adjust its steering angles. The simulation examples showed the model dynamic analysis through the robot acceleration variables, and the practical experiment proved the effectiveness of the control structure.

REFERENCES

Albagul, A. and Wahyudi (2004). Dynamic modelling and adaptive traction control for mobile robots. *Int. Jor. of Advanced Robotic Systems, Vol.1,No.3*.

Asensio, J. R. and Montano, L. (2002). A kinematic and dynamic model-based motion controller for mobile robots. *15th Triennial World Congress*.

Bruemmer, J., Marble, J., and Dudenhoffer, D. (1998). Intelligent robots for use in hazardous dor environments.

Burgard, W., Trahanias, P., Haehnel, D., Moors, M., Schulz, D., Baltzakis, H., and Argyros, A. (2002). Tourbot and webfair: Web-operated mobile robots for telepresence in populated exhibitions. *IEEE/RSJ Conf. IROS'02, Full day workshop in "Robots in Exhibitions*.

Chakravarty, P., Rawlinson, D., and Jarvis, R. (2004). Distributed visual servoing of a mobile robot for surveillance applications. *Australasian Conference on Robotics and Automation (ACRA)*.

El-Shenawy, A., Wagner, A., and Badreddin, E. (2006a). Controlling a holonomic mobile robot with kinematics singularities. *The 6th World Congress on Intelligent Control and Automation*.

El-Shenawy, A., Wagner, A., and Badreddin, E. (2006b). Dynamic model of a holonomic mobile robot with actuated caster wheels. *The 9th International Conference on Control, Automation, Robotics and Vision, ICARCV*.

El-Shenawy, A., Wagner, A., and Badreddin, E. (2006c). Solving the singularity problem for a holonomic mobile. *4th IFAC-Symposium on Mechatronic Systems (MECHATRONICS 2006)*.

Fulmer, C. R. (2003). Design and fabrication of an omnidirectional vehicle platform. *Master thesis University of Florida*.

Kartoun, U., Stern, H., Edan, Y., Feied, C., Handler, J., Smith, M., and Gillam, M. (2006). Vision-based autonomous robot self-docking and recharging. *11th International Symposium on Robotics and Applications (ISORA)*.

Khatib, O., Jaouni, H., Chatila, R., and Laumond, J. P. (1997). Dynamic path modification for car-like nonholonomic mobile robots. *IEEE Intl. Conf. on Robotics and Automation*.

Moore, K. and Flann, N. (2000). Six-wheeled omnidirectional autonomous mobile robot. *IEEE Control Systems Magazine*.

Muir, P. (1987). Modeling and control of wheeled mobile robots. *PhD dissertation Carnegie mellon university*.

Naudet, J. and Lefeber, D. (2005). Recursive algorithm based on canonical momenta for forward dynamics of multibody systems. *Proceedings of IDETC/CIE*.

Peng, Y. and Badreddin, E. (2000). Analyses and simulation of the kinematics of a wheeled mobile robot platform with three castor wheels.

Ramírez, G. and Zeghloul, S. (2000). A new local path planner for nonholonmic mobile robot navigation in cluttered environments. *IEEE Int. Conf. on Robotics and Automation*.

Steinbauer, G. and Wotawa, F. (2004). Mobile robots in exhibitions, games and education. do they really help? *CLAWAR/EURON Workshop on Robots in Entertainment, Leisure and Hobby*.

Yamashita, A., Asama, H., Kaetsu, H., Endo, I., and Arai, T. (2001). Development of omni-directional and step-climbing mobile robot. *Proceedings of the 3rd International Conference on Field and Service Robotics (FSR2001)*.

Yun, X. and Sarkar, N. (1998). Unified formulation of robotic systems with holonomic and nonholonomic constrains. *IEEE Trans. on robotics and automation, Vol.12, pp.640-650*.

IMPROVED OCCUPANCY GRID LEARNING
The ConForM Approach to Occupancy Grid Mapping

Thomas Collins, J. J. Collins and Conor Ryan
Department of Computer Science and Information Systems, University of Limerick, Ireland
{*thomas.collins,j.j.collins,conor.ryan*}*@ul.ie*

Keywords: Robotics, Occupancy Grids, Machine Learning, Artificial Intelligence.

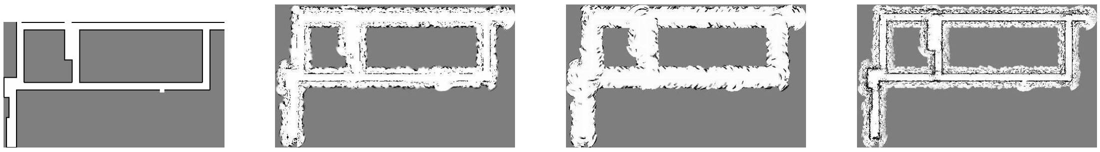
Abstract: A central requirement for the development of robotic systems, that are capable of autonomous operation in non-specific environments, is the ability to create maps of their operating locale. The creation of these maps is a non trivial process as the robot has to interpret the findings of its sensors so as to make deductions regarding the state of its environment. Current approaches fall into two broad categories: on-line and offline. An on-line approach is characterised by its ability to construct a map as the robot traverses its operating environment, however this comes at the cost of representational clarity. An offline approach on the other hand requires all sensory data to be gathered before processing begins but is capable of creating more accurate maps. In this paper we present a new means of constructing occupancy grid maps which addresses this problem.

1 INTRODUCTION

In recent times Occupancy Grids have become the dominant paradigm for environmental modelling in mobile robotics (D. Kortenkamp and Murphy, 1998). An Occupancy Grid is a tessellated 2D grid in which each cell stores fine grained qualitative information regarding which areas of a robots operating environment are occupied and which are empty (Moravec and Elfes, 1985; Elfes, 1989). Specifically, each individual cell in the grid records a certainty factor relating to the confidence that the particular cell is occupied. Such maps are extremely useful for mobile robotic applications as they facilitate tasks such as navigation, path planning, localisation and collision avoidance (Borenstein and Koren, 1991; Dissanayake et al., 2001).
Currently in the Occupancy Grid mapping domain there are two broad approaches: on-line and off-line. The on-line approach is characterised by traditional paradigms such as those from Moravec (Moravec and Elfes, 1985), Matthies (Matthies and Elfes, 1988) and Konolige (Konolige, 1997). The off-line approach has emerged from a more recent paradigm from Thrun (Thrun, 2003). The on-line approach is capable of generating maps in real-time as the robot operates. However these maps often contain inconsistencies such as over estimation of occupied or free space which is undesirable. The off-line approach on the other hand, is capable of generating more consistent maps but cannot do so in real time. These diametric approaches give rise to a mode versus clarity dilemma.
In this paper we introduce and empirically evaluate a novel robotic mapping framework called ConForM (**Con**textual **For**ward **M**odelling) which solves this dilemma through combining the beneficial aspects of both existing approaches. Results from empirical evaluations we have undertaken show that ConForM provides maps that are of better quality than existing paradigms.

2 ON-LINE VS. OFFLINE OPERATION: THE ROBOTIC MAPPING DILEMMA

Two types of model are available for sensory interpretation in robotic mapping. These are the *Inverse* and the *Forward* models (Thrun, 2003). An inverse model attempts to describe an environment by trans-



(a) Ideal Map (b) Inverse model: Moravec and Elfes 1985 (c) Inverse model: Thrun 1993 (d) Forward model: Thrun 2001

Figure 1: Illustrating map generation using inverse/forward sensory models. Overall environmental size: `44m x 35m`. Corridor width: `1.5m`.

lating from effects (sensory measurements) to causes (obstacles). The forward model describes the characteristics from causes to effects. The inverse model is associated with on-line real-time paradigms such as those mentioned previously and the forward with the offline, non real-time approach.

Traditional approaches using inverse sensor models are prone to generating maps that are inconsistent with the operational data from which they were constructed (Thrun, 2003). This is because such techniques decompose the high-dimensional mapping problem into a number of one-dimensional estimation problems, one for each cell in the map. In doing so they do not consider the dependencies that exist between these cells. The forward sensory model addresses this deficit by considering the dependencies that exist between neighbouring grid cells thereby generating more consistent maps.

Figure 1 presents some illustrative maps. Each paradigm used identical sensory data in generating the maps shown. As can be seen the map generated by the forward model is more compatible with the ideal map. This demonstrates the problem currently inherent in the domain which we are addressing. That is, the dilemma of selecting an on-line paradigm that yield maps of lower accuracy versus an off-line paradigm which produces better quality maps.

3 SPECULARITY AND REDUNDANT INFORMATION IN ROBOTIC MappINg

In addition to the type of sensory model used by a mapping paradigm two other issues have a direct correlation on the quality of map produced. These are *Specular Reflection* and *Redundant Information* (Murphy, 2000; Konolige, 1997).

- Specular Reflection: generally occurs when a sonar beam hits a smooth surface and is reflected off the surface at an obtuse angle. This results in either no reading being returned to the sensor or an erroneous reading being returned that has bounced off many surfaces.
- Redundant Information: commonly arises when the robot has been in the same pose for a period of time and hence its sensors report multiple identical readings from that pose.

4 THE CONFORM APPROACH TO ROBOTIC MAPPING

ConForM has two distinct aspects. These are:

1. The explicit modelling of sensory data to deal with the specular and/or redundant information.
2. The use of an on-line forward sensory model to translate the sensory readings into occupancy values for inclusion in the grid map.

4.1 Conform: Dealing With Specular Readings

ConForM's treatment of the problem of specularity is novel as we consider it from two perspectives. The first is labelled *Acceptability/Agreeability* and the second *Trait Verification*. At each time-step *Acceptability/Agreeability* consider solely the set of readings currently received and evaluates each with respect to its neighbouring readings. *Trait verification* on the other hand takes a wider perspective by evaluating readings in relation to the current perceived state of the environment.

4.1.1 Acceptability and Agreeability

Acceptability: Consider a reading s and let us assume that it reports a range reading with a distance of d. As operating environments are formed from regular features and as the perceptual fields of neighbouring sensors generally overlap we can assess the consistency of a particular reading by evaluating its probabilistic profile in relation to its neighbours. A reading whose measurement is corrupted by Gaussian noise of zero mean and variance σ^2 has the following probability distribution where m is the map as illustrated in equation 1. This is based on the standard specification of a sensory model (Elfes, 1989).

$$p(s_t|m) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\frac{(d)^2}{\sigma^2}} \tag{1}$$

Strictly speaking m is the local map corresponding to the current perceptual field and therefore a sub set of the overall map that is produced.

Now consider the readings s^{-1} and s^{+1} the neighbouring readings on either side of the reading s. The probabilistic profile of these readings are used to support or refute the reading s. If reporting an obstacle each will have an associated distance d^{-1} and d^{+1}. Therefore we can calculate the probability distribution for these readings using equation 1. These distributions are compared to determine if the readings are consistent. This is accomplished by translating the reading s to the position of s^{-1}. Upper and lower bound profiles for s are calculated at this position through scaling the original distance to the point of interest d by the amount of translation required and also taking cognisance of the natural error range of the sensor. If the readings are reporting on the same environmental conditions the reading s will be encompassed by the determined bounds. If this is so the reading is deemed as being acceptable and subsequently allowed to progress for further consideration. An identical procedure is utilised when considering the reading s^{+1}. A reading s is discarded only when both acceptability tests indicate that it is unacceptable.

Agreeability: The sister concept of acceptability is *Agreeability*. It considers readings that report free space. It is similar to *Acceptability* in that we evaluate a reading in terms of its neighbours. Robotic sensors are good at accurately reporting free space meaning that we can use a direct comparison method with free space readings as it is the detection of an obstacle or not which is important, not the actual difference in any distance reported. Therefore when determining agreement, for efficiency, we do not construct probabilistic profiles for the readings. Rather we use the ranges reported instead. If one of a readings immediate neighbours is not in agreement with the reading itself we allow the reading s to proceed to the next stage of the process where it will be checked in the context of the generated map, using *Trait Verification*. If neither of s's immediate neighbours report a free-space reading then the reading is discarded.

4.1.2 Trait Verification

Agreeability and acceptability deal with specular readings in a bottom up fashion at the local level. Specifically this is in the context of a single reading set. As outlined above there are cases when the reliability of readings cannot be determined from purely considering the local view of the reading set from which they originated. Therefore we also need to consider the top down, global, perspective which takes into account the environmental features determined to date and recorded in the map being constructed. This is the basis of the *Trait Verification*.

In its operation Trait Verification makes use of the fact that environments contain structural regularities and symmetries such as walls that can be approximated using line segments. This is used as a basis for the construction of two environmental views:

- V: A temporary sonar view which consists of traits, or line segments, that can be estimated from the current set of sensory readings.
- L: A local view which contains a history of the line segments estimated from past sensory readings. Line segments are maintained for an area covering four times the perceptual field of the robot along the path the robot has traversed.

L is used to form a hypothesis as to the *probable* state of the environment from the robots current perspective. This is accomplished by extending L to cover the current location of the robot using the historical perspectives as a reference point.

Following this L and V are reconciled. Firstly, certainty values in the range $0 \rightarrow 1$ are calculated for the readings that give rise to traits in V. This is accomplished through use of standard singular displacement specifications presented in (Elfes, 1989).

Having determined certainty values in the readings, V and L are reconciled. Two courses of action are applicable, depending on whether or not sufficient state was available for L's construction.

If enough state was not present to provide four perceptual lengths centred on the oath traversed by the robot, v_i's attributes are considered. v_i is a trait in V and its attributes relate to the reading(s) that gave rise to the trait. For example the certainty associated with the reading(s) or whether the reading(s) were previously flagged as potentially erroneous. If the reading was flagged as potentially erroneous from the *Acceptability/Agreeability* and *Trait Verification* steps or the reading certainty is below a determined threshold and there is not an equivalent trait in L, where in this case L has a size equivalent to maximum perceptual range available, the reading is discarded.

If sufficient state was available L and V are compared directly. If traits coincide in both views the readings that gave rise to those traits are accepted, provided that they have not been flagged as possibly erroneous. If they have been flagged the attributes of the trait v_i

in V are considered. If two or more sensors agree on the existence of the trait then the flagged reading is accepted. If the trait was detected by a single sensor the certainty value associated with that reading is consulted. If the certainty is below the threshold the reading is rejected. Otherwise it is accepted. If a trait occurs solely in V and not in L then the attributes of the trait are considered. If the flagged status and confidence value of the reading(s) that gave rise to the trait are acceptable, the reading is allowed to proceed for further utilisation. The problem of a reading relating to a trait solely in L and not in V is dealt with in the same manner.

4.2 Conform: Addressing Redundant Information

To deal with the problem of redundant information ConForM makes use of pose buckets (Konolige, 1997). With pose buckets a map has a dual representation where each cell represents both the occupancy of the area and the pose of readings that have affected that cell. Therefore a record is maintained stating whether a reading from a given distance and angle has affected a particular cell. This means that the first reading received from a specific pose will be utilised, and all following readings from that pose for this cell are discarded, as they merely duplicate information already in the model.

4.3 Conform: Sensor Model

As per the original formulation, ConForM's forward model it also based on optimisation using the EM algorithm (Dempster et al., 1977). It is a mixture model, which accounts for the potential causes of a reading (Thrun, 2003). A measurement may correspond to the detection of an obstacle somewhere in the perceptual field of the sensor, failure to detect any obstacle thereby reporting freespace, or indeed, a random fluctuation of a sensor. Each case has an associated probability. The model convolves these potential causes and associated Gaussian noise into an amalgamated probability distribution which is subsequently optimised by the EM algorithm to determine the most likely cause of the received reading.

Our model differs from the original in that operates on-line and in real-time. The on-line and real-time use of the EM algorithm in ConForM is facilitated through a two step approach. The first step consists of explicitly dealing with potentially erroneous or redundant information through *Acceptability/Agreeability*, *Trait Verification* and *Pose Buckets*. As such the readings available for the second stage encompass more accurately the true state of the perceived environment meaning that EM can be applied to a search space that is tractable during real-time operation.

Using the EM algorithm to determine a map

1. *Initialisation*: Unlike traditional occupancy grid mapping algorithms using inverse sensor models EM does not estimate posteriors. Therefore maps resulting from EM are discrete with each cell being either occupied or empty. As such the cells in the map being constructed are initialised to an occupancy of 0.5.
2. *E-step*: The E-Step calculates the expectations for the potential causes of readings conditioned on the map m and the current set of readings S.
3. *M-step*: The M-step assumes all expectations are fixed and calculates the most likely map based on these expectations. The probability distributions calculated in the E-Step encapsulate all potential causes of the readings in S when determining a new map m. Maximisation of these distributions are performed by hill climbing in the space of all maps. The search is terminated when the target function is no longer increasing.
4. *Incorporating Uncertainty*: EM calculates only a single map not an entire posterior. An approximation which conditions the posterior on the map generated by EM is utilised to incorporate uncertainty into the map, thereby providing useful information for real-time operation.
5. Finally we integrate the map generated by EM into the overall map using a Bayesian based integration.

5 EMPIRICAL EVALUATION

Real world and simulated environments were used to empirically evaluate ConForM. The simulator used was the Saphira architecture with the associated Pioneer simulator. For simulated experiments odometry error was turned off so that wheel slippage would not be a factor thus allowing us to focus on evaluating the performance of the mapping paradigms in large cyclic environments such as those illustrated earlier. For real world experimentation we used relatively small office environments purely for the reason that wheel slippage and thus odometric error is minimal over such short distances.

5.1 Benchmarking Technique

To evaluate the maps generated during our experiments we use an extensible suite of benchmarks which allow for the empirical evaluation of map building paradigms (Collins et al., 2004; Collins et al., 2005).

1. *Correlation*: As a generated map is similar to an image it is possible to use a technique from image analysis known as *Baron's cross correlation coefficient* (Baron, 1981) as a basis for evaluating the map.
2. *Map Score*: This is a technique which calculates the difference between a generated map and an ideal map of the environment (Martin and Moravec, 1996).
3. *Map Score of Occupied Cells* This metric is similar to the previous one but only tests those cells in the map that are occupied.
4. *Path Based Analysis*: To fully evaluate a generated map its usefulness to a mobile robot must be considered.
   - The degree to which the paths created in the generated map would cause the robot to collide with an obstacle in the real world, and are therefore invalid. *False Positives*.
   - The degree to which the robot should be able to plan a path from one position to the another using the generated map, but cannot. *False Negatives*.

5.1.1 Determining an Overall Score

To allow an overall score to be determined we have developed an amalgamation technique which can be used to rank the overall performance of mapping paradigms relative to each other as outlined in equation 2.

$$C_{\mathtt{map}\in M} = \frac{D_{\mathtt{map}} + P_{\mathtt{map}}}{2} \quad (2)$$

$$D_{\mathtt{map}} = \frac{(1-\text{MapScore}_{all}) + (1-\text{MapScore}_{occ}) + B_n}{300}$$

$$P_{\mathtt{map}} = 1 - \frac{(\text{FP}) + (\text{FN})}{200}$$

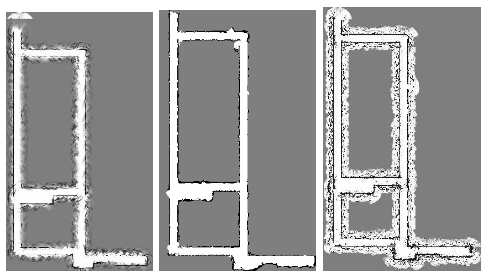
$C_{\mathtt{map}\in M}$ is the overall classification score obtained, M is the set of maps generated in an experiment, $\mathtt{map}$ is a particular map within the set of maps M. MapScore$_{all}$ and MapScore$_{all}$ are the normalised result from the *Map Score* metrics, B_n is the normalised *Correlation* result. FP is the normalised *False Positive* result and FN is the normalised *False Negative* result.

5.2 Results

In determining the performance of ConForM we empirically evaluated it in relation to its peer mapping paradigms, the original Forward Modelling paradigm of Thrun (Thrun, 2003) and an on-line paradigm from Konolige (Konolige, 1997) which has proven to have the best performance of the inverse model based paradigms (Collins et al., 2005).
Benchmarking consisted of completing a number of data acquisition runs in the environments and using this data in conjunction with the mapping paradigms to generate the grid maps. Our experiment used four differing environments, two simulated and two real world, with three data acquisition runs being completed per environment. Therefore the results presented here are derived from evaluating a total of thirty six individual grid maps. Table 5.2 presents the amalgamated score for the mapping paradigms obtained using the benchmarks outlined above. A larger evaluation recently completed and to be reported on, which consisted of ten differing environments and 3600 individual maps, reported trends consistent with those outlined here.

Table 1: Evaluating the ConForM approach to robotic mapping.

Mapping Paradigm	Result
Moravec and Elfes 1985	0.67
Matthies and Elfes 1988	0.65
Konolige 1997	0.76
Thrun 2001	0.84
ConForM	0.87



(a) Konolige 1997 (b) ConForM (c) Thrun 2003

Figure 2: Illustrative maps from the ConForM evaluation.

5.3 Analysis

Overall the results show that ConForM has outperformed the other approaches. ConForM outperforms

the inverse model based approaches because of its improved sensor model and the manner in which it tackles the problem of specularity in addition to its use of pose buckets. In dealing with specularity, the multi-faceted approach consisting of *Acceptability and Agreeability* and *Trait Verification* is capable of a finer reading set analysis when compared to inverse model based approaches. This also has the knock-on effect of making the operation of the pose buckets more accurate as they suffer less from the problem of spurious readings giving rise to false hypothesis regarding the perceived state of the environment.
ConForM outperforms the original Forward Modelling approach because of its pro-active approach to the problems of specularity and redundant information. That original approach addressed the problems of seemingly conflicting information through the EM algorithm. The likelihood of the reading was evaluated in a global context meaning that some localised accuracy may be sacrificed. In ConForM the Forward Model used considers the local perspective meaning that it is capable of capturing and retaining more subtle characteristics that may be dismissed in the offline approach.

6 SUMMARY

Overall ConForM overcomes the problems inherent in traditional approaches such as the need for assumption of cell independence or the need for offline operation. It also overcomes the issue of the existing forward model approach not being applicable in an on-line context. In addition it generates maps that are more consistent then existing approaches. The areas for further consideration and research in relation to ConForM include refining the threshold used with *trait verification*, investigating the use of EM as a basis for refining already generated portions of the map and investigating alternative EM formulations such as Bayesian based approximations.

REFERENCES

Baron, R. J. (1981). Mechanisms of human facial recognition. In *International journal of man machine studies*, volume 15, pages 137–178.

Borenstein, J. and Koren, Y. (1991). The vector field histogram - fast obstacle avoidance for mobile robots. *IEEE Transactions on Robotics and Automation*, 7(3):278–288.

Collins, J., O'Sullivan, S., Mansfield, M., Eaton, M., and Haskett, D. (2004). Developing an extensible benchmarking framework for map building paradigms. In *Proceedings Ninth International Symposium on Artificial Life and Robots*, Oita, Japan.

Collins, T., Collins, J., O'Sullivan, S., and Mansfield, M. (2005). Evaluating techniques for resolving redundant information and specularity in occupancy grids. In *AI 2005: Advances in Artificial Intelligence*, pages 235–244.

D. Kortenkamp, R. B. and Murphy, R. (1998). *AI-based Mobile Robots: Case studies of successful robot systems*. MIT Press, Cambridge, MA.

Dempster, A., Laird, A., and Rubin, D. (1977). Maximum likelihood from incomplete data via the em algorithm. In *Journal of the Royal Statistical Society*, volume 39 of *B*, pages 1–38.

Dissanayake, G., Newman, P., Clark, S., Durrant-Whyte, H., and Csorba, M. (2001). A solution to the simmultaneous localisation and map building (slam) problem. In *IEEE Transactions of Robotics and Automation*, volume 17, pages 229–241. IEEE.

Elfes, A. (1989). *Occupancy Grids: A Probabilistic Framework for Robot Perception and Navigation*. PhD thesis, CMU.

Konolige, K. (1997). Improved occupancy grids for map building. *Autonomous Robots*, (4):351–367.

Martin, M. C. and Moravec, H. (1996). Robot evidence grids. Technical Report CMU-RI-TR-96-06, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.

Matthies, L. and Elfes, A. (1988). Integration of sonar and stereo range data using a grid-based representation. In *Proceedings of the 1988 IEEE International Conference on Robotics and Automation*.

Moravec, H. and Elfes, A. (1985). High resolution maps from wide angle sonar. In *Proceedings of the 1985 IEEE International Conference on Robotics and Automation*.

Murphy, R. R. (2000). *Introduction to AI Robotics*. MIT Press.

Thrun, S. (2003). Learning occupancy grids with forward models. *Autonomous Robots*, 15:111–127.

MODELING ON MOLTEN METAL'S PRESSURE IN AN INNOVATIVE PRESS CASTING PROCESS USING GREENSAND MOLDING AND SWITCHING CONTROL OF PRESS VELOCITY

Ryosuke Tasaki, Yoshiyuki Noda, Kazuhiko Terashima

Department of Production System Engineering, Toyohashi University of Technology

Hibarigaoka 1-1, Tempaku, Toyohashi, 441-8580, Japan

{tasaki, noda, terasima}@syscon.pse.tut.ac.jp

Kunihiro Hashimoto*, Yuji Suzuki**

Sintokogio, Ltd., Honohara 3-1, Toyokawa, 442-8505, Japan

**Aisin Takaoka Co., Ltd., Tennoh 1, Takaokashin-machi, Toyota, 473-8501, Japan*

k-hashimoto@sinto.co.jp, yuji.suzuki@to.at-takaoka.co.jp

Keywords: Press casting, Pressure control, Computational fluid dynamics, Modeling, Casting detect.

Abstract: This paper presents modeling and control of fluid pressure inside a mold in a press casting process using greensand molding as an innovative casting method. The defect-free manufactures of casting product in the press process are very important problem. Then, it is made clear that the press velocity control achieves to reduce the rapid increase of fluid pressure. A mathematical model of the molten metal's pressure in a casting mold is built by using a simplified mold and investigated the availability by comparison with the CFD model. A pattern of the press velocity from the high speed to the lower speed is derived by using the mathematical model. Finally, the effectiveness of the proposed switching velocity control has been demonstrated through CFD computer simulations.

1 INTRODUCTION

Recently, an innovative method called the press casting process using the greensand mold has been actively developed for improving the productivity by authors group. The casting process is shown in Figure 1. In the casting process, the molten metal is poured

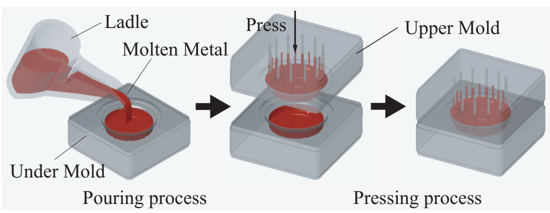


Figure 1: Press casting process.

into the under mold by tilting the ladle. After pouring, the upper mold is fallen down towards the lower mold, and pressed. The process enables us to enhance the production yield rate from 70[%] to over 95[%], since sprue runner and cup are not required in the casting plan(Y.Noda et al., 2006). This process is comprised of two parts such as a pouring and a pressing processes. In the pouring part of the casting process, it is needed to pour the molten metal into the under mold precisely and quickly, and suppress the splash of the molten metal in the mold. In the conventional pouring method, the outflow quantity from the ladle is larger than the volume required in the actual product, and the production yield rate is then decreased. Pouring controls on the pouring process in the press casting were studied by past studies(Y.Matsuo et al., 2006), (Y.Noda et al., 2006).

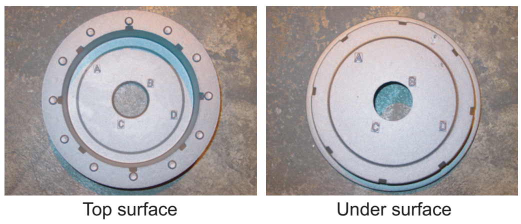
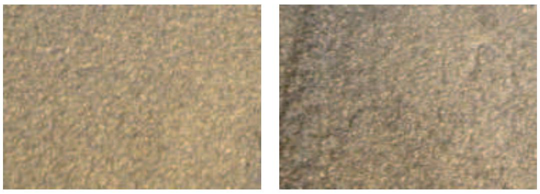


Figure 2: Casting product by an innovative press casting using greensand mold.

On the other hand, in the press part, the casting



Sound case: v=5[mm/s] Defect case: v=122[mm/s]

Figure 3: Inner surface of products.

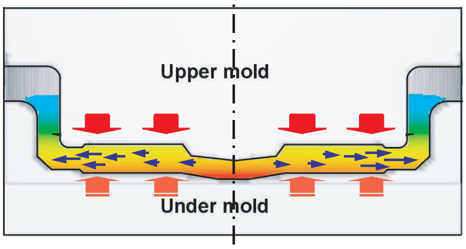


Figure 4: Illustrative figure on fluid behavior of molten metal.

defects are caused by the pattern of press velocity. As an representative example of casting product with a press casting process, a brake drum is shown in Figure 2, where the press velocity is 5[mm/s]. Figure 3 shows the photographs of surface of iron casting produced by Figure 2. In the high speed press such as the velocity of v=122[mm/s], a casting product generates rough surface. This surface defect is called a metal penetration such that the solidified molten metal is soaked among the sand particles in the greensand mold. As seen from Figure 4, this defect is thought to be the generation of high pressure of fluid in the mold due to the rapid velocity of press. Whereas, in the case of slow velocity in the press, a defect of the oxide film in the surface of products and a defect of void due to the rapid solidifications are generated. Therefore, the press velocity control is demanded to adequately suppress the fluid pressure in the high speed press.

The pressure control methods have been proposed in the conventional casting method. To realize the high quality product such as spheroidizing and densification for iron casting, optimal design of sink head is achieved by using the simulation analysis on understanding the explicit solidification property(Louvo et al., 1990). To simulate the filling behavior of molten metal, the rheological characterization has been experimentally studied by H. Devaux(H.Devaux, 1986). 3D-visualization technology was developed by C. Galaup et al.(C.Galaup and H.Luehr, 1986), (I.Ohnaka, 2004). In the injection molding process, the pressure control problem has been successfully achieved by Hu J, 1994. A model on PID gain's selection is proposed for the pressure control in filling process. Then, the effectiveness of a mathematical model with the identified the physical parameters for control performance is verified experimentally(Hu, 1994).

The first keynote on the press casting process using greensand molding has been published by Terashima(K.Terashima, 2006). The press casting process is that the molten metal poured in the under mold is fluidized by the falling down of the upper mold(K.Terashima, 2006). The pressure control by changing the press velocity has not yet been applied, although its importance has been addressed by Terashima in the press casting method. Therefore, we propose to suppress the pressure adequately by controlling the press velocity in the press casting system. The pressure of molten metal in the mold must be detected to control the process adequately. However, measurement of the fluid pressure is difficult, and the use of the contact pressure sensor can not be applied, because the fluid temperature is very high around about 1400. Then, in this paper, the pressure is estimated by using the reaction force measured by a load-cell which is set above the upper mold. A mathematical model of the molten metal pressure in a casting mold is newly given. Based on this mathematical model, an ideal pattern of press velocity is proposed to fall down the upper mold rapidly towards the lower mold with suppressing the fluid pressure.

2 PRESS PROCESS IN PRESS CASTING SYSTEM

The panoramic photograph of the press casting ma-

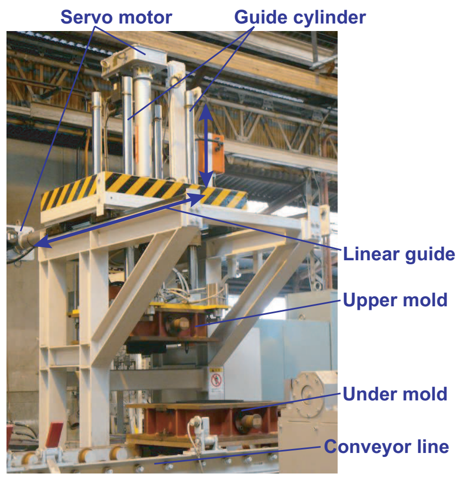


Figure 5: Press casting machine.

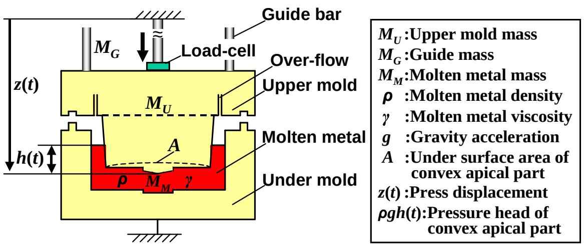


Figure 6: Outline of press process.

chine is shown in Figure 5. Figure 6 shows the illustration diagram of the press casting system. The molten metal is pressed by making the upper mold falling down towards the under mold. The upper mold consists of a greensand mold and a molding box. The upper mold has several passage parts in the convex part, which is called the over-flow as shown in Figure 6. The molten metal over the product volume flows into the over-flow part in the pressing.

The upper mold is moved towards up-and-down by using the press cylinder. The position of the upper mold is continuously measured by an encoder set in the servo cylinder. The position feedback control to obtain the desired behavior for the upper mold, is realized by using the PID controller. Then, the reaction force from molten metal is also measured by the load-cell installed on the servo cylinder.

3 MODELING OF PRESS PROCESS

3.1 Pressure Analysis by CFD

Visualization technology for observing time behavior of filling the fluid has been extensively developed. The pressure of molten metal in the mold during the press process is investigated by using commercial scientific software of CFD (Computational Fluid Dynamics). In this paper, FLOW-3D, a well-known CFD analysis software designed by FLOW SCIENCE Inc., is applied. The filling behavior analysis in press process is available by means of an expressive function of moving obstacle for the fluid.

To investigate the relationship between the load-cell response in experiments and the pressure behavior of molten metal using CFD, simulations using CFD and experiments using Figure 6 were conducted. As an example, simulation and experiment in the conditions of Table 2 were done. Here, the sampling period is 0.01[s], and the mesh block width is 2[mm] in CFD analysis. The relation between the calculated

Table 1: Simulation and experimental condition.

Press velocity; v	30[mm/s]
Pouring fluid mass; M_M	5.37[kg]
Pouring time; T_p	10.1[s]
Pouring fluid temperature; T_M	1405[°C]
Molten metal viscosity; γ	0.00235[Pa·s]

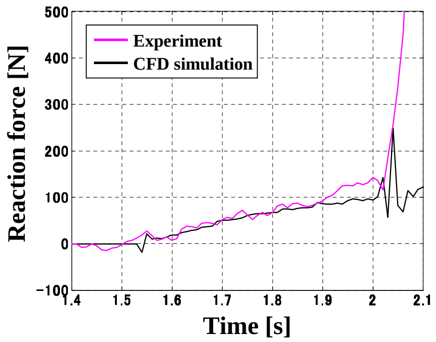


Figure 7: Comparison of reaction force between simulation and experiment.

pressure P_c[Pa] and the reaction force F_u[N] measured by load-cell is expressed by Eq.(1).

$$F_u = AP_c \tag{1}$$

, where A[m^2] in Figure 6 is the under surface area of upper mold.

The comparative result is shown in Figure 7. In Figure 7, gray line is experimental result, and black line is simulation result using CFD. The upper mold touches at the molten metal in time of 1.52[s]. Concerning the time behaviors, the significant increasing reaction force appears at the time of about 2.03[s]. This time is approximately equal to the time when the molten metal flows into the over-flow parts. The high pressure of molten metal in the mold generates at this time. Subsequently, in the end period of press, the feature of the responses is greatly different. This is due to the gravity release by the upper mold sets on the under mold.

From Figure 7, the reaction forces measured in the both of CFD and experiments are thought to be approximately equal up to the 2.03[s]. Then, it was confirmed that the pressure calculated by CFD represented the actual pressure of molten metal in the mold.

3.2 Modeling with Respect to Pressure of Molten Metal

The pressure results by CFD analysis in the filling process well explained experimental results with high

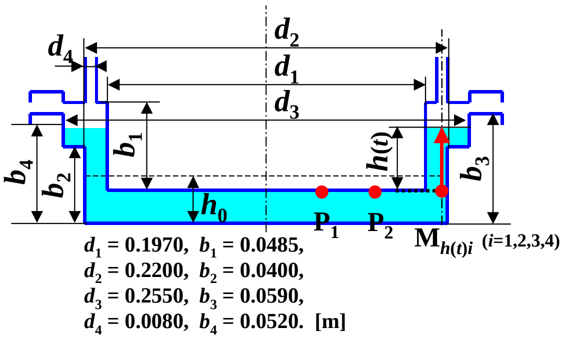


Figure 8: Shape of simplified casting mold shape.

reliability. However, press behavior cannot be calculated in real-time by the CFD analysis. The online estimation of pressure in the mold is required in the press casting system. The CFD is very effective to analyze the fluid behavior in off-line, and hence it is useful to predict the behavior and also optimize a casting plan. However, it is not enough for control design in real-time, because of calculation time. Therefore, we need to build a brief model for control design by using CFD simulation and experiment.

The estimation of the pouring volume is available by using the position data of the upper mold and estimating the contact time between under surface of the upper mold and the molten metal. They are measured respectively using the encoder and the load-cell. Then, to suitably realize the press velocity control without the excessive pressure, the estimation of pressure behavior is done by using the estimated data of pouring volume. From this reason, we build a mathematical model of molten metal pressure for the press velocity.

The mold shape used in authors study has a large convex parts with cross section of A as shown in Figure 6. To examine the pressure behavior for the genesis part of defect, a simplified mold shape plumbed the parts of curve, slope and draft angle for the primary mold shape. The simplified mold shape is shown in Figure 8, where b and d mean the height and the diameter respectively. $P_j(j$=1,2) are genesis parts of defect. The pressure fluctuation in press is represented by using a pressure model for the ideal fluid such that the incompressible and nonviscous fluid is assumed. Here, $h(t)$ in Figure 8 means the fluid level from under surface of upper mold. The head pressure P_j is directly derived from $h(t)$. The press distance $z(t)$ of upper mold is a downward distance from the position at the contact time of the poured fluid and the upper mold. As the press velocity increases, the dynamical pressure is varied by the effect of the liquidity pressure. Then, the hydrodynamic pressure for peak fluid height area is involved in P_j. Therefore, pressure P_b in P_j is consisted of head and hydrodynamic pressure, and is represented by Eq.(2).

$$P_b(t) = \rho g h(t) + \frac{\rho}{2}\dot{h}(t)^2 \tag{2}$$

The flow passage areas have three situations, *case* 1: $\pi(d_2 - d_1)^2/4$, *case* 2: $\pi(d_3 - d_1)^2/4$ and *case* 3: $n\pi d_4^2/4$, where the number n of the over-flow as diameter d_4 is equal to twelve. Figure 8 represents *case* 2. The following equations represent the fluid level variation in the each situation, and they are simply derived by assuming the incompressible fluid.

$$h(t) = \begin{cases} case\ 1: \ h(t) < h_{sw1}, \\ \frac{d_2^2}{d_2^2 - d_1^2} z(t) \\ case\ 2: \ h_{sw1} \le h(t) < b_1, \\ \frac{1}{d_3^2 - d_1^2}(d_3^2 z(t) + d_1^2 h_{sw1}) \\ case\ 3: \ b_1 \le h(t), \\ \frac{1}{nd_4^2}\{d_3^2 z(t) + (nd_4^2 - d_3^2) b_1\} \end{cases} \tag{3}$$

, where h_{sw1} and b_1 represent the threshold fluid level of $h(t)$ on *case* 1*case* 2, *case* 2*case* 3 respectively. h_{sw1} is expressed as follows. And,

$$h_{sw1} = \frac{d_2^2}{d_1^2}(b_2 - h_0) \tag{4}$$

, where h_0 means the initial fluid height before the upper mold touches to the molten metal. When the fluid height $h(t)$ equals to h_{sw1}, the equation of $h(t)$ changes from *case* 1 to *case* 2. Then, when $h(t)$ reaches to the height of b_1, $h(t)$ of Eq.(3) is changed from *case* 2 to *case* 3. As described the above, the pressure response for press velocity is determined from the both of initial fluid height and mold shape.

Eq.(2) or the mathematical pressure model of the molten metal in a mold is validated from the fluid behavior analysis by FLOW-3D on the filling in a press. Comparison of ideal fluid height $h(t)$ in a simplified mold and $h(t)$ in CFD simulation, is shown in Figure 9. As the CFD analysis results, height behavior of $M_{h(t)_i}$(: the measurement points of the over-flow) in Figure 8. The fluid height $h(t)$ for the parts of overflow is obtained. The press velocity is set as 5[mm/s].

When the ideal(incompressible and nonviscous) fluid height becomes steady-state response, the height in CFD results show the lower value of $h(z)$ due to the compression of the fluid by a gravity force. Next, pressure in the generation area of metal penetration defect is compared with a simplified mold.

The pressure behavior of P_2 in Figure 8 as the CFD result is shown, because the pressure response of P_2 is approximately equal to response of P_1, and area of P_2 is generation point of metal penetration defect. As comparing the results between CFD and a simply mathematical model, the pressure responses in press velocities of 5∼30[mm/s](5[mm/s] steps) are shown in Figure 10. The pressure performances of ideal fluid in the mathematical model are in excellent agreement with CFD analysis. Therefore, the pressure expressed by Eq.(2) is thought to be validated for the pressure of molten metal.

4 PRESSURE CONTROL

In this section, the simulation for suppression of rapid increase of pressure is executed using CFD analysis. It is already confirmed that the defect of metal penetration in a press process appears around over 80[mm/s]. In the case of over 80[mm/s], the defect is caused by rapid increase of pressure, when the molten metal flows arrives at the over-flow. Then, the switching action of press velocity at the time of the over-flow is started. The switching time is derived by using Eq.(3) and Eq.(4) of a simplified mold. In this pressure suppression simulation, the initial press velocity sets at 100[mm/s], and switches to the velocity of 10[mm/s] at the switching time of 1.29[s], where the initial fluid height of pouring outflow sets at 0.0192[mm]. Then, the calculated switching time is 0.14[s]. This switching time means the elapsed time, since the molten metal contacts with the under surface of the upper mold. The temperature of the molten metal in the mold is assumed to be about 1300[s].

The simulation results for pressure suppression in press process is shown in Figure 11. As seen from

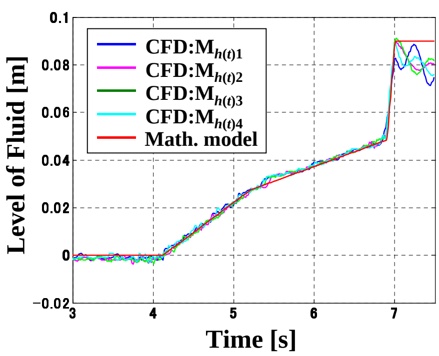
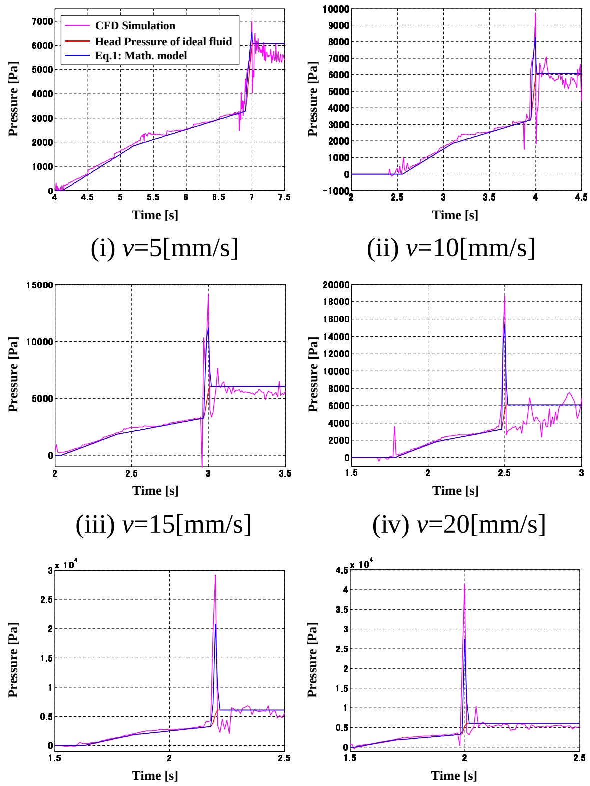


Figure 9: Fluid level in the case of v=5[mm/s].



(i) v=5[mm/s] (ii) v=10[mm/s]
(iii) v=15[mm/s] (iv) v=20[mm/s]
(v) v=25[mm/s] (vi) v=30[mm/s]

Figure 10: Simulation Results of pressure behavior.

Figure 11, the fluctuation of pressure behavior using the switching control of velocity is dramatically smaller than that of non-switching case. In the time of 1.17[s], rapid excessive rise pressure is caused by the contact of molten metal with upper mold.

In the case of the constant velocity of 100[mm/s], the pressure peak value of 304080[Pa] is observed at the time of the over-flow. As seen from Figure 11, using the velocity switch from high-speed to low-speed at the specified time, rapid increase of pressure was drastically reduced. Then, press process time on this switch velocity is approximately equal to the time in press velocity of 80[mm/s] (time lag +0.015[s]). Velocity of 80[mm/s] is high speed press, and the defect of metal penetration is caused by high pressure due to this press velocity. By using the velocity switch from high speed to low speed, the press process of suppressing the pressure is realized in short time.

At this time, the control of press velocity pattern is decided by switching velocity obtained using tried and error method. In the near future, the newly proposed pattern of press velocity must be obtained by the optimal decision for the switching velocity. Furthermore, its validity must be demonstrated by actual

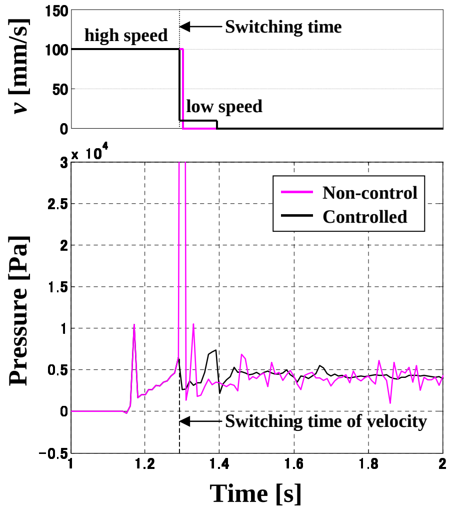


Figure 11: FLOW-3D simulation results of pressure control using switching of velocity.

experiments.

The molten metal in a mold is pressed by the upper mold by means of the position feedback control system using PID controller. The dynamical and static pressures are added as the disturbance elements in the middle of press process. In the near future, we advance the implementation using this pressure control system. Here, Figure 12 shows the block diagram of press casting system, where R_{Fout} means the reaction force from a load-cell. And, z_{in} and z_{out} are respectively reference and output position of the upper mold. M is total mass of the upper mold mass and the cylinder guide mass. W_1 and W_2 show the relation as follows.

$$W_1 : z(t) \longrightarrow h(t), \qquad W_2 : \dot{z}(t) \longrightarrow \dot{h}(t)^2 \tag{5}$$

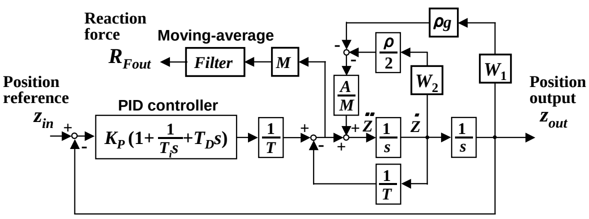


Figure 12: Block diagram of press control system in a press casting method using greensand molding.

5 CONCLUSION

In this paper, in order to realize the pressure control by controlling a press velocity, a design method of press velocity pattern for high speed press control with the reduction of rapid increase of pressure inside a mold has been proposed. Then, a mathematical model for the pressure of the molten metal in a mold was built. This model showed its effectiveness by using CFD analysis. Next, a switching pattern of press velocity from the high speed to low speed was derived from the simplified mold obtained to reduce the rapid increase of pressure. Using the obtained velocity pattern, the press control simulation has been done by CFD analysis. Good control results have been performed by the proposed method.

REFERENCES

C.Galaup, U. and H.Luehr (1986). 3d visualization of foundry molds filling. In *Proc. of 53rd International Foundry Congress, Prague (Czech), sep., pp. 107-115*.

H.Devaux (1986). Refining of melts by filtration. a water model study. In *Proc. of 53rd International Foundry Congress, Prague (Czech.), pp. 107-115*.

Hu, J, V. J. H. (1994). Dynamic modeling and control of packing pressure in injection molding. In *Journal of Engineering Materials and Technology, vol. 116, No. 2, pp. 244-249*.

I.Ohnaka (2004). Development of the innovative foundry simulation technology. In *Jornal of the Materials Process Thchnology,'SOKEIZAI', vol. 45, No. 9, pp. 37-44*.

K.Terashima (2006). Development of press casting process using green sand molding for an innovative casting production. In *Proc. of the 148th Japan foundry Engineering Society meeting, Osaka (Japan), No. 141, pp. 146*.

Louvo, A., Kalavainen, P., Berry, J. T., and Stefanescu, D. M. (1990). Methods for ensuring sounds sg iron castins. In *Trans. of American Foundry Society, vol. 98, pp. 273-279*.

Y.Matsuo, Y.Noda, and K.Terashima (2006). Modeling and identification of pouring flow process with tilting-type ladle for an innovative press casting method using greensand mold. In *Proc. of 67th World Foundry Congress, Harrogate (United Kingdom), pp. 210/1-10*.

Y.Noda, S.Makio, and K.Terashima (2006). Optimal sequence control of automatic pouring system in press casting process by using greensand mold. In *Proc. of SICE-ICASE International Joint Conference, Busan (Korea), pp. FE06-4*.

BELL SHAPED IMPEDANCE CONTROL TO MINIMIZE JERK WHILE CAPTURING DELICATE MOVING OBJECTS

Arjun Nagendran, Robert C Richardson
School of Computer Science, University of Manchester, Manchester, United Kingdom
arjun.nagendran@cs.man.ac.uk, R.C.Richardson@cs.man.ac.uk

William J. Crowther
School of Mechanical, Aerospace & Civil Engineering, University of Manchester, Manchester, United Kingdom
bill.crowther@manchester.ac.uk

Keywords: Catching, Impedance Control, Minimum Jerk, Bell Shaped Acceleration.

Abstract: Catching requires the ability to predict the position and intercept a moving object at relatively high speeds. Because catching is a contact task, it requires an understanding of the interaction between the forces applied and position of the object being captured. The application of force to a mass results in a change in acceleration. The rate of change of acceleration is called jerk. Jerk causes wear on the manipulator over time and can also damage the object being captured. This paper uses a curve that asymptotes to zero gradient at +/- infinity to develop an impedance controller, to decelerate an object to a halt after it has been coupled with the end effector. It is found that this impedance control method minimizes the jerk that occurs during capture, and eliminates the jerk spikes that are existent when using spring dampers, springs or constant force to decelerate an object.

1 INTRODUCTION

A catch can be defined as the entire process of intercepting a moving object (by a manipulator), wherein the object becomes attached to the manipulator, and decelerating the object to bring it to a halt. Catching in robotics is an important task since it is an extension to being able to pick up stationary objects. Catching has a wide variety of application areas including manufacturing industries, sports and space robotics. The ability to consistently catch objects can be useful in certain sports like baseball for repeated pitching practice. Catching a ball using a baseball glove (Riley and Atkeson, 2002) and juggling and catching balls (Sakaguchi et. al, 1991, Beuhler et. al 1994) have been studied previously. Burridge et. al (1995), provide an insight into dynamical pick and place robots. This can be useful in picking moving objects randomly from conveyor belts. Most of the literature on catching describes trajectory planning and interception of the object before the catch. The catch itself is generally thought to be an inelastic collision. Minimizing impact during capture and regulating the forces thereafter is important to limit damage to the object.

The task of capturing a moving object by robotic manipulators presents significant difficulties. The process involves being able to accurately predict the moving object's position in time and move the manipulator to the position where it can intercept the object (Sakaguchi et. al, 1991). Once the object has been intercepted, it becomes a part of the manipulator (Kovecses et. al, 1999) and hence, the dynamics of the manipulator change. These need to be taken into consideration during the post-capture phase. It is required to decelerate the object within the allowable workspace of the manipulator (Lin et. al, 1989) to prevent mechanical damage to the system. At the same time, care must be taken to decelerate the object within its permissible limits.

During the capture phase, a certain amount of impact occurs depending on the mismatch in velocities of the manipulator and the moving object. Yoshikawa et. al (1994) present a relationship between the relative velocities between moving objects and the resulting impulse forces and go on to calculate the optimum attitude of arms to minimize mechanical shock. Once the object has been captured, the kinetic energy of the object must be dissipated as work done. This is achieved by decelerating the object over a certain distance. There are several methods of decelerating an object after capture. A well known method is the use of damped springs. Constant force or springs can also be used in order to perform the same task. The force profile used (models of spring dampers, springs or constant force) is crucial in determining the deceleration and jerk experienced by the object.

During the process of catching, position control of the manipulator is an important task. Although position control can be used to move a manipulator to intercept the object, this alone is insufficient to successfully capture the object. While decelerating the object, it is important to take into account, both the position of the manipulator with respect to its workspace and also the force being applied to decelerate the object. Hogan N (1985) in his three-part paper presents an approach to control the dynamic interaction between the manipulator and its environment. The author states that control of force or position alone is insufficient and that dynamic interaction between the two is required. This is referred to as Impedance Control. Applying force depending on time is inappropriate since it does not ensure that the object is stopped over a certain distance. By applying a force, depending on the position of the object, the method ensures that the moving body is brought to a halt by removing it's kinetic energy over a certain distance.

The first derivative of acceleration is called jerk. Jerk is undesirable as it increases the rate of wear on the manipulator and can also cause damage to the object being captured. It is known to cause vibration and is a measure of impact levels that can excite unmodelled dynamics. This effect is more evident in delicate or flexible structures (Muenchhof and Singh, 2002, Barre et. al, 2005). It has been stated (Kyriakopoulos and Saridis, 1991) that jerk adversely affects the efficiency of the control algorithms and joint position errors increase with jerk. P Huang et. al (2006) in their work state that jerk affects the overall stability of the system and also causes vibrations of the manipulator and hence must be minimized. Macfarlane and Croft (2001) state that jerk limitation results in improved path tracking, reduced wear on the robot and also results in smoothed actuator loads.

In this paper, we assume that the process of tracking and intercepting an object has been completed. We then analyze the use of springs, spring dampers and constant force in decelerating the object during post-capture (once capture has occurred). It is found that these methods result in a high jerk. Hence a method to decelerate an object over a certain distance keeping the jerk to a minimum is proposed. The method establishes a bell shaped impedance relationship between force and position. The results of this method are then compared to the other methods.

2 CAPTURE METHODS

A moving object has a certain amount of kinetic energy associated with it. This is dependant on the mass of the object and its velocity. For a body of mass ‘m’ kg, travelling with a velocity ‘v’ m/s, the kinetic energy is given by:

$$Kinetic\ Energy = \tfrac{1}{2}\ m\ v^2 \qquad (1)$$

In order to bring the object to rest, a certain amount of force must be applied in a direction, opposite to that of the motion of the object. For the object to completely come to rest, it is required that the amount of work done be equal to the kinetic energy of the object. The work done is given by:

$$Work\ Done = Force * Displacement \qquad (2)$$

Equating (1) and (2),

$$Force * Displacement = \tfrac{1}{2}\ m\ v^2 \qquad (3)$$

Using equation (3), the force required to decelerate an object over a certain distance can be worked out. This however is a constant force. As the distance over which the object must be decelerated to a halt becomes small, the amount of force to be applied becomes large and vice versa. Since force is directly proportional to acceleration (from Newton's equation $F = m * a$), it follows that the deceleration experienced by an object is greater when the object is brought to a halt over a shorter distance than over a longer distance. Hence, if the maximum deceleration tolerable by a body is known, the distance over which it can be brought to a halt by applying a certain amount of force can be worked out using equation (3). To decelerate the body, force can be applied in different ways. Although force control alone is sufficient to decelerate the object, it is important to take into account, both the position of the object and the force being applied to it (Hogan, 1985). An impedance controller can be used wherein the output force is dependant on the position of the object. This ensures that the amount of deceleration experienced by the object at any position can be kept within predefined limits. Impedance control requires measuring the position of the object, and applying a force depending on the desired impedance. The desired impedance determines the amount of force to be applied depending on the object's position. The amount of force applied controls the position of the object, thus establishing a dynamic relationship between force and position. Although the term impedance control is usually associated with spring damper response, in a broader sense, the desired impedance can be a constant force, a spring or a spring damper.

3 SIMULATION

The dimensional parameters used in the simulation are mass, velocity and distance. We define the following dimensionless variables in order to perform non dimensional analysis of the results:

$$\hat{x} = \frac{x}{s}; \quad \hat{F} = \frac{F}{\frac{mv^2}{s}}; \quad \hat{a} = \frac{a}{\frac{v^2}{s}}; \quad \hat{j} = \frac{j}{\frac{v^3}{s^2}} \qquad (4)$$

where x is displacement, s is total distance over which body decelerates, m is mass, v is velocity, F is force, a is acceleration and j is jerk.

To compare the above impedance control methods a simulation model was built using Visual Nastran 4D software. This was interfaced to a simulink model of the impedance controller. It involves an object of mass 5 kg, moving with a velocity of 5m/s. It is assumed that the object has been successfully intercepted and coupled to the end effector. A linear actuator is used to decelerate the object. The impedance controller varies the amount of force exerted by the linear actuator depending on the position of the moving object (and the force model – spring. etc). In order to make a fair comparison of the different impedance controllers, it was decided to decelerate the object to a halt over a fixed distance of 2m. The results for each of the methods are discussed below.

3.1 Jerk Analysis - Constant Force

The first model of the impedance controller was designed to exert a constant force to decelerate the object. Because the desired impedance is a constant force irrespective of the position, the requirement for a feedback loop is eliminated. The constant force required was worked out using equation (3). For the chosen values of mass (5kg) and velocity (5m/s), the kinetic energy of the object is 62.5Nm. The distance over which the object must decelerate is given to be 2m. Hence using (3), the force required is 31.25N. This constant force was applied to the moving object in the simulation. When constant force is used to decelerate the vehicle, the sudden application of force at the point of contact and also the sudden removal of force at the end, result in a jerk. A graph of $\hat{x}$ against $\hat{j}$ is shown in Figure 1. The spikes at the beginning and the end indicate a high jerk at the points of application and removal of the force, and in theory are infinite.

Figure 1: Jerk experienced when constant force is used.

3.2 Jerk Analysis - Spring

In order to minimize the jerk that occurs at the beginning of the capture, it is important that the force being applied gradually increases from zero to a maximum value, with time. This kind of behaviour is characteristic of a spring, since the amount of force applied by the spring is proportional to the displacement of the object. As the spring is compressed, the force being applied increases. This behaviour was simulated using the impedance controller shown in Figure 2. The relation between the force and position (or desired impedance) is given as *Force = Spring Constant * displacement*. The distance over which the body comes to rest is kept the same as before (2m). The spring constant 'k' was chosen to achieve this behaviour by equating the energy of the object to the energy of a spring:

$$\tfrac{1}{2}\ m\ v^2 = \tfrac{1}{2}\, k\, x^2 \qquad (5)$$

where 'k' is the spring constant, and 'x' is the displacement. The kinetic energy of the object is 62.5 Nm. The displacement 'x' is 2m, which is the distance over which the body must decelerate. Using these values in the equation (5), 'k' is found to be 31.25 N/m. The free body diagram equivalent to the resulting system is shown in Figure 3. It must be noted, that using a spring to stop the object over the same distance as before (2m) requires the maximum value of deceleration to be twice as much as when using constant force.

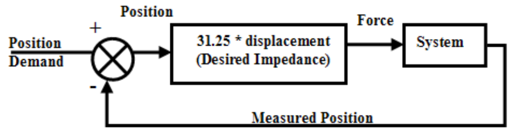


Figure 2: Impedance controller as spring.

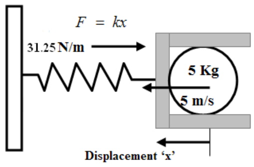


Figure 3: Free Body Diagram: Spring system.

The jerk profile when using a spring to decelerate the object is shown in Figure 4. It can be seen that the jerk is zero initially when a spring is used as compared to when applying a constant force. However, at the end, when the body comes to rest, the spring continues to apply a force proportional to the displacement, and stopping the body at that position results in a jerk spike as indicated.

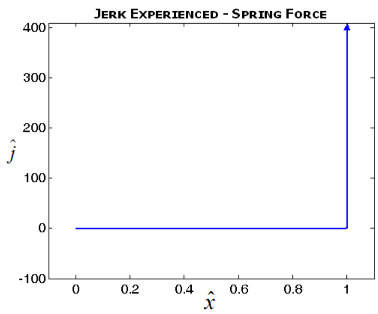


Figure 4: Jerk when spring behaviour impedance is used.

3.3 Jerk Analysis – Spring Damper

In order to eliminate the jerk that occurs towards the end of a spring system, the use of a critically damped spring damper system is considered. The impedance controller for this system is shown in Figure 5. The desired impedance for this system is given by $Force = kx + c\dot{x}$, where '*k*' is the spring constant, '*c*' is the damping constant, '*x*' is the displacement and '$\dot{x}$' is the velocity of the object. The spring constant and damping constant are chosen so that the body decelerates over 2m.

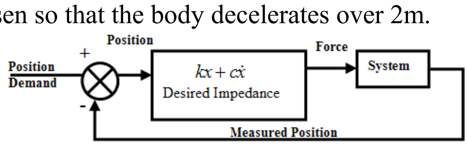


Figure 5: Spring Damper impedance control.

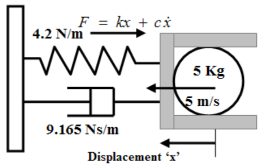


Figure 6: Free Body Diagram: Spring Damper System.

The values of 'c' and 'k' to achieve this are found to be 9.165 Ns/m and 4.2 N/m respectively. The resulting system would then behave as a spring and a damper, the free body diagram of which is shown in Figure 6. The force exerted to stop the object is high initially and gradually decreases when a spring damper is used.

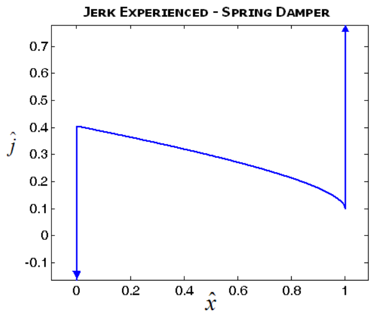


Figure 7: Jerk when spring damper impedance is used.

Because the force is less towards the end, the jerk towards the end is lower (for the chosen sampling interval) than in the case of the spring. However, the large amount of force applied at the beginning results in a high jerk as shown in Figure 7.

4 BELL SHAPED IMPEDANCE CONTROL

From the above analysis of using constant force, spring and a spring damper to decelerate a body, it is immediately clear that jerk is an issue with all the methods. In theory, all these methods cause an infinite amount of jerk on the body, and for the chosen sample interval, a finite but large amount of jerk as shown in the graphs. This jerk can be responsible for an unsuccessful catch as the object may bounce off on impact, or sustain damage. In order to keep the jerk to a minimum, we propose a new method of impedance control, where the relationship between force and position is in the form of a bell curve. The method uses knowledge of statistics and probability distributions to establish the required relationship. The graph of the probability density of a raised cosine distribution is in the shape of a bell curve. This knowledge can be used to establish a relationship between the force and position. The probability density function of this distribution is given as:

$$f(x;u,s)=\frac{1}{2s}\left[1+\cos\left(\frac{x-u}{s}\pi\right)\right] \qquad (6)$$

and is supported in the interval $u - s$ to $u + s$. The amplitude of this distribution is $1/s$ and occurs at u (Figure 8).

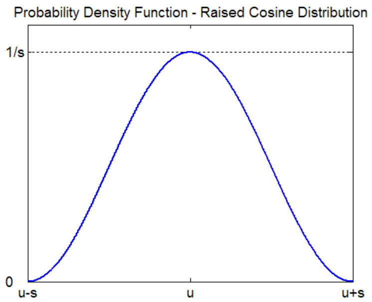


Figure 8: Raised Cosine Distribution - Bell Shaped.

It will be advantageous to establish a relationship between force and position such that the body being captured decelerates over a known distance and experiences a certain maximum deceleration. From the above equation (6), the distance over which the object must decelerate is between u – s and u + s. Hence, u and s are chosen as half the maximum distance. Because the maximum amplitude is dependant on s, a scaling factor is required to achieve the required maximum deceleration for a given distance. Hence, equation (6) is modified to include a scaling factor A chosen such that A/s is the maximum force tolerable. If the maximum deceleration is known, the maximum force tolerable by the object, using Newton's equation is *Force = mass * deceleration*. In order to establish an impedance relationship, a force must be applied depending on the position of the object and hence, equation (6) can be written in terms of force and position as

$$Force=\frac{A}{2s}\left[1+\cos\left(\frac{Position-u}{s}\pi\right)\right] \qquad (7)$$

Equation (7) results in a force being output depending on the position of the object and ensures that the deceleration of the object is kept within the tolerable limit. It is important to note that the area under this bell curve determines the total work done, and in order to decelerate the body to a complete halt, this must be equal to the total kinetic energy of the object. The area under this bell curve is 50% of the total area under the rectangle with sides equal to the maximum deceleration and maximum distance over which the body decelerates. This is illustrated in the example that follows. We compare this method to the example used with the spring-damper, spring and constant force methods. The distance over which the body decelerates is 2m. Hence, u and s are chosen to be 1 and the relative position of the object is from 0m to 2m during which the force is applied to decelerate the object. The maximum amplitude of this curve is however $1/s$ which is equal to 1, for the chosen s. The area under the curve must be equal to the kinetic energy of the object. For the 5kg mass travelling at 5m/s, the kinetic energy is 62.5kgm^2/s^2 (or Nm), as established previously. The area under the bell curve is given as *Area = ½ * Force * displacement* where *Force* is worked out using Newton's equation and displacement is the distance over which the body decelerates (50% area as mentioned earlier). Equating this to the kinetic energy of the object, the force required is found to be 62.5N. Hence, A must be chosen such that A/s = 62.5. Since s = 1, A = 62.5. Using the calculated values of A, u and s, the final equation for force, in terms of position or the desired impedance to minimize jerk is implemented.

The force applied to decelerate the object was determined by the impedance relationship established in equation (7). The maximum deceleration experienced by the object is the same as when a spring is used. A graph of force applied using the impedance relationship to decelerate the object against time is shown in Figure 9. Because the position of the object changes faster initially due to its approach velocity, the force required rises steeply at the beginning. The force applied based on the object's position, slows the object down and gradually eases off so as to stop the object over the desired distance of 2m. The jerk profile for this method is shown in Figure 10.

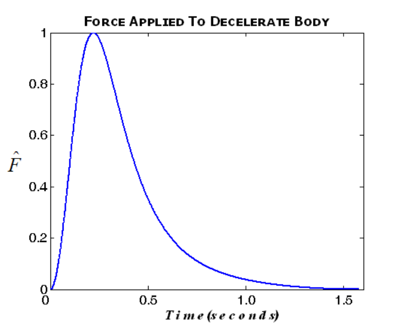


Figure 9: Force applied using Bell Shaped Impedance.

It is a smooth curve, with no spikes and the amount of maximum jerk is very low as compared to any of the other methods. In reality, actuators themselves have inherent dynamics that prevent them from generating instantaneous changes in force. The greater the required instantaneous change in force, the more pronounced the actuator dynamics will become. Therefore, minimum jerk profiles, that limit the required rate of change of force, can be implemented with a greater degree of accuracy.

5 DISCUSSIONS & CONCLUSION

The jerk graphs reveal that the amount of jerk is greatly reduced if a bell shaped curve of force against position is used to capture the object (Figure 10). However, in comparison with the constant force method, the amount of deceleration experienced by the object is high. A trade off between the amount of tolerable jerk and tolerable acceleration is required to be able to generate the required response. An important assumption in this method is that the velocity and mass of the object at the point when capture occurs is known. This ensures that the body decelerates within a certain maximum distance and allows for the force to be specified at every position along its path. Any error in this estimation can result in incorrect calculation of kinetic energy and the object will not stop within the required distance.

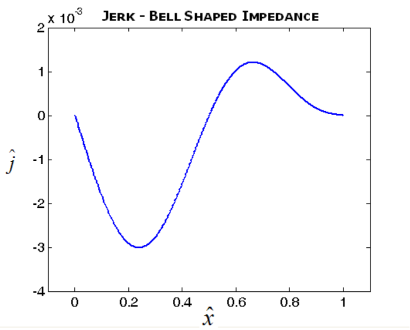


Figure 10: Jerk for Bell Shaped Impedance control.

For accurate calculation, the velocity and mass of the object must be estimated in real time, after which self tuning can be used to generate the required bell shaped impedance control. Additionally, capturing an object requires a high speed of operation and it is much more difficult to apply quick changing forces from actuators at high speeds. The smooth bell shaped acceleration profile also means that forces can be applied with much more ease, due to the gradually changing curve.

REFERENCES

P. Huang, Y. Xu, B. Liang (2006) Global Minimum-jerk trajectory planning of Space Manipulator, *International Journal of Control, Automation, and Systems*, Vol. 4 (4), pp. 405-413.

K. J. Kyriakopoulos, G. N. Saridis (1991) Minimum jerk trajectory planning for robotic manipulators, *Proc. SPIE*, Vol. 1387, pp. 159-164.

A. Piazzi, A. Visioli (1997) An interval algorithm for minimum-jerk trajectory planning of robot manipulators, *Proc. of the 36th Conference on Decision and Control*, San Diego, California USA, pp. 1924.

M. Muenchhof, T. Singh (2002) Desensitized Jerk Limited-Time Optimal Control of Multi-Input Systems, *Journal of Guidance, Control, and Dynamics*. Vol. 25(3) pp.474-481.

K. J. Kyriakopoulos, G. N. Saridis (1988) Minimum-jerk path generation, *Proc. of IEEE International Conference on Robotics and Automation*, Philadelphia, PA, pp. 364-369.

N. Hogan (1985) Impedance control: An approach to manipulation: Part I- theory. *Journal of Dynamic Systems Measurement and Control*, Vol.107(11), pp.1-7.

P. J. Barre, R. Bearee, P. Borne, E. Dumetz (2005) Influence of a jerk controlled movement law on the vibratory behaviour of high-dynamics systems. *Journal of Intelligent and Robotic Systems.* Vol. 42, pp. 275–293.

S. Macfarlane and E. A. Croft (2001) Design of Jerk Bounded Trajectories for On-line Industrial Robot Applications. *IEEE International Conference on Robotics and Automation*, Seoul, S. Korea.

J. Kovecses, W. L. Cleghorn, R. G. Fenton (1999) Dynamic modelling and analysis of a robot manipulator intercepting and capturing a moving object with the consideration of structural flexibility. *Multibody System Dynamics 3*, pp. 137–162.

Z. Lin, V. Zeman , R. V. Patel (1989) On-line robot trajectory planning for catching a moving object. *Proceedings of the IEEE International Conference on Robotics and Automation*, Scottsdale, AZ, pp. 1726–1731.

T. Sakaguchi, Y. Masutani, F. Miyazaki (1991) A Study On Juggling Task, *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 91,* pp. 1418–1423.

M. Buehler, D.E. Koditschek, P.J. Kindlmann (1994) Planning and control of robotic juggling and catching tasks. *International Journal of Robotics Research,* Vol. 13(2), pp. 101-108.

R. R. Burridge, A. A. Rizzi, D.E. Koditschek (1995) Toward a Dynamical Pick and Place. *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 95,* Vol.2, pp. 292–297.

AUTHOR INDEX

AUTHOR INDEX (CONT.)

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